

Oracle Fail Safe for Windows

概要および管理ガイド

リリース 3.3.1

2002 年 9 月

部品番号 : J06612-01

ORACLE[®]

Oracle Fail Safe for Windows 概要および管理ガイド, リリース 3.3.1

部品番号: J06612-01

原本名: Oracle Fail Safe Concepts and Administration Guide, Release 3.3.1 for Windows

原本部品番号: A96684-01

Copyright © 1996, 2002, 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載された制約条件に従うものとします。著作権、特許権およびその他の知的財産権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	xv
対象読者	xvi
構成	xvi
関連資料	xviii
表記規則	xviii
1 Oracle Fail Safe の基礎知識	
1.1 Oracle Fail Safe とは	1-2
1.2 Oracle Fail Safe の利点	1-3
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	1-3
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	1-7
1.3 典型的な Oracle Fail Safe の構成	1-8
1.4 Oracle Fail Safe ソリューションの展開	1-10
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 クォーラム・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	2-5
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	2-12
2.5	クラスター・グループとクラスター別名	2-12
2.6	フェイルオーバー	2-13
2.6.1	計画外フェイルオーバー	2-14
2.6.1.1	リソース障害による計画外フェイルオーバー	2-14
2.6.1.2	ノードの障害または使用不可状態による計画外フェイルオーバー	2-16
2.6.1.3	グループの計画的フェイルオーバー	2-17
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	2-18
2.6.3	リソース障害の検出方法	2-20
2.6.4	リソース再起動ポリシー	2-21
2.6.5	リソースのフェイルオーバー・ポリシー	2-21
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	2-24
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 フェイルオーバーへの影響	2-26
2.6.9	グループ・フェイルオーバーと「優先所有者ノード」リスト	2-27
2.6.10	グループのフェイルオーバー・ノードの決定	2-27
2.7	フェイルバック	2-29
2.7.1	グループ・フェイルバックと「優先所有者ノード」リスト	2-30
2.7.2	フェイルオーバー後のクライアントの再接続	2-31

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	3-2
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	3-4
3.1.3	作業負荷のパーティション化構成	3-6
3.1.4	階層化構成	3-8
3.2	障害耐久力と高可用性	3-10
3.3	クライアントとアプリケーションの統合	3-11

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	4-2
4.2	ウィザード入力項目の Oracle Fail Safe での処理	4-3
4.3	クラスタ・セキュリティの管理	4-5
4.3.1	Oracle Services for MSCS	4-6
4.3.1.1	Oracle Fail Safe のセキュリティ設定ツールを使用したアカウント更新	4-7
4.3.2	Oracle Fail Safe Manager	4-8
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	4-9
4.7	複数仮想アドレスを使用する構成	4-10
4.8	既存クラスタへのノードの追加	4-11

5 FSCMD コマンドライン・インタフェース

5.1	FSCMD	5-2
5.1.1	書式	5-2
5.1.2	説明	5-2
5.1.3	コマンド・パラメータ	5-3
5.1.4	コマンド修飾子	5-4
5.1.5	使用上の注意	5-6
5.1.6	コマンドの例	5-7
5.1.7	スクリプトの例	5-7

6 トラブルシューティング・ツール

6.1	検証操作	6-2
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

7 可用性を高めるためのシングルインスタンス・データベースの構成

7.1	スタンドアロン・シングルインスタンス・データベースの検出	7-2
7.2	スタンドアロン・シングルインスタンス・データベースのための Oracle Net 構成	7-3
7.2.1	DBCA により作成したデータベースのための Oracle Net 構成の更新	7-3
7.2.2	リスナーの定義	7-4
7.2.3	SID リスト・エントリおよび Oracle8i または Oracle9i ソフトウェアへの アップグレード	7-4
7.2.4	複数のリスナーを伴うノードでの Oracle Net の構成	7-6
7.2.5	共有サーバーの構成とスタンドアロンの Oracle8i または Oracle9i データベース	7-7
7.3	シングルインスタンス Oracle データベースのグループへの追加	7-9
7.3.1	構成前に	7-9
7.3.2	構成手順	7-10
7.3.3	Oracle データベースの構成データ	7-11
7.3.3.1	ノードの選択	7-12
7.3.3.2	仮想アドレス	7-14
7.3.3.3	データベース識別情報	7-15
7.3.3.3.1	パラメータ・ファイルと、SPFILE を使用する Oracle9i データベース	7-16
7.3.3.3.2	パラメータ・ファイルと、DBCA により作成した Oracle9i データベース	7-16
7.3.3.4	データベース認証	7-17
7.3.3.5	データベース・パスワード	7-19
7.4	Oracle Net リスナー・リソースの作成および構成	7-20
7.4.1	専用サーバー・モードでの共有ソケットの使用法	7-21
7.4.2	可用性の高いシングルインスタンス・データベースへのクライアント接続	7-21
7.4.3	データベースをグループに追加すると更新される Oracle Net 構成	7-22
7.4.3.1	Oracle Fail Safe により行われる tnsnames.ora ファイルの更新内容	7-22
7.4.3.2	Oracle Fail Safe により行われる listener.ora ファイルの更新内容	7-24
7.4.3.3	Oracle Fail Safe により行われる sqlnet.ora ファイルの更新内容	7-24
7.4.4	可用性が高ま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法	7-24
7.4.5	共有サーバーを使用するデータベースのサポート	7-25
7.4.5.1	Oracle8 データベース用の共有サーバー	7-25
7.4.5.2	Oracle8i または Oracle9i データベースのための共有サーバー	7-28
7.5	シングルインスタンス・データベースのセキュリティ要件	7-29
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	7-29
7.5.2	SYSDBA アカウントのパスワードの変更	7-30
7.5.3	Oracle Database Upgrade Assistant を使用したフェイルセーフ・データベースの アップグレード	7-31

7.6	可用性を高めるための ctxsrv サーバーの構成	7-33
7.7	Oracle Enterprise Manager との統合	7-34
7.7.1	Oracle Intelligent Agent のグループへの追加	7-35
7.8	シングルインスタンス・データベース・リカバリの最適化	7-36
7.9	シングルインスタンス・フェイルセーフ・データベースに対する管理作業の実行	7-37
7.10	透過的アプリケーション・フェイルオーバー (TAF) の構成	7-38
7.10.1	透過的アプリケーション・フェイルオーバーによりリストアされる機能	7-39
7.10.2	透過的アプリケーション・フェイルオーバー実行時の処理	7-40
7.10.3	透過的アプリケーション・フェイルオーバーのための tnsnames.ora ファイルの 変更	7-42
7.10.3.1	HOST パラメータの構成	7-42
7.10.3.2	CONNECT_DATA パラメータの構成	7-43
7.10.4	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	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化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	データベースをグループに追加する際の問題	7-47
7.11.4	グループのオンライン化の問題	7-48
7.11.5	プロセスが集中している操作の際のグループ・フェイルオーバー	7-50
7.11.6	データベース認証	7-50
7.11.7	サンプル・データベースの問題	7-51
7.11.8	仮想サーバーの構成で検出される問題	7-52
7.11.8.1	仮想アドレスの構成で発生する問題	7-52
7.11.8.2	リスナー作成時の問題	7-54
7.11.8.3	アーカイブされた listener.ora または tnsnames.ora ファイル	7-54
7.11.8.4	ロールバック・ファイル	7-54
7.11.9	セキュリティ・アクセスおよび認証の問題	7-54
7.11.10	クライアントがデータベースにアクセスできない	7-55

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

8.1	概要	8-2
8.2	スタンドアロン Oracle Forms Load Balancer Server の検出	8-5
8.3	Oracle Forms Load Balancer Server のグループへの追加	8-5
8.3.1	構成前に	8-6
8.3.2	構成手順	8-6
8.3.3	Oracle Forms Load Balancer Server の構成データ	8-7

8.3.3.1	ノードの選択	8-8
8.3.3.2	Oracle Forms Load Balancer Server リスナーの仮想アドレス	8-10
8.3.3.3	Oracle Forms Load Balancer Server の識別情報	8-10
8.4	Oracle Forms Load Balancer Server のセキュリティ要件	8-12
8.5	起動パラメータの変更	8-12
8.5.1	Forms CGI-bin 実行可能ファイルの起動パラメータの変更	8-13
8.5.2	Oracle Forms Load Balancer Client の起動パラメータの変更	8-13
8.5.3	Oracle Forms Server の起動パラメータの変更	8-13
8.5.4	Oracle Forms Load Balancer Server の起動パラメータの変更	8-14
8.6	Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング	8-14

9 可用性を高めるための Oracle Forms Server の構成

9.1	概要	9-2
9.2	スタンドアロン Oracle Forms Server の検出	9-3
9.3	Oracle Forms Server のグループへの追加	9-4
9.3.1	構成前に	9-4
9.3.2	構成手順	9-5
9.3.3	Oracle Forms Server の構成データ	9-6
9.3.3.1	ノードの選択	9-7
9.3.3.2	Forms Server の仮想アドレス	9-9
9.3.3.3	Forms Server の識別情報	9-10
9.3.3.4	Forms のパラメータ	9-11
9.3.3.5	Oracle Forms アプリケーション・ファイルのパス	9-13
9.3.3.6	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	9-14
9.4	Oracle Forms Server のセキュリティ要件	9-15
9.5	Oracle Forms Server へのクライアント接続	9-16
9.6	Oracle Forms Server に関連する問題のトラブルシューティング	9-16
9.6.1	ユーザーが Oracle Forms Server にアクセスできない	9-17
9.6.2	Web ブラウザで Java アプレットをダウンロードできない	9-17
9.6.3	Java アプレットを実行できない	9-18
9.6.4	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	9-18
9.6.5	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	9-19
9.6.6	Oracle Forms アプリケーションが正しく実行されない	9-19
9.6.7	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	9-19

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

10.1	概要および実装のオプション	10-2
10.1.1	Web ベースの実装	10-2
10.1.2	クライアント / サーバー実装	10-4
10.1.3	マスター / スレーブ実装	10-6
10.2	スタンドアロン Oracle Reports Server の検出	10-6
10.3	Oracle Reports Server のグループへの追加	10-7
10.3.1	構成前に	10-7
10.3.2	構成手順	10-8
10.3.3	Oracle Reports Server の構成データ	10-9
10.3.3.1	ノードの選択	10-10
10.3.3.2	仮想アドレス	10-12
10.3.3.3	Oracle Reports Server の識別情報	10-13
10.3.3.4	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	10-14
10.3.3.5	Oracle Reports Server によりアクセスされるデータベース	10-16
10.3.3.6	Oracle Reports Server の実行に使用されるアカウント	10-17
10.4	Oracle Reports Server へのクライアント接続	10-18
10.4.1	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	10-19
10.5	マスター / スレーブ実装のための構成後の手順	10-20
10.5.1	マスター / スレーブ例の概要	10-20
10.5.2	マスター / スレーブ間の通信を使用可能にする方法	10-21
10.5.2.1	マスター・サーバーのための調整	10-21
10.5.2.2	スレーブ・サーバーのための調整	10-23
10.5.3	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	10-23
10.5.3.1	マスター Oracle Reports Server のための調整	10-24
10.5.3.2	スレーブ・サーバーのための調整	10-25
10.6	可用性の高い Oracle Reports Server へのユーザー・アクセスの制御	10-25
10.7	Oracle Reports のスケジューリング	10-26
10.7.1	可用性の高い Oracle Reports Server でのジョブ・リポジトリの使用 方法	10-27
10.8	Oracle Reports Server に関連する問題のトラブルシューティング	10-28
10.8.1	Oracle HTTP Server で rwcgi60.exe ファイルを見つけられない	10-28
10.8.2	Oracle Reports Server で rwcgi60.exe ファイルをダウンロード	10-28
10.8.3	ユーザーが Oracle Reports Server と通信できない	10-29
10.8.4	Web ブラウザで Oracle Reports 定義ファイルが見つからない	10-29

10.8.5	Oracle Reports アプリケーションが正しく実行されない	10-29
10.8.6	ユーザーが Reports を出力できない	10-30

11 可用性を高めるための Oracle HTTP Server の構成

11.1	概要	11-2
11.2	スタンドアロン Oracle HTTP Server の検出	11-4
11.3	Oracle HTTP Server のグループへの追加	11-4
11.3.1	構成前に	11-4
11.3.2	Oracle HTTP Server の構成手順	11-5
11.3.3	Oracle HTTP Server の構成データ	11-6
11.3.3.1	ノードの選択	11-7
11.3.3.2	Oracle HTTP Server の識別情報	11-8
11.3.3.3	Oracle HTTP Server のディレクトリ	11-10
11.3.3.4	Oracle HTTP Server の仮想アドレス	11-12
11.4	Oracle HTTP Server へのクライアント接続	11-13
11.5	Oracle HTTP Server のグループからの削除	11-14
11.6	Oracle HTTP Server のセキュリティ要件	11-15
11.7	Oracle HTTP Server に関連する問題のトラブルシューティング	11-15
11.7.1	Oracle HTTP Server のグループへの追加の問題	11-15
11.7.2	ユーザーが Oracle HTTP Server にアクセスできない	11-16
11.7.3	ユーザーが Oracle HTTP Server Web サイトに接続できない	11-16
11.7.4	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	11-17

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

12.1	概要	12-3
12.2	スタンドアロン Oracle Applications Concurrent Manager の検出	12-4
12.3	Oracle Applications Concurrent Manager のグループへの追加	12-4
12.3.1	構成前に	12-5
12.3.2	構成手順	12-5
12.3.3	Oracle Applications Concurrent Manager の構成データ	12-7
12.3.3.1	ノードの選択	12-8
12.3.3.2	Oracle Applications Concurrent Manager 仮想アドレス	12-10
12.3.3.3	Concurrent Manager の識別情報	12-11
12.3.3.4	Concurrent Manager の認証	12-13
12.3.3.5	Concurrent Manager のデータベース	12-14
12.4	Oracle Applications Concurrent Manager のセキュリティ要件	12-15

12.5	Oracle Applications Concurrent Manager に関連する問題のトラブルシューティング	12-16
12.5.1	Oracle Applications Concurrent Manager のグループへの追加の問題	12-17
12.5.2	グループのオンライン化の問題	12-17
12.5.3	Concurrent Manager 用の Windows サービスが誤って削除された	12-18
12.5.4	Concurrent Manager がオンライン化できない	12-18

13 可用性を高めるための Oracle MTS Service の構成

13.1	概要	13-3
13.2	スタンドアロン Oracle MTS Service の検出	13-4
13.3	Oracle MTS Service のグループへの追加	13-5
13.3.1	構成前に	13-5
13.3.2	構成手順	13-6
13.3.3	Oracle MTS Service の構成データ	13-7
13.3.3.1	ノードの選択	13-9
13.3.3.2	Oracle MTS Service 仮想アドレス	13-11
13.3.3.3	Oracle MTS Service 識別情報	13-12
13.3.3.4	Oracle MTS Service データベース	13-13
13.3.3.5	Oracle MTS Service の認証	13-14
13.4	Oracle MTS Service のセキュリティ要件	13-15
13.5	Oracle MTS Service に関連する問題のトラブルシューティング	13-16
13.5.1	Oracle MTS Service のグループへの追加の問題	13-17
13.5.2	グループのオンライン化の問題	13-17
13.5.3	削除された Oracle MTS Service の Windows サービス	13-17

14 可用性を高めるための汎用サービスの構成

14.1	概要	14-2
14.1.1	Oracle Fail Safe を使用する利点	14-2
14.1.2	可用性を高める構成にしない汎用リソース	14-3
14.2	スタンドアロン汎用サービスの検出	14-3
14.3	汎用サービスのグループへの追加	14-3
14.3.1	構成手順	14-4
14.3.2	汎用サービス用構成データ	14-5
14.3.2.1	ノードの選択	14-6
14.3.2.2	汎用サービスの識別情報	14-7
14.3.2.3	汎用サービスの起動パラメータ	14-9
14.3.2.4	汎用サービスにより使用されるディスク	14-11

14.3.2.5	汎用サービスの依存性	14-12
14.3.2.5.1	汎用サービスの依存性の指定	14-13
14.3.2.5.2	汎用サービスと仮想アドレスの依存性	14-14
14.3.2.6	汎用サービスのレジストリ・キー	14-14
14.4	汎用サービスのセキュリティ要件	14-15
14.5	サンプル汎用サービスの構成	14-16
14.6	汎用サービスに関する問題のトラブルシューティング	14-17

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-2
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 5 以上に推奨されるソリューション	B-4
B.3.2	Windows 2000 に推奨されるソリューション	B-4

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-2
C.2	バージョン情報の検出	C-3
C.3	Oracle Fail Safe の問題のトレース	C-3
C.4	トレース・ファイルとアラート・ファイルの保管	C-6

用語集

索引

図のリスト

1-1	Microsoft クラスタでの Oracle Fail Safe によるフェイルオーバー	1-4
1-2	Oracle Fail Safe Manager	1-6
1-3	Oracle Fail Safe Manager のメニューと内容	1-7
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	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	2-10
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	2-11
2-7	「ツリーにクラスタを追加」ダイアログ・ボックスの「クラスタ別名」	2-13
2-8	リソースのフェイルオーバー	2-15
2-9	ノードのフェイルオーバー	2-17
2-10	グループの「フェイルオーバー」プロパティ・ページ	2-19
2-11	リソースの「ポリシー」プロパティ・ページ	2-20
2-12	「ノード」プロパティ・ページ	2-24
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	2-25
2-14	グループの「フェイルバック」ポリシー・プロパティ・ページ	2-29
3-1	2 ノードのアクティブ / パッシブ (スタンバイ) 構成	3-3
3-2	4 ノードのアクティブ / パッシブ (スタンバイ) 構成	3-4
3-3	アクティブ / アクティブ構成	3-5
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	3-7
3-5	Oracle Reports Server の階層化構成	3-9
3-6	Oracle Data Guard を使用した障害耐久力を高めるデータベース構成	3-11
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 Services for MSCS 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	「クラスタのダンプ」の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	7-12
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	7-13
7-3	「データベース仮想アドレス」ウィザード・ページ	7-14
7-4	「データベースの識別情報」ウィザード・ページ	7-15
7-5	「データベースの認証」ウィザード・ページ	7-18
7-6	「DBA グループへ追加の確認」ウィンドウ	7-18
7-7	「データベース・パスワード」ウィザード・ページ	7-19
7-8	「リソースをグループに追加」ウィザードの「リソース」ページ	7-35
7-9	透過的アプリケーション・フェイルオーバーのフロー・チャート	7-40
8-1	Forms Load Balancer Server を使用した Forms の 3 層アーキテクチャ	8-4

8-2	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	8-8
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	8-9
8-4	「仮想アドレスの選択」ウィザード・ページ	8-10
8-5	「Load Balancer Server の識別情報」ウィザード・ページ	8-12
9-1	Oracle Forms の 3 層アーキテクチャ	9-3
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	9-7
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	9-8
9-4	Oracle Forms の「仮想アドレスの選択」ウィザード・ページ	9-9
9-5	「Forms の識別情報」ウィザード・ページ	9-10
9-6	「Forms のパラメータ」ウィザード・ページ	9-12
9-7	「Forms パス」ウィザード・ページ	9-13
9-8	「Forms のデータベース」ウィザード・ページ	9-14
9-9	Web ベースの Forms を示すツリー・ビュー	9-15
10-1	Oracle Reports の Web 実装のための 3 層アーキテクチャ	10-3
10-2	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	10-5
10-3	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	10-10
10-4	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	10-11
10-5	Oracle Reports の「仮想アドレスの選択」ウィザード・ページ	10-12
10-6	「Reports の識別情報」ウィザード・ページ	10-13
10-7	「Reports のディスク」ウィザード・ページ	10-15
10-8	「Reports のデータベース」ウィザード・ページ	10-16
10-9	「Reports のアカウント」ウィザード・ページ	10-17
11-1	可用性の高い Web 構成のための 3 層アーキテクチャ	11-2
11-2	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	11-7
11-3	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	11-8
11-4	「Oracle HTTP Server の識別情報」ウィザード・ページ	11-9
11-5	「Oracle HTTP Server のディレクトリ」ウィザード・ページ	11-11
11-6	「仮想ディレクトリのマップ」ダイアログ・ボックス	11-11
11-7	「Oracle HTTP Server の仮想アドレス」ウィザード・ページ	11-13
12-1	Oracle Applications Concurrent Manager の 3 層アーキテクチャ	12-3
12-2	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	12-8
12-3	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	12-9
12-4	Oracle Applications Concurrent Manager の「仮想アドレスの選択」ウィザード・ ページ	12-10
12-5	「Concurrent Manager の識別情報」ウィザード・ページ	12-12
12-6	「Concurrent Manager の認証」ウィザード・ページ	12-13
12-7	「Concurrent Manager のデータベース」ウィザード・ページ	12-15
13-1	可用性の高い Oracle MTS Service のための 3 層アーキテクチャ	13-4
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	13-9
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	13-10
13-4	Oracle MTS Service の「仮想アドレス」ウィザード・ページ	13-11
13-5	「Oracle MTS Service の識別情報」ウィザード・ページ	13-12
13-6	「Oracle MTS Service のデータベース」ウィザード・ページ	13-13
13-7	「Oracle MTS Service の認証」ウィザード・ページ	13-14
14-1	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	14-6

14-2	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	14-7
14-3	「汎用サービスの識別情報」ウィザード・ページ	14-9
14-4	「汎用サービスのアカウント」ウィザード・ページ	14-10
14-5	「汎用サービスのディスク」ウィザード・ページ	14-12
14-6	依存性ツリー	14-13
14-7	「汎用サービスの依存性」ウィザード・ページ	14-14
14-8	「汎用サービスのレジストリ」ウィザード・ページ	14-15

表のリスト

2-1	Test_Group グループ内のリソースの可能所有者の例	2-28
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	7-10
7-2	FAILOVER_MODE パラメータの TYPE サブパラメータ・オプション	7-44
8-1	Oracle Forms Load Balancer Server の構成手順	8-6
9-1	Oracle Forms Server の構成手順	9-5
10-1	Oracle Reports Server の構成手順	10-8
10-2	マスター / スレーブ実装の例	10-20
11-1	Oracle HTTP Server の構成手順	11-5
12-1	Oracle Applications Concurrent Manager の構成手順	12-6
13-1	Oracle MTS Service を構成する手順	13-6
14-1	汎用サービスの構成手順	14-4
14-2	サンプル汎用サービスの構成手順	14-16
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-4

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 シングルインスタンス・データベース・サーバー
- Oracle iAS:
 - Oracle Forms Load Balancer Server
 - Oracle Forms Server
 - Oracle Reports Server
 - Oracle HTTP Server
 - Oracle Applications Concurrent Manager
- Oracle Services for Microsoft Transaction Server (Oracle Fail Safe リリース 3.2.1 より前のリリースの場合)
- Windows 汎用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、Oracle Fail Safe によって、Microsoft クラスタ・システム上で稼働するソフトウェア・コンポーネントの停止時間を最短にする方法に関心がある方を対象としています。

読者は、Microsoft Cluster Server (MSCS)、Oracle Net 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

このマニュアルは、14 の章、3 つの付録、用語集および索引で構成されています。

第 1 章

この章では、クラスタおよび Oracle Fail Safe の概要を示します。

第 2 章

この章では、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

この章では、高い可用性で業務上の要求に合致する、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

この章では、複数の Oracle ホームでの Oracle Fail Safe Manager の使用方法など、セキュリティ上の注意事項を説明します。

第 5 章

この章では、FSCMD コマンドの参照情報を掲載しています。

第 6 章

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す。

第 7 章

この章では、シングルインスタンス 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や、Oracle Call Interface (OCI) と ODBC クライアントおよび ODBC 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

この章では、Oracle Forms Load Balancer Server に高い可用性を与える構成方法を説明します。

第 9 章

この章では、Oracle Forms Server に高い可用性を与える構成方法を説明します。

第 10 章

この章では、Oracle Reports Server に高い可用性を与える構成方法を説明します。

第 11 章

この章では、Oracle HTTP Server に高い可用性を与える構成方法を説明します。

第 12 章

この章では、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 13 章

この章では、Oracle Services for Microsoft Transaction Server (Oracle MTS Service と呼ぶ) に高い可用性を与える構成方法を説明します。(この情報は Oracle 8.1.6 および 8.1.7 のデータベース・サーバーのみに関連します。)

第 14 章

この章では、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

この付録では、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方法を示します。

付録 B

この付録では、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せん。

付録 C

この付録では、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の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、およびその他のリリース固有の情報は、『Oracle Fail Safe for Windows 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 for Windows インストール・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 で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。
- Oracle Call Interface の詳細は、『Oracle Call Interface プログラマーズ・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。（フェイルオーバーの設定方法などが説明されています。）
- Oracle MTS Service の詳細は、『Oracle と Microsoft Transaction Server の連携』（Oracle8i for Windows ドキュメント・セットの一部）を参照してください。

その他の関連製品の詳細は、各製品のマニュアル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します。
...	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します。
イタリック体	イタリック体は、変数を示すために使用します。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>nnn</i> ）、コマンドライン内（たとえば、 <i>/Producer=name</i> ）、および本文中のコマンド・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）で変化する情報を保持します。
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します。
<>	ユーザー指定の名前を囲む記号。

表記規則	意味
[]	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します。
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します。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。

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。現在、100% の **可用性** を保証できるソリューションは存在せず、Oracle Fail Safe もその例外ではありません。しかし、Oracle Fail Safe を使用すると、**Microsoft Cluster Server (MSCS)** で構成され Microsoft クラスタ上で稼働する、Oracle データベースおよびその他のアプリケーションの **停止時間** を、最小限に抑えることが可能になります。

この章では、次の項目について説明します。

項目	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 つ以上の 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シングルインスタンス・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、シングルインスタンス Oracle データベース、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停止時間を短縮できます。

Oracle Fail Safe は Oracle Services for MSCS と Oracle Fail Safe Manager から構成されます。

- Oracle Services for MSCS 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度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（これらが依存するソフトウェアおよびハードウェアを含む）があります。また、Oracle Services for MSCS 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。

注意： 以前のリリースでは、Oracle Services for MSCS は Oracle Fail Safe Server と呼ばれていました。

- 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

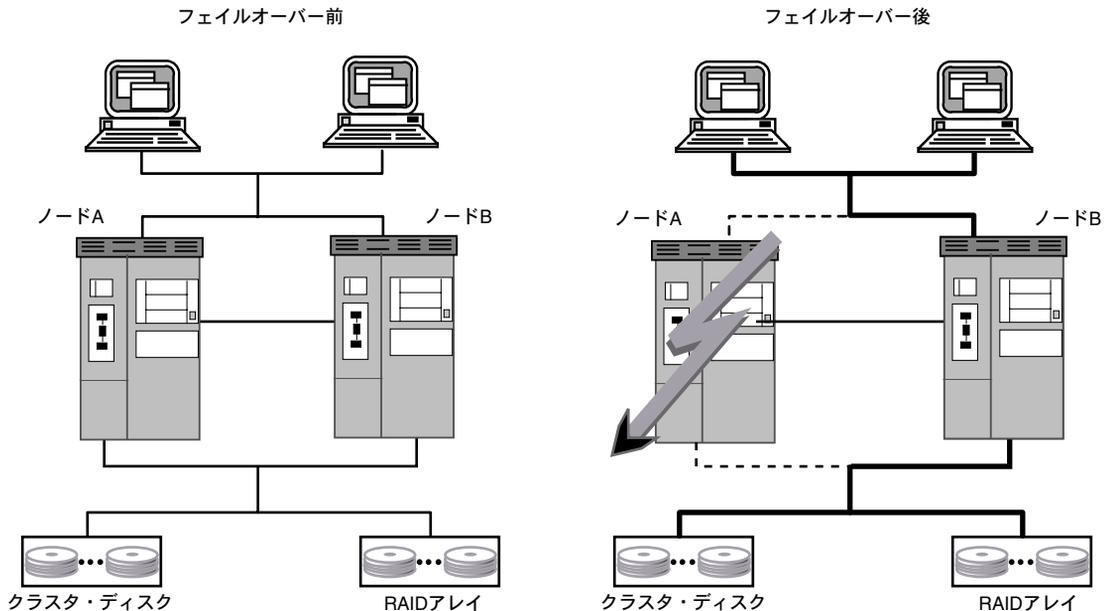
1.2.1 高い可用性を備えたリソースおよびアプリケーション

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可になった場合、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設定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可になった場合に、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。シングルインスタンス Oracle データベースがノード 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。対照的に、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おり、クラスタ内のソフトウェアの構成が簡素化されています。

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**でのインストールの場合、レスポンス・ファイルを使用して Oracle Universal Installer への入力を行い、ソフトウェアをインストールできます。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、停止時間が最短になります。詳細は、『Oracle Fail Safe for Windows 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設定、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使用するコマンドライン・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、クラスタおよびクラスタ・リソースの複数のビュー（ならびに現在の状態）が示されたツリー・ビューが表示されます。右側のペインには、ツリー・ビューから選択されたクラスタの全グループとそれらグループの現在の状態が示された、プロパティ・ページが表示されます。右側のペインの表示内容は、ツリー・ビューで選択されたオブジェクトによって異なります。特定のクラスタ、ノード、グループまたはリソースを選択すると、そのクラスタ、ノード、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

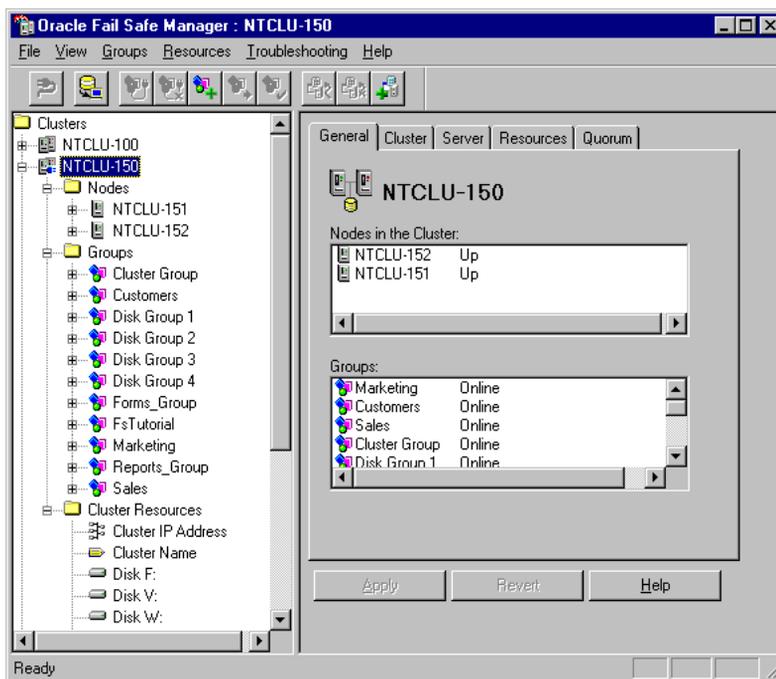
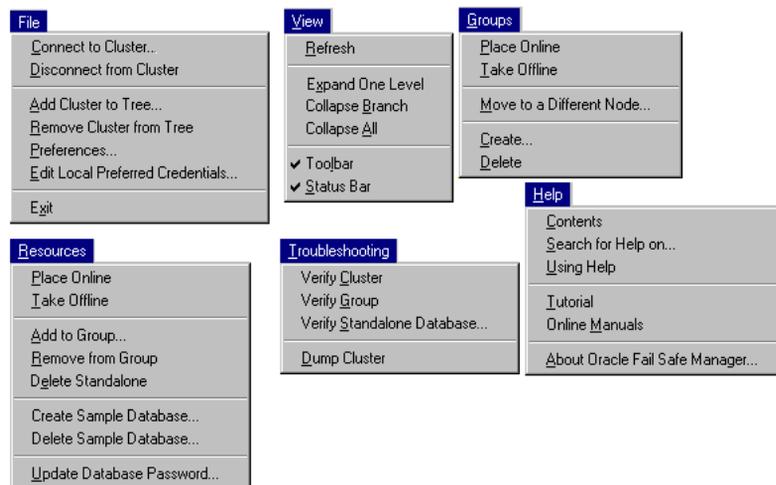


図 1-3 に、Oracle Fail Safe のメニューと各メニューの項目を示します。

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、フェイルオーバーを短時間のノードの再起動として扱います。

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。OCI（Oracle ODBC ドライバを使用する ODBC クライアントを含む）によって開発されたアプリケーションでは、フェイルオーバー後の自動再接続を利用できます。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

Oracle Fail Safe のソリューションは、Microsoft で認証されるすべての Windows クラスタに展開され、MSCS を使用して構成できます。

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タの選択（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）・インターコネクタ。
 - Local Area Network（LAN）または Wide Area Network（WAN）に接続するパブリック・インターコネクタ（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タ**（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。

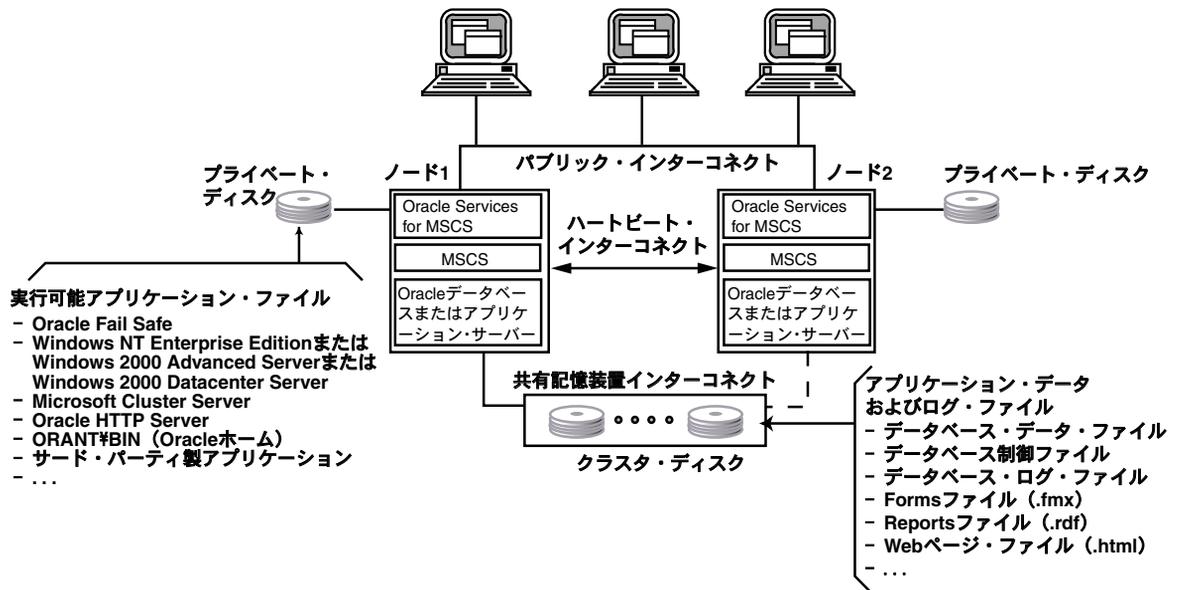
- その他の**冗長周辺機器**（UPS、ネットワーク・カード、ディスク・コントローラなど）。
- ソフトウェア（各ノードにインストール）
 - Windows NT Enterprise Edition、Windows 2000 Advanced Server または Windows 2000 Datacenter Server
 - Oracle Services for MSCS
 - Oracle Fail Safe Manager（1つ以上のクラスタ・ノードまたは1台以上のクライアント・ワークステーション、あるいはその両方にインストール）
 - 次のリソースのうち、可用性を高める1つ以上のリソース
 - * Oracle シングルインスタンス・データベース・サーバー
 - * Oracle Forms Load Balancer Server

- * Oracle Forms Server
- * Oracle Reports Server
- * Oracle HTTP Server
- * Oracle Applications Concurrent Manager
- * Oracle MTS Service (リリース 9.0.1 より前の Oracle データベース・サーバーの場合)
- * Windows 汎用サービスとして構成可能な Oracle またはサード・パーティ製のアプリケーション

これらのコンポーネントに関するリリース別のサポートの詳細は、『Oracle Fail Safe for Windows 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）による停止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、**計画外グループ・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可になった場合、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、**計画的グループ・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つ以上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、Windows 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。

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	参照
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

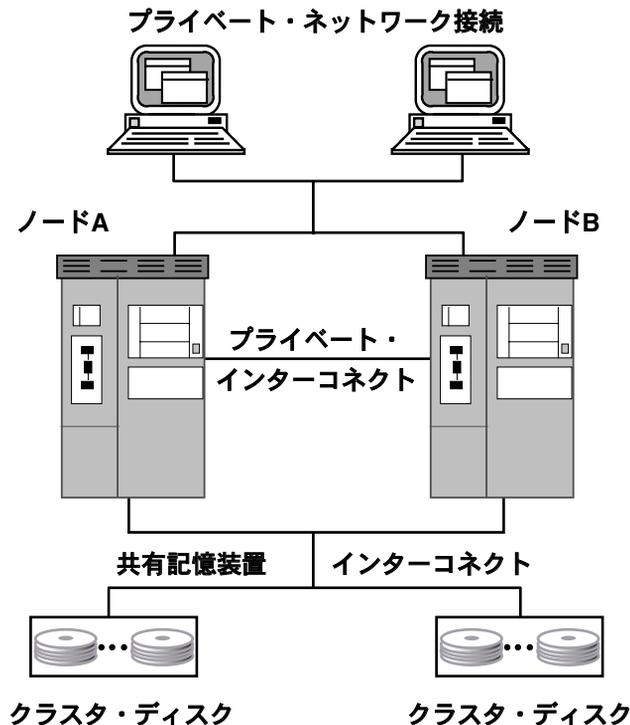
2.1 クラスタ・テクノロジー

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、プライベート・ノード間のネットワーク接続と同様の、パブリックな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可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 Windows 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 では、次の構成がサポートされています。

- 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

これらの構成の詳細は、[第 3 章](#)を参照してください。

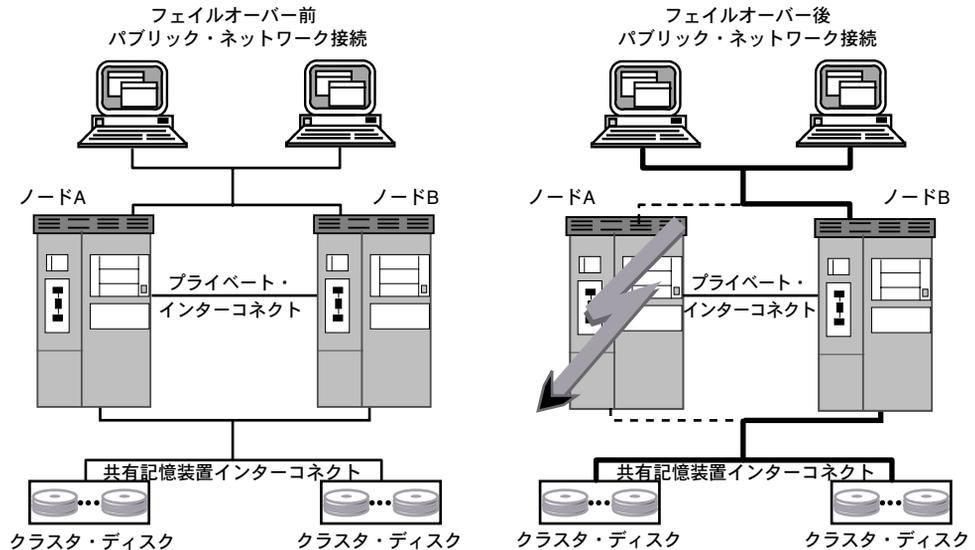
2.1.3 ディスク・レベルの構成

MSCS 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可になった場合に、障害発生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 クォーラム・リソース

クォーラム・リソースでは、停電が発生した場合やメモリー内のデータ損傷が発生した場合に、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。クォーラム・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。クォーラム・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ギン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。

ある時点でクォーラム・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）、クォーラム・リソースの制御を獲得しているノードが、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。

クォーラム・リソースの場所およびクォーラム・ログの最大サイズを表示するには、**Oracle Fail Safe Manager** のツリー・ビューでクラスタを選択し、「クォーラム」タブをクリックします。クォーラム・リソースの場所またはクォーラム・ログの最大サイズを変更するには、**MSCS クラスタ アドミニストレータ**を開き、「ファイル」メニューから「プロパティ」を選択して「クォーラム」タブをクリックします。

2.2 リソース、グループおよび高可用性

あるサーバー・ノードが使用不可になった場合、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およびグループについて説明し、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、次のような特長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 Oracle 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

リソ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も同じグループに含まれるので注意してください。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。

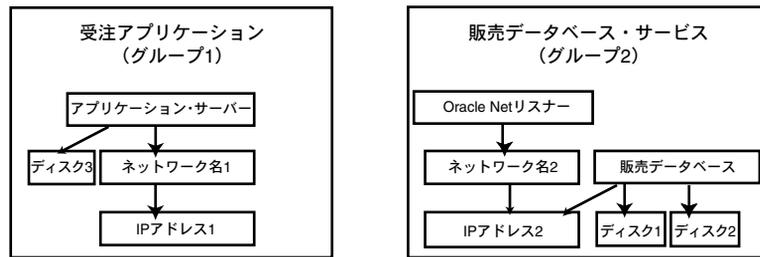
Oracle Fail Safe は、グループを作成し、アプリケーションを実行するために必要なリソースを追加する際に役立ちます。グループ作成の手順ごとの指示は、Oracle Fail Safe のチュートリアルを参照してください。

2.2.3 リソースの依存性

図 2-3 は、受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、受注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、シングルインスタンス・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・**インスタンス** に使用されるシェアド・ナッシング・ディスクが追加され、Oracle 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 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、関連付けられたリソースの独立したセットで構成されます。グループ内の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（および Oracle Net リスナー）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 Oracle Net 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 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 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 データベース・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 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・リソースと通信し、それらを監視できます。

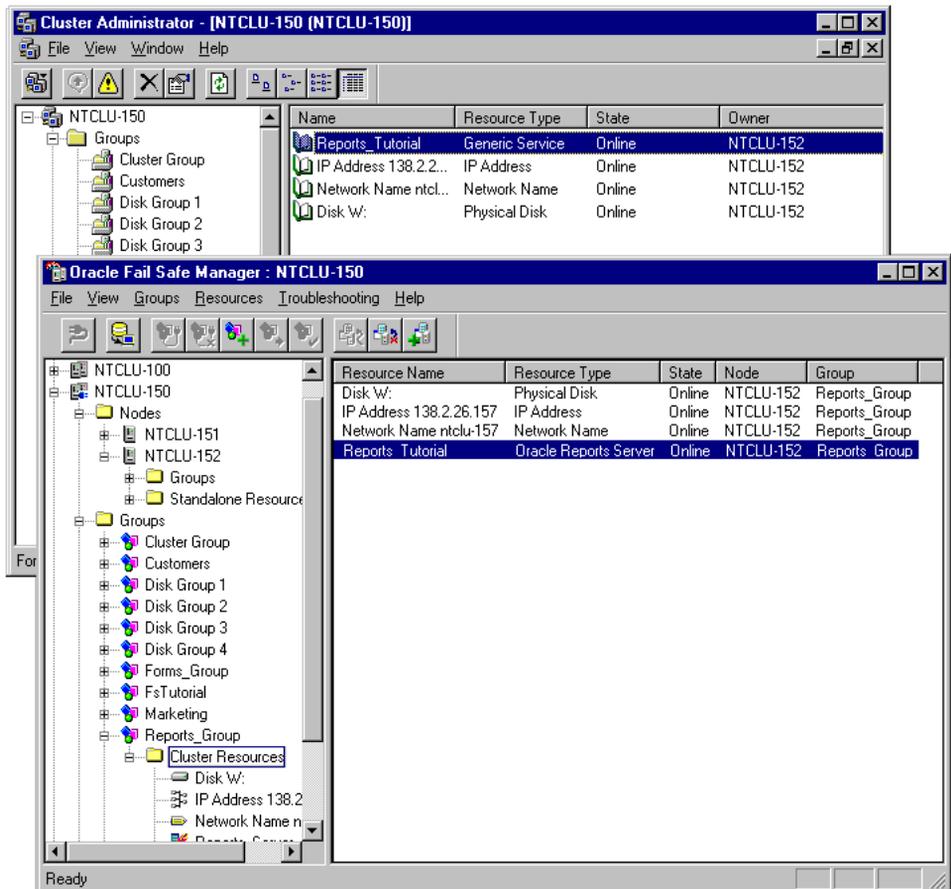
- FsResOdbbs.dll が提供する機能によって、MSCS は Oracle データベースをオンライン化またはオフライン化し、Is Alive ポーリングを介してデータベースの状態をチェックできます。
- FsResOdbbsEx.DLL によって、MSCS クラスタアドミニストレータが Oracle データベース・リソースのプロパティ表示に使用するリソース管理拡張 DLL ファイルが提供されます。

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。

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 HTTP Server および Oracle Forms Load Balancer Server の場合も同様です。

Oracle Fail Safe Manager には、Oracle クラスタ・リソースに関する情報が MSCS よりも多く含まれています。このため、Oracle データベースおよびアプリケーションを構成し管理する場合は Oracle Fail Safe Manager (または FSCMD コマンド) を使用することをお勧めします。

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 の全情報は、『Oracle Fail Safe for Windows 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。

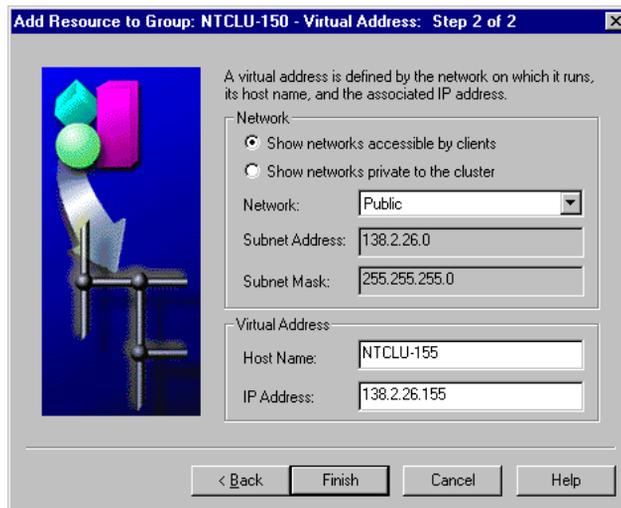
2.3 グループ、仮想アドレスおよび仮想サーバー

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ー・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ー・ノードを知らずに、クライアントは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可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。グループへの仮想アドレスの追加の手順ごとの指示は、Oracle Fail Safe のチュートリアルを参照してください。

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

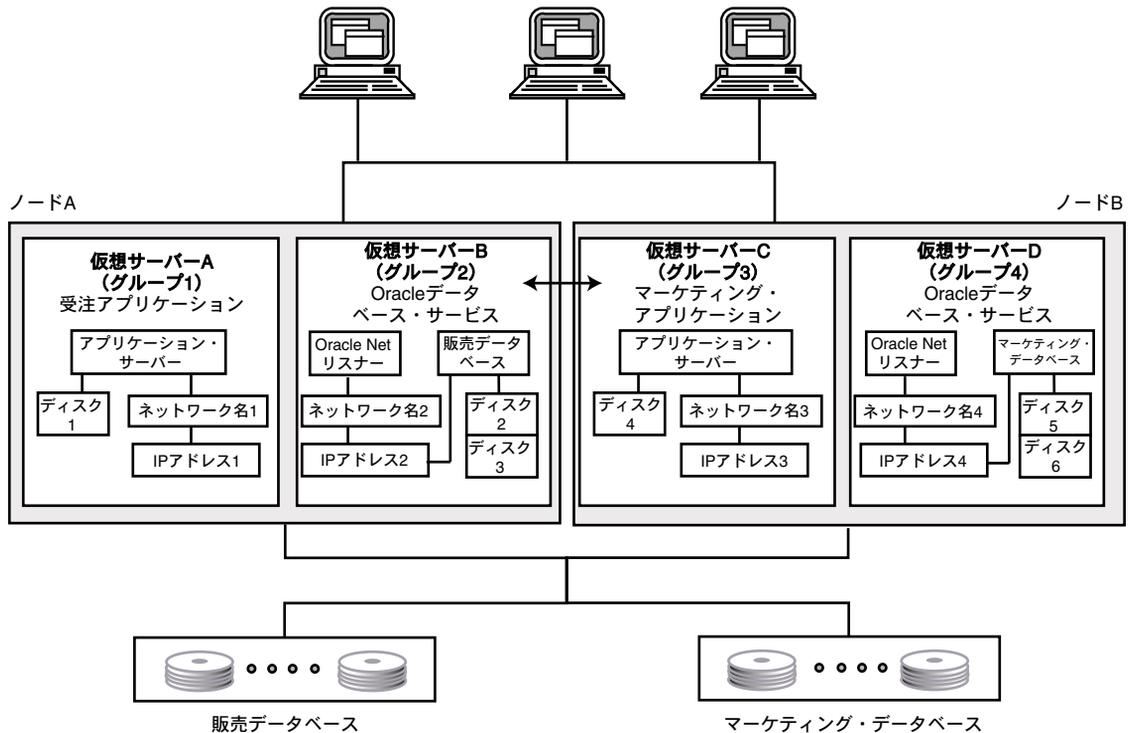


あるグループに仮想アドレスを追加すると、そのグループは**仮想サーバー**になります。クライアント・アクセスのためにグループには少なくとも1つの仮想アドレスが必要ですが、グループに複数の仮想アドレスを割り当てることができます。帯域幅を増やしたり、グループ内のリソースのセキュリティを区分化するために、複数の仮想アドレスを割り当てることもできます。

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ーフェースであり、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かかわらず、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・ノードに1つずつ、クラスタ別名に1つ、そして4つのグループに1つずつで合計7つの IP アドレスが必要です。(1つのグループに対して複数の仮想アドレスを指定できます。詳細は、[4.7 項](#)を参照してください。)

Oracle Fail Safe 環境での IP アドレス割当ての詳細は、『Oracle Fail Safe for Windows インストール・ガイド』を参照してください。

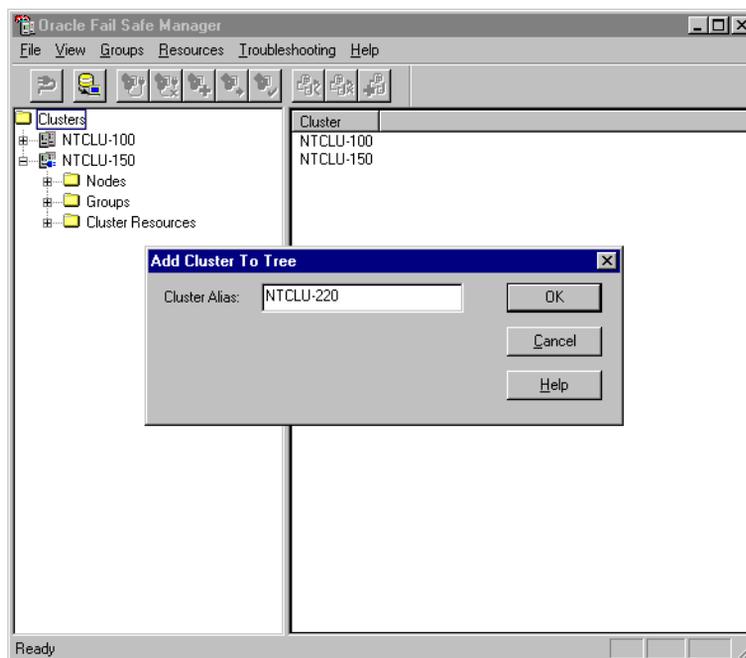
2.5 クラスタ・グループとクラスタ別名

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になります。Oracle Services for MSCS はクラスタ・グループ内にあるリソースです。このためクラスタ・グループの可用性が高められると同時に、すべてのクラスタ・ノードの Oracle Fail Safe の処理が常に Oracle Services for MSCS により調整されます。

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 でツリー・ビューを追加する際には、[図 2-7](#) に示すようにクラスタ別名を指定します。クラスタ別名は、クラスタのどのノードのコンピュータ名とも同じではありません。ツリー・ビューへのクラスタの追加時にクラスタ別名を指定しておく、Oracle Fail Safe Manager ではそのクラスタに接続する際に、Oracle Services for MSCS が稼働している仮想サーバーが確実に使用されます。クラスタ別名は、常にクラスタ・グループ (Oracle Services for MSCS と同じグループ) に属するためです。Oracle Fail Safe Manager のツリー・ビューの追加およびクラスタへの接続の手順ごとの指示は、Oracle Fail Safe のチュートリアルを参照してください。

図 2-7 「ツリーにクラスタを追加」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、クラスタ・リソースとの通信の際にクラスタ別名を使用しません。かわりに、そのリソースを含む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、計画的なものと同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

計画外グループ・フェイルオーバーには2つのタイプがあり、それぞれ次のいずれかの原因によって実行されます。

- 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可状態

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

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。

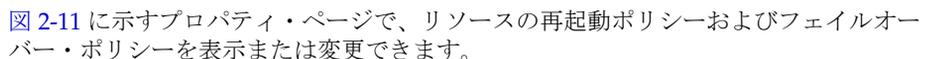
リソース障害を検出するために、クラスタ・ソフトウェアはリソースが稼働状態であるかどうかを（リソース DLL 経由で）定期的に問い合わせます。詳細は、[2.6.3 項](#)を参照してください。

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、リソース障害が起きた場合にグループをフェイルオーバーする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。

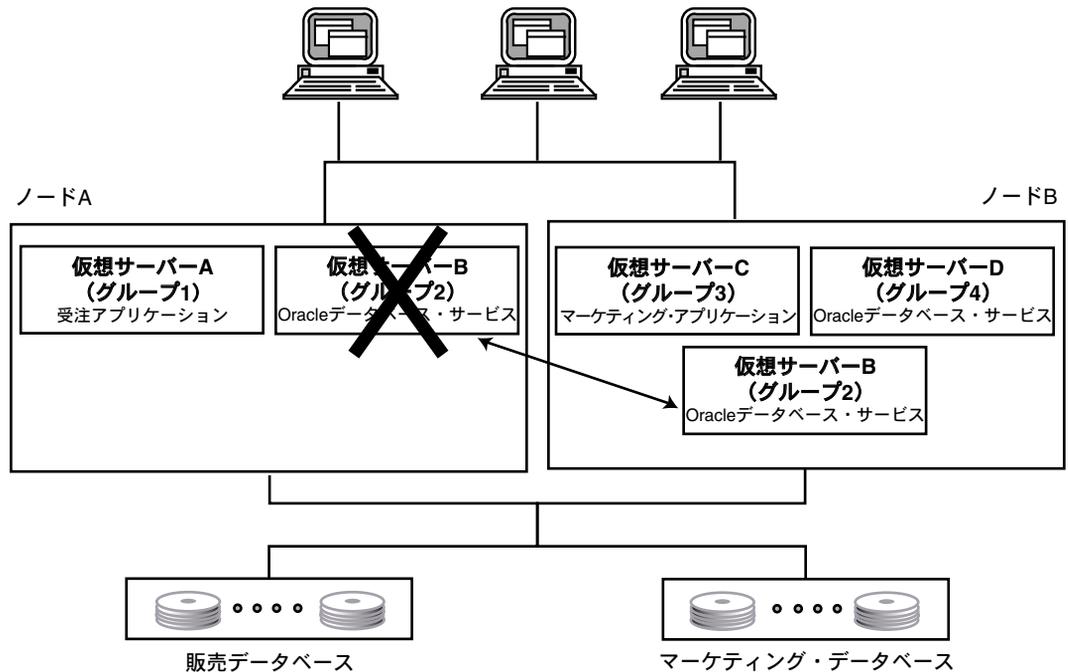
 **図 2-11** に示すプロパティ・ページで、リソースの再起動ポリシーおよびフェイルオーバー・ポリシーを表示または変更できます。

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](#)を、グループの「優先所有者ノード」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るかどうかは、**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可状態による 計画外フェイルオーバー

クラスタ・ノードが使用不可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。

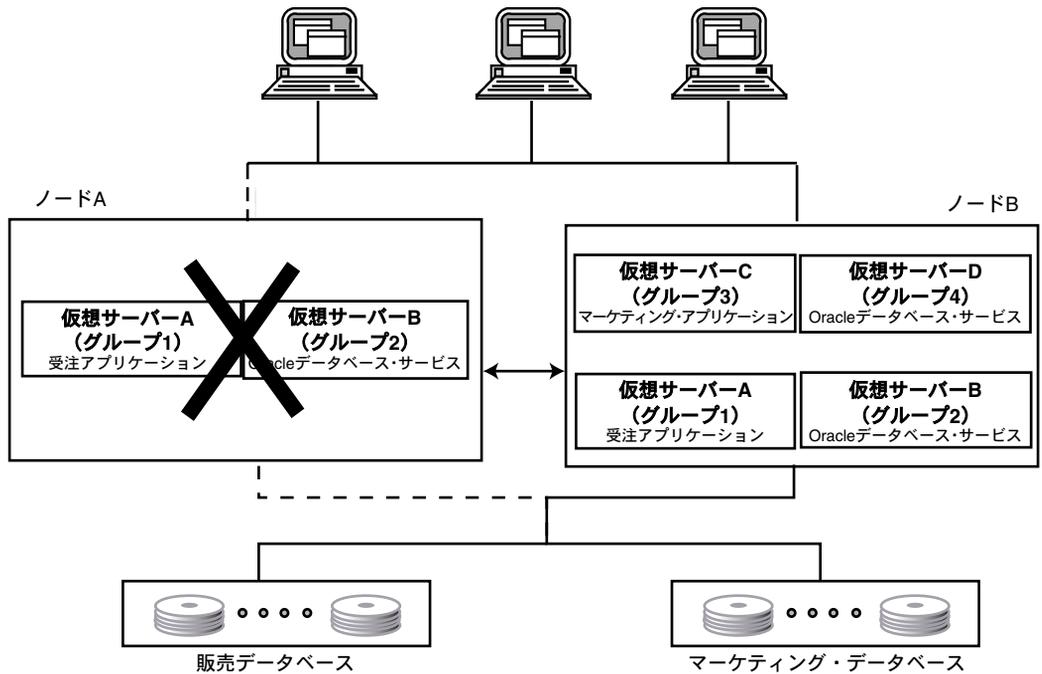
1. クラスタ・ソフトウェアによって、クラスタ・ノードが使用不可になったことが検出されます。

ノードの障害または使用不可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、または使用不可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「優先所有者ノード」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](#) を、グループの「優先所有者ノード」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、[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、前述の手順 4 および 5 ([2.6.1.1 項](#)) と同じです。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、手順は同じになります。

図 2-9 ノードのフェイルオーバー



2.6.1.3 グループの計画的フェイルオーバー

計画的グループ・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・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・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 Services for MSCS 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、透過的アプリケーション・フェイルオーバー（TAF）を構成していない場合、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（TAF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、処理を実行していたノードがオフライン化された場合でもクライアントは別のクラスタ・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するか、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」プロパティ・ページ、グループの「フェイルバック」プロパティ・ページおよびリソースの「ポリシー」プロパティ・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」プロパティ・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・ビューで該当するグループを選択し、「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・ビューで該当するリソースを選択し、「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」プロパティ・ページ

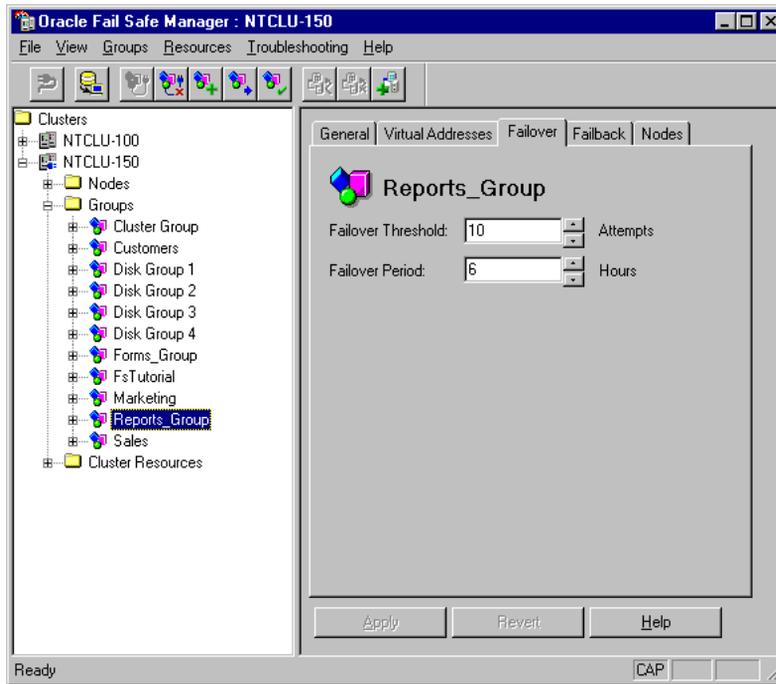
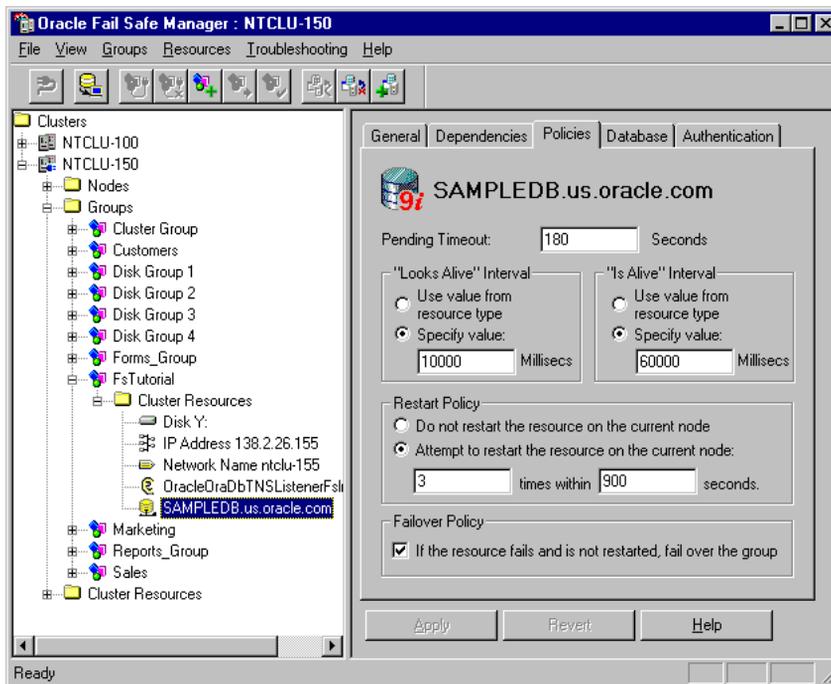


図 2-11 リソースの「ポリシー」プロパティ・ページ



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、消費するシステム・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、消費するシステム・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、「優先所有者ノード」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。（「優先所有者ノード」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。

- 特定のリソースの 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、必須ではありませんが、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されるよう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され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際の詳細は、そのリソースの可用性を高める構成方法について説明している章を参照してください。

- 可用性を高めるためのシングルインスタンス・データベースの構成
- 可用性を高めるための Oracle Forms Load Balancer Server の構成
- 可用性を高めるための Oracle Forms Server の構成
- 可用性を高めるための Oracle Reports Server の構成
- 可用性を高めるための Oracle HTTP Server の構成
- 可用性を高めるための Oracle Applications Concurrent Manager の構成
- 可用性を高めるための Oracle MTS Service の構成
- 可用性を高めるための汎用サービスの構成

- 「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 Oracle データベース・サーバーと、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。ただし、ノード3にはデータベース・インスタンスとそれ以外の作業負荷を両方とも実行するための、十分なメモリがないと仮定します。ノード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、次のいずれかのプロパティ・ページを使用して調整できます。

- リソースの「一般」プロパティ・ページ

リソースの「一般」プロパティ・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。このプロパティ・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」プロパティ・ページ

「ノード」プロパティ・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。このプロパティ・ページを使用して、可能所有者ノードを1つ削除した場合にグループの可能所有者ノードとなるノードが存在しなくなるかどうかを確認できます。[図 2-12](#) は「ノード」プロパティ・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。

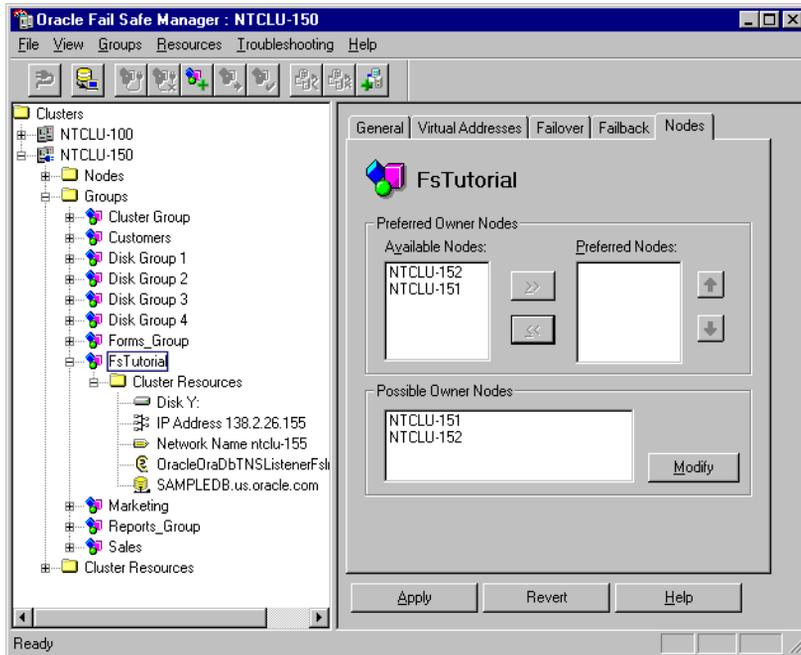
2つのノードから成るクラスタでは、すべてのリソースの「可能所有者ノード」リストに両方のノードが含まれるのが普通です。フェイルオーバー機能を使用するには、少なくとも2つのクラスタ・ノードがリソースの可能所有者ノードである必要があります。

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。

このため、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」プロパティ・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、ノードが使用不可になった場合にも、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時間以内に3回のフェイルオーバーが許容されます。

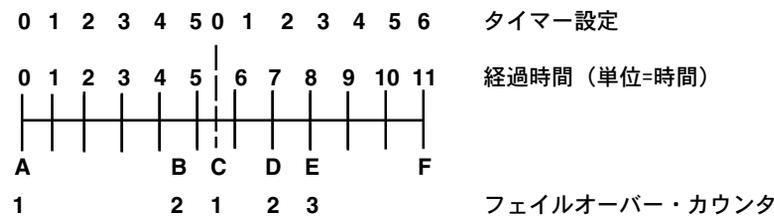
最初のフェイルオーバーが発生すると、フェイルオーバー期間を測定するタイマーは0にリセットされ、フェイルオーバー回数を測定するカウンタは1に設定されます。フェイルオーバー期間を超過した時点では、タイマーは0にリセットされません。そのかわりに、フェイルオーバー期間を超過してから最初のフェイルオーバーが発生した時点で、タイマーは0にリセットされます。

たとえば、前述の例と同様にフェイルオーバー期間が5時間で、フェイルオーバーしきい値が3であるとしします。図 2-13 に示すように、A の時点で最初のグループ・フェイルオーバーが発生すると、タイマーが0に設定されます。2 度目のグループ・フェイルオーバーが4.5 時間後の B の時点で発生し、3 度目のグループ・フェイルオーバーは C の時点で発生すると仮定します。3 度目のフェイルオーバー発生時点（C の時点）ではフェイルオーバー期間を超過しているため、グループ・フェイルオーバーは続行可能で、タイマーは0にリセットされ、フェイルオーバーのカウンタは1にリセットされます。

D の時点（A の時点から7時間が経過し、B の時点から2.5時間が経過した時点）でフェイルオーバーがもう1回発生すると仮定します。フェイルオーバーは停止するでしょうか。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。

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



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」プロパティ・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」プロパティ・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・インスタンスの再起動が試行されます。データベース・インスタンス再起動の試行が、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・インスタンスの再起動を試行しますが、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・インスタンス再起動の試行と **Customers** グループのフェイルオーバーは、データベース・インスタンスが再起動するか、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 回を超えるまで続行されます。データベース・インスタンスが再起動できず、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「優先所有者ノード」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「優先所有者ノード」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「優先所有者ノード」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が使用不可になると、グループは2番目に一覧されているノード4で実行されます。ノード1もノード4も使用不可の場合、次に一覧されているノード3でグループが実行されます。「優先所有者ノード」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かぎ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「優先所有者ノード」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、リスト内のノードのうち、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。

グループの「優先所有者ノード」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ます。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「優先所有者ノード」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 - はい	ノード 1 - はい	ノード 1 - はい
ノード 2 - はい	ノード 2 - いいえ	ノード 2 - はい
ノード 3 - はい	ノード 3 - はい	ノード 3 - はい
ノード 4 - はい	ノード 4 - はい	ノード 4 - はい

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「優先所有者ノード」リストがクラスタ・ソフトウェアで使用されます。Test_Group の「優先所有者ノード」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 がいずれも使用不可の場合、Test_Group

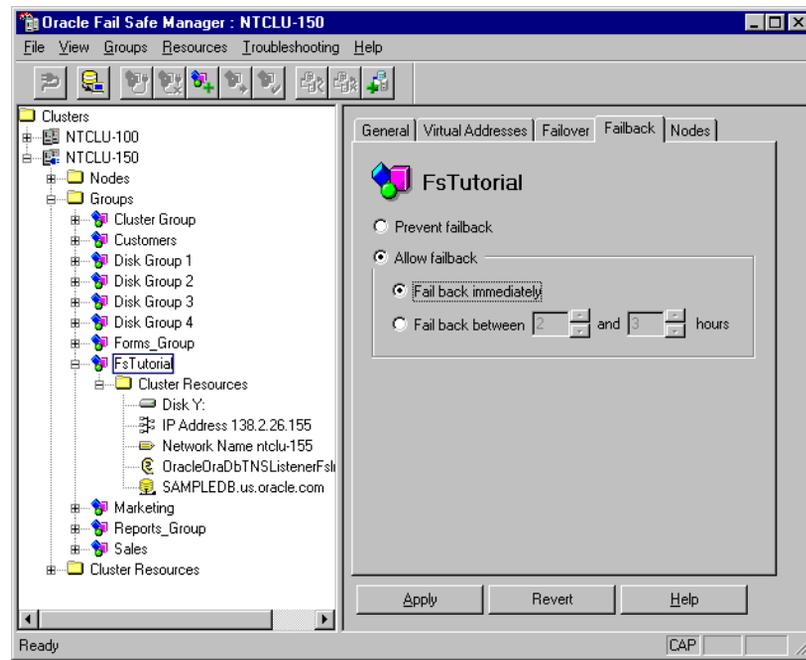
はノード 1 にフェイルオーバーされます。もしノード 1、3 および 4 がすべて使用不可であれば、Test_Group はノード 2 にフェイルオーバーされます。ただし、ノード 2 は Test_Group 内のすべてのリソースの可能所有者ではないので、Test_Group はノード 2 で障害発生状態のままになります。

2.7 フェイルバック

フェイルバックは、優先所有者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・ノードから優先所有者ノードに自動的に戻るプロセスです。**優先所有者ノード**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。

グループをフェイルオーバー・ノードから優先所有者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**フェイルバック・ポリシー**を設定できます。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プロパティ・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」ポリシー・プロパティ・ページ



2.7.1 グループ・フェイルバックと「優先所有者ノード」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「優先所有者ノード」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。以前に使用不可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「優先所有者ノード」リストを読み込み、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所有者ノードであるかどうかを判断します。「優先所有者ノード」リスト中で、再オンライン化されたノードの順位が、現在グループが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「優先所有者ノード」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「優先所有者ノード」リストを読み込み、**My_Group** の優先所有者ノードがノード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も使用不可なため)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。「優先所有者ノード」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しないように注意してください。ほとんどのアプリケーションで、「優先所有者ノード」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3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3に移動したためです)。ノード4が再起動されると、ノード1(**My_Group** の「優先所有者ノード」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4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、オンライン化されたそのノードが、「優先所有者ノード」リストの中で、現在各グループが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。

「優先所有者ノード」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。TAF が構成されているクライアント・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再起動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。

TAF 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
障害耐久力と高可用性	3.2 項
クライアントとアプリケーションの統合	3.3 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に必要なすべての**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 Services for MSCS 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 Oracle HTTP Server 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 Services for MSCS、Oracle Forms Server、Oracle Reports Server および Oracle データベース・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。現在、ノード 2 では何も実行されていません。フェイルオーバー時の際、ノード 2 はノード 1 の作業負荷を引き継ぎます。

図 3-1 2 ノードのアクティブ/パッシブ (スタンバイ) 構成

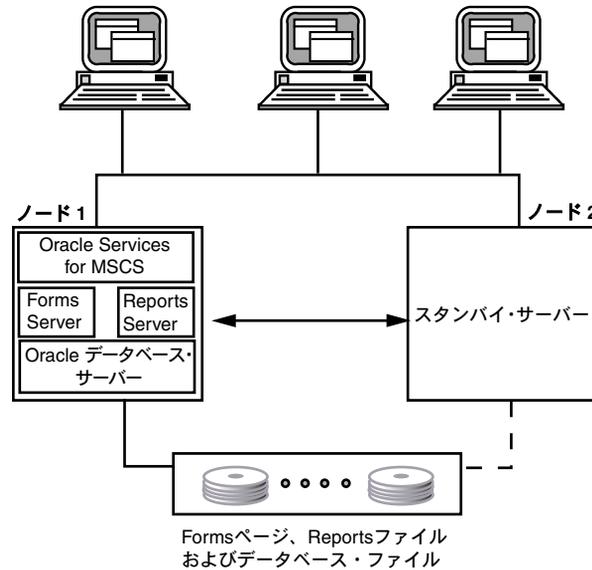
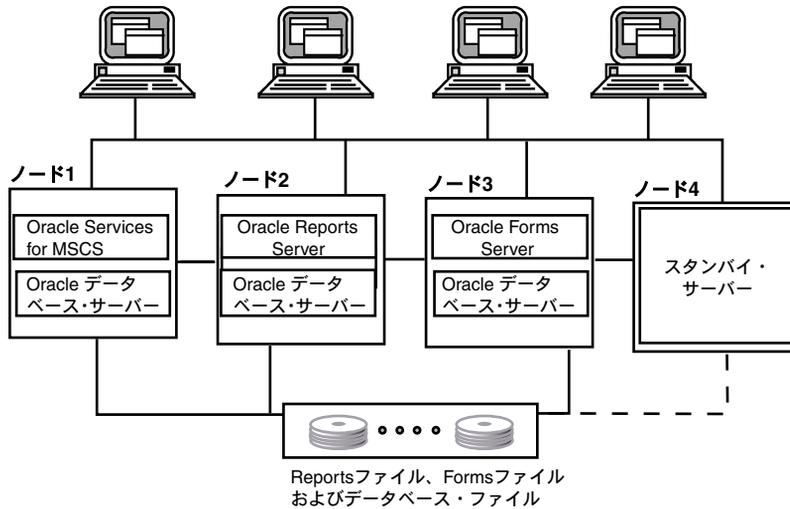


図 3-2 は、ノード 1 に Oracle Services for MSCS および Oracle データベース・サーバー、ノード 2 に Oracle Reports Server および Oracle データベース・サーバー、ノード 3 に Oracle Forms Server および Oracle データベース・サーバーを置き、ノード 4 を **スタンバイ・ノード** とした 4 ノードの構成です。現在、ノード 4 では何も実行されていません。フェイルオーバーの際、ノード 4 はフェイルオーバーの作業負荷を引き継ぎます。

図 3-2 4 ノードのアクティブ/パッシブ（スタンバイ）構成



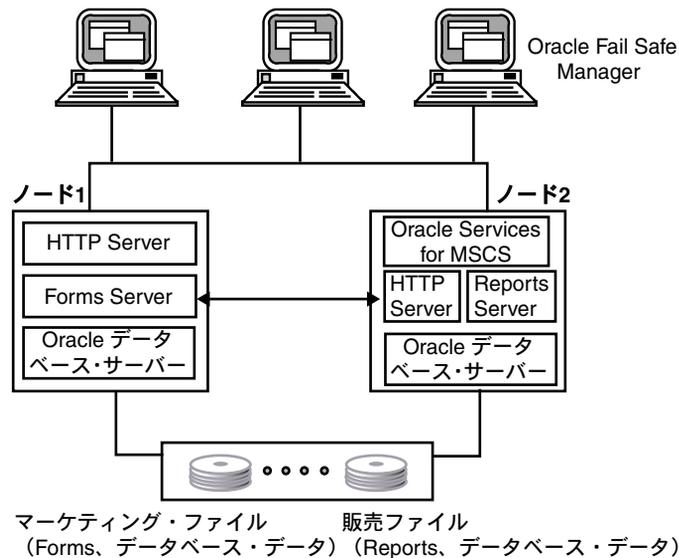
パッシブのスタンバイ・ノードにはそれ自体の作業負荷がないため、アクティブ/パッシブ構成は最高速のフェイルオーバー構成です。

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・ノードに Oracle データベース・サーバーを 1 つずつ置いた 2 ノードのアクティブ / アクティブ構成を示します。その他に、ノード 1 で Oracle HTTP Server および Oracle Forms Server、ノード 2 で Oracle Services for MSCS、Oracle HTTP Server および Oracle Reports Server が稼働しています。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・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、停止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）75%～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・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 Oracle またはサード・パーティ製のアプリケーション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、クラスター・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

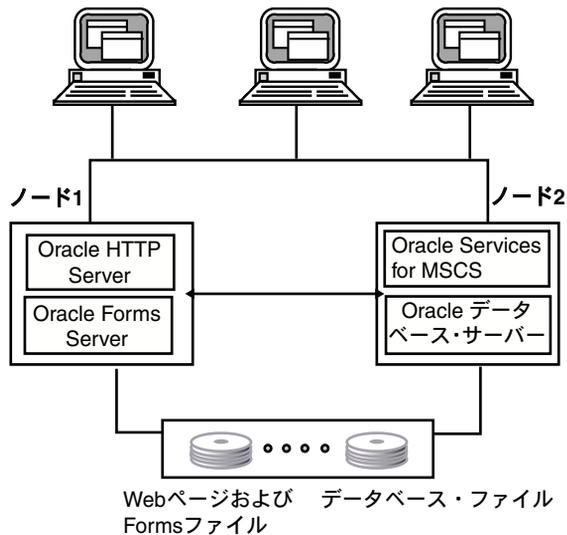


図 3-4 では、ノード 1 で Oracle HTTP Server および Oracle Forms Server が稼働し、ノード 2 で Oracle Services for MSCS および Oracle データベース・サーバーが稼働しています。ノード 1 が所有するクラスター・ディスクには Web ページと Oracle Forms のファイルが格納され、ノード 2 が所有するクラスター・ディスクにはデータベース・ファイルが格納されています。

プライベート・インターコネクト (ハートビート・インターコネクト) が広い帯域幅を持っている場合、Forms Server では、データベース・インスタン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、Oracle HTTP Server はプライベート・ネットワークをデータベース・インスタン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、アプリケーション層とデータベース層のコンポーネントは別々のクラスター・ノードで稼働するように構成されます。また、複数のクラスターおよびシステム間で稼働するようにも構成できます。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。

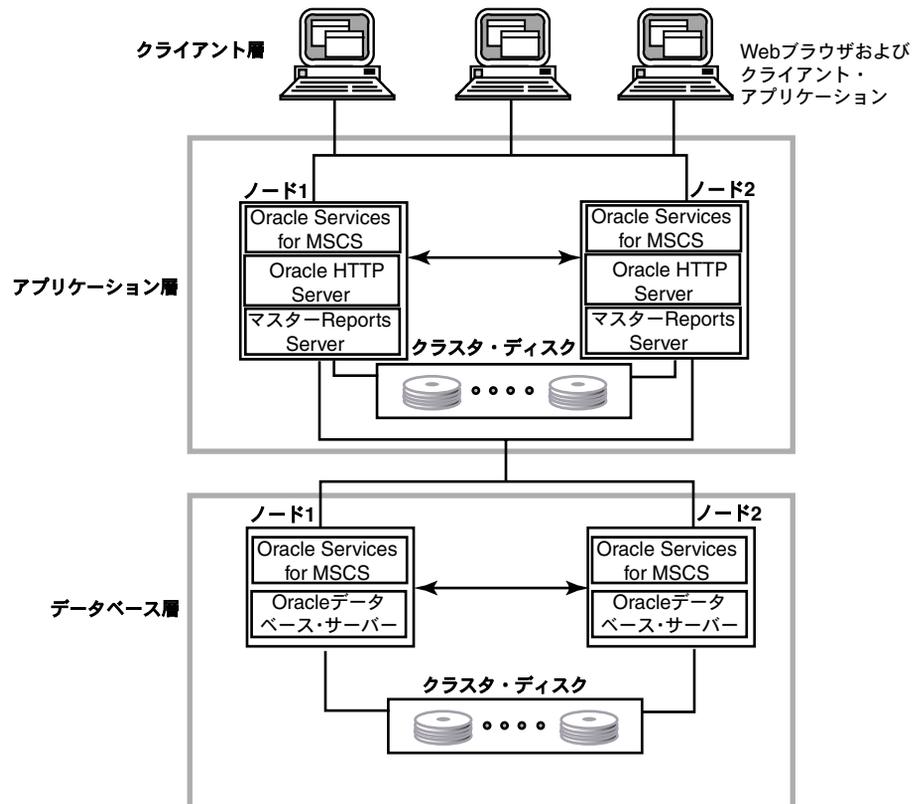
- **マスター Oracle Reports Server**

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・ノードに Oracle データベース・サーバーおよび Oracle Services for MSCS 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・ノードに Oracle Services for MSCS、Oracle HTTP Server 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。次に例を示します。
 - 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 または Windows NT システムで稼働しているバックエンドの Oracle Real Application Clusters 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 システム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。

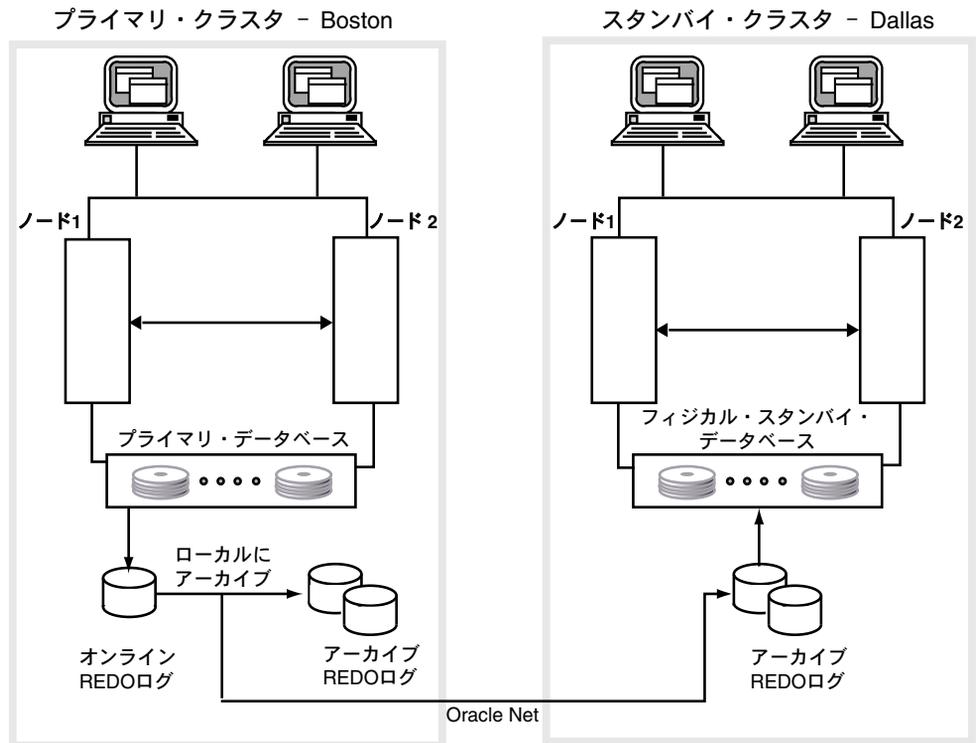
3.2 障害耐久力と高可用性

Oracle Fail Safe でシングルインスタンスの Oracle データベースに対する高可用性が提供される一方、Oracle Data Guard では障害耐久力が提供されます。たとえば、Oracle Fail Safe では、特定のシステムに対するほぼ連続的な高可用性は保証されますが、そのシステムが存在するサイトを不能にするような障害からは保護されません。同様に、Oracle Data Guard は優れた障害時リカバリ機能を備えていますが、プライマリ・サイトから物理的に離れたサイトへの切替え操作に必要な時間は数分から数時間におよびます。Oracle Fail Safe と Oracle Data Guard を組み合わせることによって、データベースは高可用性と障害耐久力を得ることができます。

個別の Windows クラスタに配置された 2 つのデータベース（1 つは Boston で、もう 1 つは Dallas）による構成例を、[図 3-6](#) に示します。Oracle Data Guard を使用して、1 つのデータベースをプライマリ・データベース、もう一方のデータベースをフィジカル・スタンバイ・データベースとして構成します。次に、Oracle Fail Safe を使用して、各データベースを可用性を高めるように構成します。Boston のプライマリ・データベースは、クラスタ上で可用性を高めるように構成されます。プライマリ・データベースのオンライン REDO ログ・ファイルはローカルに、および Oracle Net を使用してリモートでフィジカル・スタンバイ・データベースにアーカイブされます。また、Dallas のクラスタ上で可用性を高めるように構成されます。（必要に応じて、ロジカル・スタンバイ・データベースを使用できます。）

多くの本番環境では、同期および非同期ログ転送の組合せで複数のスタンバイ・データベースが配置されます（たとえば、ローカルのスタンバイ・データベースは同期ログ転送によってプライマリ・データベースとともに最新の状態に保たれ、非同期ログ転送を使用する複数のリモート・スタンバイ・データベースはプライマリ・データベースとは異なる遅延時間でメンテナンスされます）。

図 3-6 Oracle Data Guard を使用した障害耐久力を高めるデータベース構成



3.3 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7章から第14章に、クライアントとアプリケーションの統合方法を説明した項があります。第7章に、データベ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方法を説明しています。

高い可用性を実現するための管理

Oracle Fail Safe には、Windows 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手順ごとの指示、およびグループに属しているこれらのリソースの管理に関する詳細は、このマニュアルの第 7 章から第 14 章まで、ならびに 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。

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 for Windows インストレーション・ガイド』で説明していません。）
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 Oracle Net ファイル（tnsnames.ora ファイルなど）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、これ以外に手順や注意事項を追加する場合があります。

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 Services for MSCS 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。

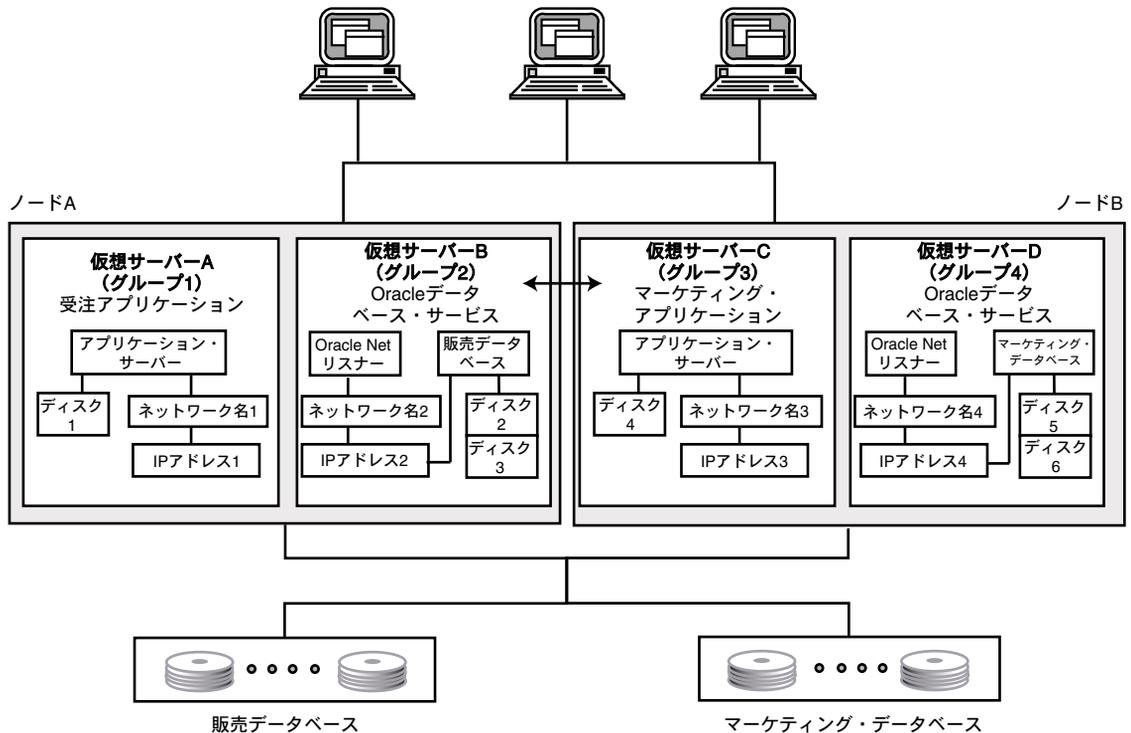
1. 仮想アドレスを使用したデータベースへのアクセスを構成します。
 - a. Oracle Net を、データベースの「可能所有者ノード」リストにあるすべてのノードで仮想アドレスまたは関連付けられたアドレスを使用できるように構成します。（2つのノードから成るクラスタでは、両方のクラスタ・ノードが可能所有者ノードになります。3つ以上のノードから成るクラスタでは、「リソースをグループに追加」ウィザードのステップとして、リソースの可能所有者ノードを指定するように求められます。）
 - b. ネットワーク構成情報を、「可能所有者ノード」リストのすべてのノードに複製します。
2. 次のようにデータベースを構成します。
 - 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
 - * Oracle 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、1回に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。

- 「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、優先所有者ノードで再度オンライン化します。「優先所有者ノード」リストでノードが選択されていない場合、グループは構成プロセスの過程で最後に指定したフェイルオーバー先のノードに残ります。

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 エントリ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・システム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 Services for MSCS	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 Load Balancer Server、 Oracle Forms 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)	8.4 項、9.4 項 および 10.3.3.6 項

表 4-1 許可と権限 (続き)

サービス	必要な権限	参照
Oracle Applications Concurrent Manager	Windows 管理者アカウント	12.4 項
Oracle HTTP Server	なし	適用なし
Oracle MTS Service	<p>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。</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 	13.4 項
汎用サービス	<p>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。</p>	14.4 項

4.3.1 Oracle Services for MSCS

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 Services for MSCS 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・アカウント以外では稼働しない Windows サービスとして稼働します。Oracle Services for MSCS 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 for Windows インストール・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 Services for MSCS 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も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セキュリティ設定ツールがあります。

4.3.1.1 Oracle Fail Safe のセキュリティ設定ツールを使用したアカウント更新

Oracle Fail Safe にはセキュリティ設定ツールがあり、これを使用して Oracle Services for MSCS が稼働するアカウントの情報を更新できます。Oracle Services for MSCS のインストール時に、Oracle Services for MSCS のセキュリティ設定ツールもインストールされます。

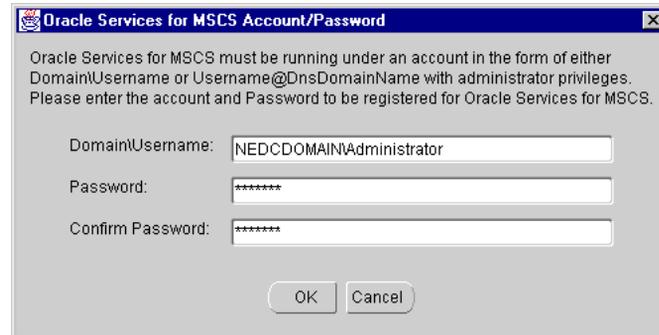
Windows のタスクバーから Oracle Services for MSCS のセキュリティ設定ツールにアクセスするには、クラスタ・ノードから次のように選択します。

「スタート」 → 「プログラム」 → 「<Oracle_Home>」 → 「Oracle Services for MSCS 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 Services for MSCS のセキュリティ設定ツールを使用し、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 NEDCDOMAIN にユーザー・アカウント Administrator を設定する例を示します。

図 4-2 Oracle Services for MSCS の Windows ユーザー・アカウント設定



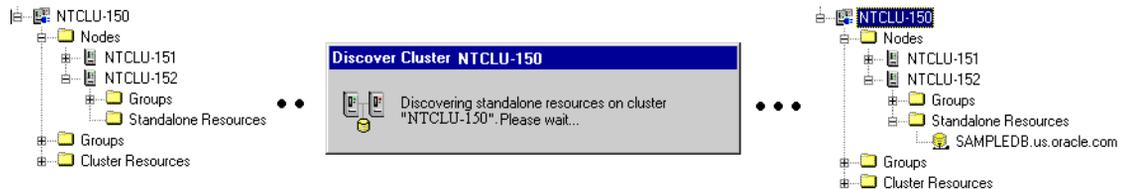
4.3.2 Oracle Fail Safe Manager

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 Services for MSCS 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 は、Oracle Fail Safe Manager 内での検出例です。第 7 章から第 14 章には、Oracle Fail Safe で可用性を高め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関する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 Services for MSCS 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 Services for MSCS の 1 つのバージョンにかぎられます。
- Oracle Fail Safe Manager の最新リリースを使用すると複数のクラスタを管理できます。Oracle Fail Safe Manager リリース 3.3.1 は、Oracle Fail Safe リリース 3.3.1、3.2.1 および 3.1.2 とともに使用できます。リリース 3.1.2 より前の Oracle Fail Safe の場合は、リリースが一致する Oracle Fail Safe Manager を使用する必要があります。
- 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せん。

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、最新リリースの 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・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と、Oracle Net のリスナー・リソースもグループに追加されます。（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。（Oracle Fail Safe Manager で、「リソース」メニューから「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、[7.3.3.2 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、どのクラスター・ノードがそのクラスターのホストになっているかにかかわらず、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 for Windows インストール・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。

このため、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手段として、Oracle Fail Safe Manager に加えて、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。

5.1 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。

5.1.1 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。

5.1.2 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、Oracle Fail Safe Manager がインストールされているシステムで実行できます。(FSCMD ソフトウェアは Oracle Fail Safe Manager のコンポーネントです。)

FSCMD コマンドを実行する場合、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。

5.1.3 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
DISABLEISALIVE	次のいずれかの操作を行うまで、指定したデータベース・インスタンスの Is Alive ポーリングおよび Looks Alive ポーリングを使用不可にします。 <ul style="list-style-type: none"> ■ FSCMD ENABLEISALIVE コマンドを使用して Is Alive ポーリングを明示的に使用可能にすること ■ インスタンスをオンライン化すること 「グループの検証」コマンドでは、あるインスタンスに対して Is Alive ポーリングが使用不可にされていると警告が発行されますが、Is Alive ポーリングを再度使用可能にはできません。
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。
ENABLEISALIVE	指定したデータベース・インスタンスの Is Alive ポーリングおよび Looks Alive ポーリングが FSCMD DISABLEISALIVE コマンドにより使用不可にされている場合、それらを使用可能にします。
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、シングルインスタンス・データベースの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。

5.1.4 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かぎり使用します。

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かぎり使用します。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・インスタンスを強制終了することにより、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、もしくはそのアプリケーションの1つが不適切に機能していて、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>
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つが不適切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`

すべてのクラスター・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。

5.1.5 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 Services for MSCS 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。

5.1.6 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
```

5.1.7 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. Disable Is Alive polling for the database resource.
```

```
fscmd disableisalive db.world /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. Reenable Is Alive polling for the database resource.
```

```
fscmd enableisalive db.world /cluster=ntclu-140
```

```
REM The backup operation is complete.
```

6

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、この章で説明する情報を示します。

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 Services for MSCS for Windows 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などを含む)、クラスタ・ネットワーク構成およびクラスタ・リソース DLL 登録を検証します。	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します。	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します。	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。

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。

- Oracle ソフトウェアがインストールされている Oracle ホームの名前が、すべてのクラスタ・ノードに共通であること

たとえば、あるクラスタ・ノードの Oracle Fail Safe ソフトウェアの Oracle ホームの名前が OFS である場合、Oracle Fail Safe がインストールされているクラスタ内のすべてのノードで OFS という Oracle ホーム名が使用されている必要があります。同じように、あるクラスタ・ノードの Oracle データベース・サーバーの Oracle ホームの名前が OfsDb である場合、Oracle データベース・サーバー・ソフトウェアがインストールされているクラスタ内のすべてのノードで OfsDb という Oracle ホーム名が使用されている必要があります。

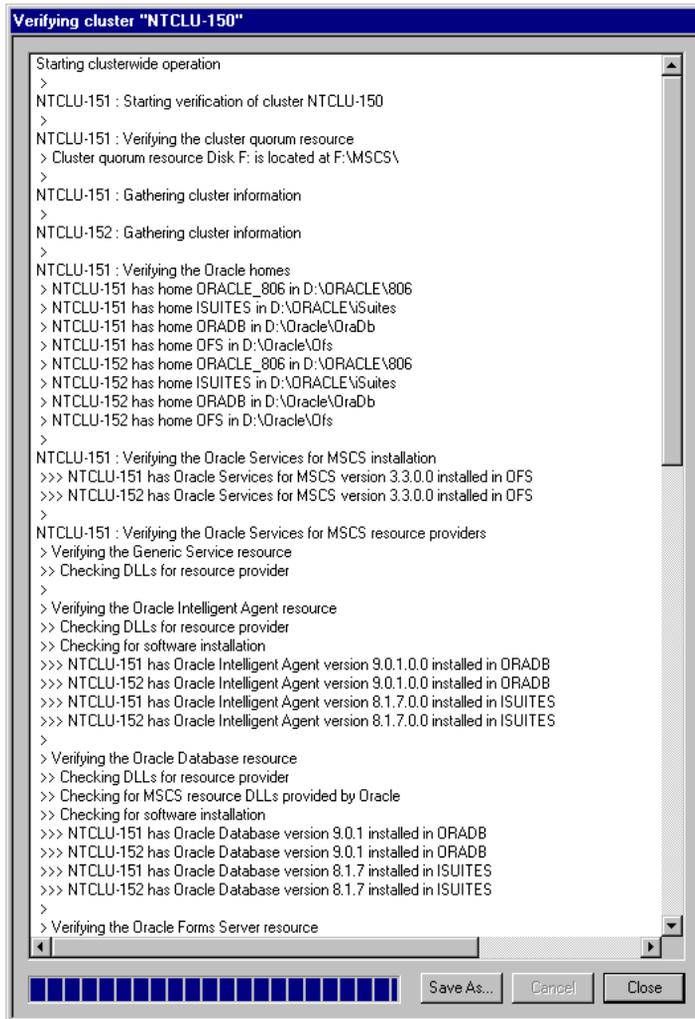
- Oracle Services for MSCS のリリースがすべてのノード上で同一であること
- リソース・プロバイダ（コンポーネント）が各リソースの少なくとも 2 つの可能所有者であるノードでまったく同じに構成されていること
- ホスト名と IP アドレスのマッピングがクラスタ内のすべてのノード間で矛盾なく解決していること

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。

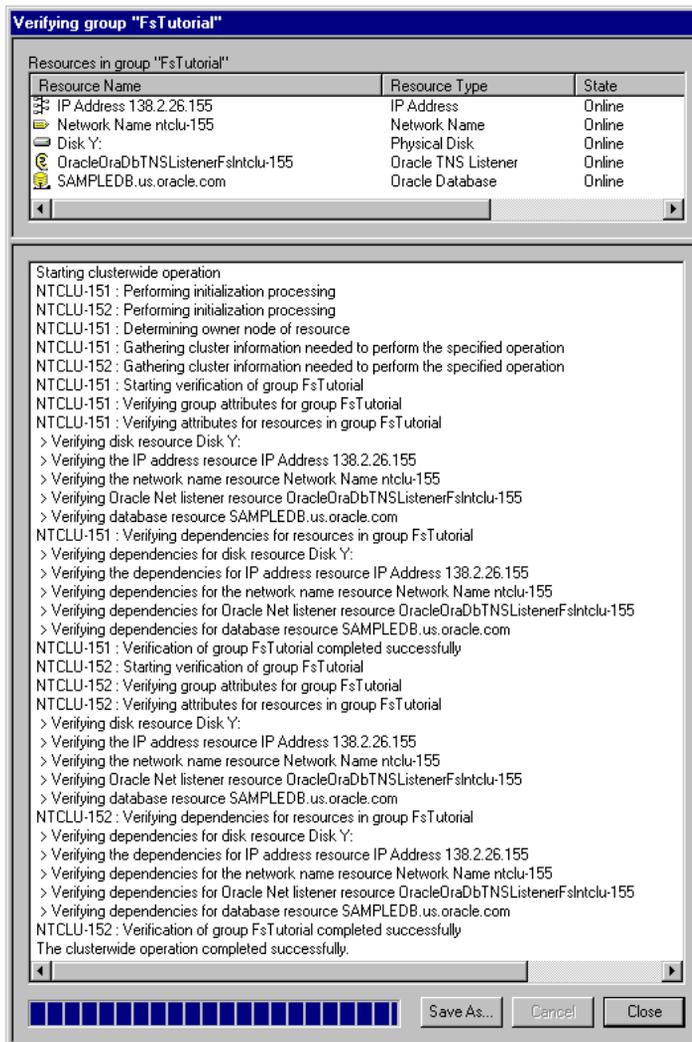
「トラブルシューティング」→「グループの検証」

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。

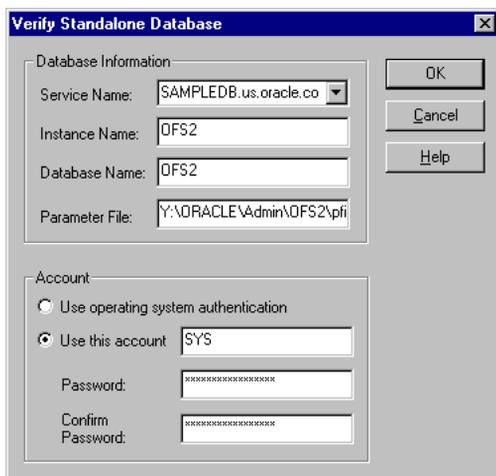
グループにスタンドアロン・データベースを追加する前に、そのデータベースに対して「スタンドアロン・データベースの検証」コマンドを実行してください。また、スタンドアロン・データベースへのアクセスに問題が生じた場合に、いつでもコマンドを使用できます。ただし、検証操作中、データベースは Oracle Fail Safe により停止および再起動されます。

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 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 for Windows インストール・ガイド』に記載されています。

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・データベースの初期化パラメータ・ファイルのドライブ文字、パス名およびファイル名
- 「アカウント」領域に、Oracle Fail Safe でデータベースにアクセスするためのアカウント

この情報を Oracle Fail Safe では次の場合に使用します。

- Oracle Net によるクラスタワイドの問題の修正
- 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
- 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、Oracle 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ます。「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対して行われた変更の記録を保持するために定期的に実行（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撮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（クォーラム・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

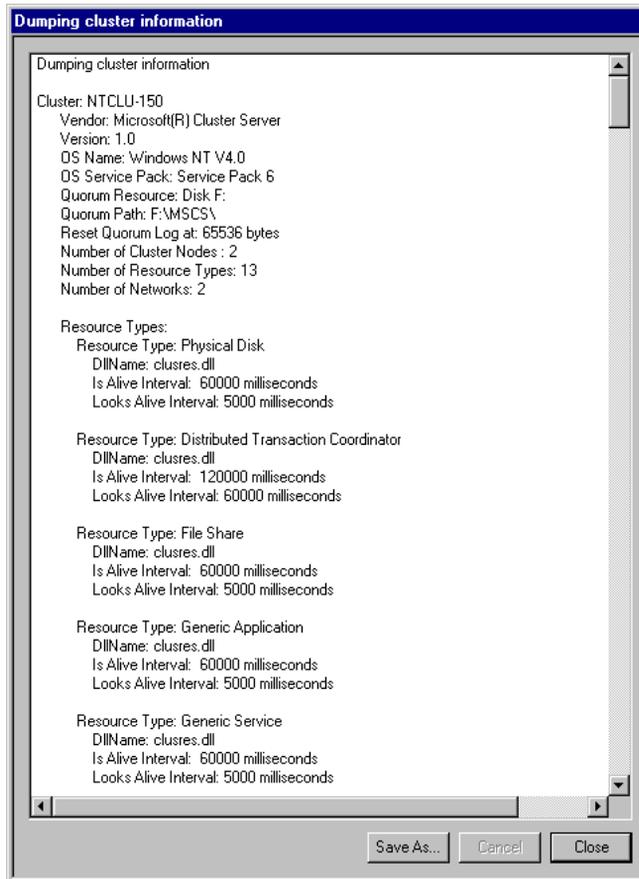
オプションで、「クラスタのダンプ」データをファイルに保存できます。「別名保存」をクリックしてください。

「クラスタのダンプ」コマンドを発行するには、Oracle Fail Safe Manager のツリー・ビューからクラスタを選び、次のように選択します。

「トラブルシューティング」→「クラスタのダンプ」

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

図 6-6 「クラスタのダンプ」の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する情報は、可用性の高いコンポーネントの構成方法について説明する、第7章から第14章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、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 Services for MSCS により、問題を識別するイベントのログが取られます。

可用性を高めるためのシングルインスタンス・データベースの構成

Oracle Fail Safe により、MSCS を使用して構成された Windows クラスタで稼働しているシングルインスタンス Oracle データベース (Standard Edition および Enterprise Edition の両方) の可用性が向上します。

シングルインスタンスの 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。こうしたフェイルオーバー・イベントの発生後、アプリケーションは TAF によって自動的にデータベースに再接続できるため、フェイルオーバーがユーザーの作業に悪影響を与え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・シングルインスタンス・データベースの検出	7.1 項
スタンドアロン・シングルインスタンス・データベースのための Oracle Net 構成	7.2 項
シングルインスタンス Oracle データベースのグループへの追加	7.3 項
Oracle Net リスナー・リソースの作成および構成	7.4 項
シングルインスタンス・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
シングルインスタンス・データベース・リカバリの最適化	7.8 項
シングルインスタンス・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

項目	参照
透過的アプリケーション・フェイルオーバー (TAF) の構成	7.10 項
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・シングルインスタンス・データベースの検出

Oracle Services for MSCS では、スタンドアロン・シングルインスタンス・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・シングルインスタンス・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・シングルインスタンス・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・シングルインスタンス・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・シングルインスタンス・データベース

その他のスタンドアロン・シングルインスタンス・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・シングルインスタンス・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・シングルインスタンス・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・シングルインスタンス・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。)

あるインスタンスに対して有効なネット・サービス名エントリがない場合、インスタンス名は Oracle Fail Safe Manager 内のデータベース・インスタンスを表す名前になります。

Oracle Fail Safe Manager のツリー・ビューにスタンドアロン・シングルインスタンス・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。

7.2 スタンドアロン・シングルインスタンス・データベースのための Oracle Net 構成

次の項では、スタンドアロン・シングルインスタンス・データベースのための Oracle Net の構成について簡単に説明します。

7.2.1 DBCA により作成したデータベースのための Oracle Net 構成の更新

Database Configuration Assistant (DBCA) を使用してスタンドアロン・シングルインスタンス・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 Oracle Net 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 Oracle Net 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）に新規シングルインスタンス・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。

7.2.2 リスナーの定義

Oracle8i リリース 8.1.5 から、Oracle Net リスナーの動作が変更されています。Oracle8i リリース 8.1.5 以上のリスナーの定義でシステムの**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、後でグループにデータベースを追加する際にリスナーが競合する原因となり、Oracle Fail Safe とともに構成されたリスナーは仮想アドレスでリスニングするよう定義されます。

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。シングルインスタンス・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 以上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-152)
      (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=138.2.26.152)
      (PORT=1521)
    )
```

7.2.3 SID リスト・エントリおよび Oracle8i または Oracle9i ソフトウェアへのアップグレード

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 または Oracle9i にアップグレードする場合、Oracle8i または Oracle9i リスナー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 リストのエントリを Oracle8i または Oracle9i リスナーに移す必要があります。

1. Oracle8i または Oracle9i 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。次に例を示します。
 - Oracle リリース 7.3.4、Oracle8i および Oracle9i の listener.ora ファイルは、<Oracle_Home>%NETWORK%ADMIN ディレクトリにあります。
 - Oracle8 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 または Oracle9i 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 ファイルでの更新後の 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. 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」を使用して「サービス」ウィンドウにアクセスしてください。)

5. Oracle8i または Oracle9i リスナーを起動します。

Oracle8i または Oracle9i 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。

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

7.2.4 複数のリスナーを伴うノードでの Oracle 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 サーバーが（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「サービス」の「スタートアップ」リストで、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際、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り、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・シングルインスタンス・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。

7.2.5 共有サーバーの構成とスタンドアロンの Oracle8i または Oracle9i データベース

データベースが可用性が高まるように構成されている場合、Oracle Fail Safe ではデフォルト・リスナーの調整を行います。この調整は、スタンドアロン・データベースを含むすべてのデータベースの Oracle Net 構成に影響を与えます。このため、クラスタ内のデータベースが可用性が高まるように構成されている場合、Oracle Fail Safe 環境のすべてのスタンドアロン・データベースの Oracle Net 構成を調整する必要があります。

スタンドアロン・シングルインスタンスの Oracle8i および Oracle9i データベースの共有サーバーの構成がデフォルト・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。

次の手順を実行してこの問題を解決します。

1. LOCAL_LISTENER パラメータをデータベース初期化パラメータ・ファイルに追加します。この LOCAL_LISTENER パラメータは Oracle Net デフォルト・リスナーのアドレスになるネットワーク名を示します。

Oracle8i または Oracle9i データベースのデータベース初期化パラメータ・ファイルを検索します。LOCAL_LISTENER パラメータをそのファイルに追加します。

```
LOCAL_LISTENER = <network-name>
```

2. Oracle Net デフォルト・リスナーのアドレスを特定します。

Oracle8i または Oracle9i 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。この定義内で、TCP プロトコルが使用されている最初のアドレスを調べます。

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)
        )
      )
    )
  )
```

最初のアドレスは次のとおりです。

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

3. tnsnames.ora ファイルに <network-name> エントリを作成します。

手順 2 で調べたアドレスを使用して、tnsnames.ora ファイルに <network-name> のエントリを作成します。

この例では、次のようにエントリを作成します。

```
<network-name>= (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。

7.3 シングルインスタンス 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シングルインスタンスの Oracle データベースを構成するには、少なくとも 1 つの仮想アドレスが含まれているグループにデータベースを追加します。Oracle Fail Safe により、シングルインスタンスの 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・サーバーのインスタンス
- 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 Oracle 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)

7.3.1 構成前に

シングルインスタンス・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、シングルインスタンス・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、7.3.3.3 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 つのシングルインスタンス・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、『Oracle9i データベース概要』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。

7.3.2 構成手順

表 7-1 に、可用性が高まるようにシングルインスタンス 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	Oracle データベース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 データベース・サーバーの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。
2	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」メニューで「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ため、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。)
3	サンプル・データベースの作成 (必要な場合)	「リソース」メニューで「サンプル・データベースの作成」を選択してサンプルのスタンドアロン・シングルインスタンス・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。
4	スタンドアロン・データベースの検証	「トラブルシューティング」メニューで「スタンドアロン・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 Oracle Net の構成を検証します。このコマンドにより、Oracle Fail Safe がデータベースに接続でき、スタンドアロン・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。
5	Oracle データベースのグループへの追加	「リソース」メニューで「グループに追加」を選択し、「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、可用性の高いシングルインスタンス Oracle データベース・サーバーを構成する際に役立ちます。
6	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 (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。

7.3.3 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シングルインスタンス 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。

- サービス名、インスタンス名、データベース名およびデータベース初期化パラメータ・ファイルのファイル指定などのシングルインスタンス Oracle データベースの識別情報
- データベースの SYSDBA（通常 SYS または 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を指定するように要求されます。

次の項では、シングルインスタンス・データベースの構成要件について詳細に説明します。

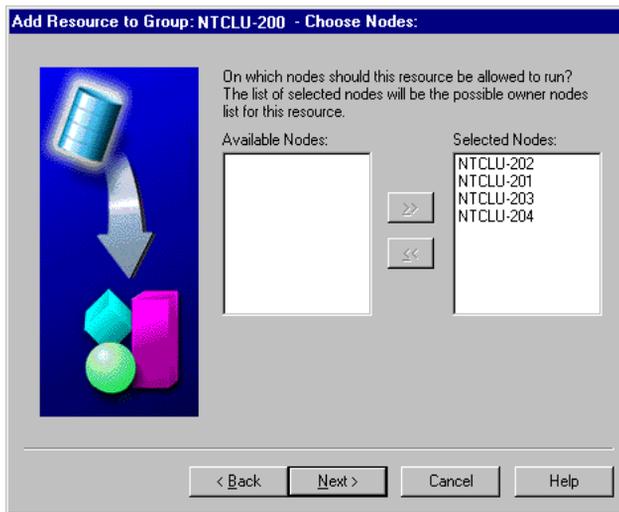
注意： INTERNAL アカウントは、Oracle9i より前の Oracle データベースでサポートされています。

7.3.3.1 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。

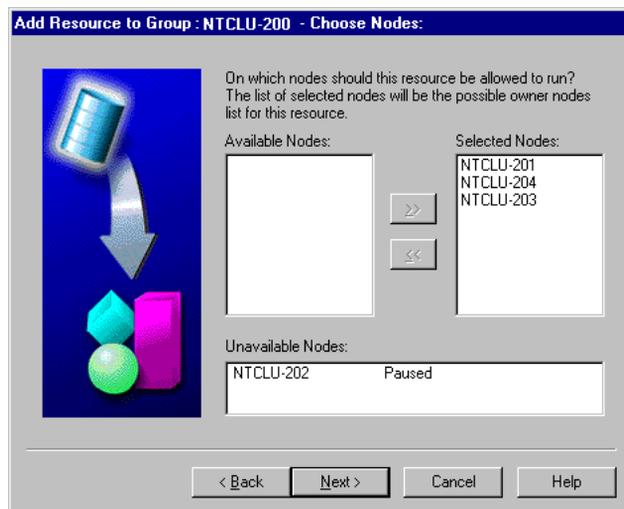
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

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



シングルインスタンス・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

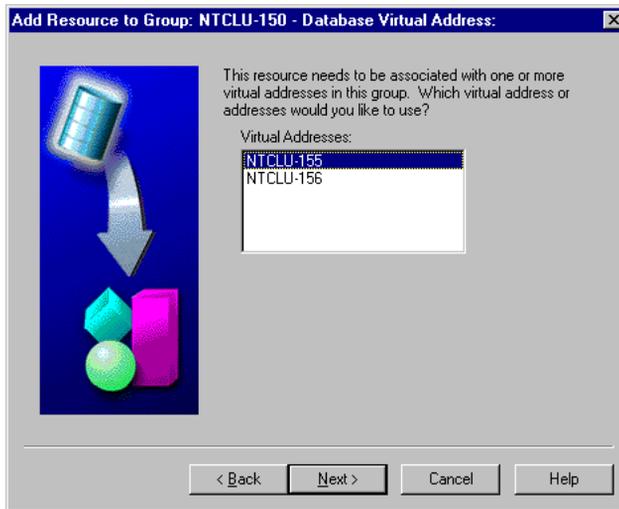
図 7-2 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



7.3.3.2 仮想アドレス

図 7-3 に示すように、シングルインスタンス・データベースを追加する対象の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 つ以上のシングルインスタンス・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シングルインスタンス・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。

7.3.3.3 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シングルインスタンス・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。

図 7-4 「データベースの識別情報」ウィザード・ページ

Add Resource to Group: NTCLU-150 - Database Identity

Group Name: FsTutorial

What is the identity of the database?

Service Name: SAMPLEDB.us.oracle.com

Instance Name: OFS2

Database Name: OFS2

What is the name of the initialization parameter file?

Parameter File: Y:\ORACLE\Admin\OFS2\pfile\INI

< Back Finish Cancel Help

Oracle Fail Safe では、(たとえば、tnsnames.ora ファイルを更新する場合) 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 より前のリリースで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。

Oracle Net サービス名にドメイン名を指定しない場合、7.4.3.1 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 データベース・サーバーの起動時には、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。Database Configuration Assistant を使用して作成した Oracle9i データベースでこの方法を行う場合は、7.3.3.2 項に示す注意事項を参照してください。

7.3.3.1 パラメータ・ファイルと、SPFILE を使用する Oracle9i データベース Oracle Fail Safe では、「パラメータ・ファイル」フィールドでテキスト初期化パラメータ・ファイル (PFILE) を指定する必要があります。バイナリ・サーバー・パラメータ・ファイル (SPFILE) を可用性が高められた Oracle9i データベースとともに使用する場合は、`SPFILE=<SPFILE-location>` パラメータを使用して、PFILE 内で SPFILE の位置を指定します。たとえば、PFILE に次のようなパラメータを含めることが可能です。

```
SPFILE=I:¥Oracle_Home¥admin¥oradb9i¥pfile¥spfileoradb9i.ora
remote_login_password=none
```

(Oracle Fail Safe で使用する PFILE 内に SPFILE を指定する場合、その SPFILE をエクスポートするには次の事項に注意する必要があります。ファイル指定なしで `CREATE PFILE FROM SPFILE` コマンドを使用すると、Oracle Fail Safe で使用する PFILE が上書きされます。このため、SPFILE をエクスポートする PFILE には必ず一意のファイル名を指定してください。サーバー・パラメータ・ファイルの詳細は、『Oracle9i データベース管理者ガイド』を参照してください。)

7.3.3.2 パラメータ・ファイルと、DBCA により作成した Oracle9i データベース Database Configuration Assistant を使用して Oracle9i データベースを作成すると、テキスト初期化パラメータ・ファイル (`init.ora`) は `<Oracle_Home>¥database` ディレクトリに作成されます。このファイルには、IFILE パラメータが含まれます。IFILE パラメータ値は、データベースの初期化パラメータが含まれる別のテキスト初期化パラメータ・ファイルのファイル指定です。

初期化パラメータ・ファイルの IFILE パラメータのファイル指定は、Windows レジストリ内に格納されます。Oracle Fail Safe のデフォルトでは、IFILE パラメータを含む初期化パラメータ・ファイルのファイル指定が「パラメータ・ファイル」フィールドに表示されます（「サンプル・データベースの作成」により作成したデータベース以外）。「サンプル・データベースの作成」により作成したデータベースの場合、IFILE パラメータで指定する初期化パラメータ・ファイルが「パラメータ・ファイル」フィールドに表示されます。

クラスタ環境では、IFILE パラメータで指定するファイルを「パラメータ・ファイル」フィールドに入力することをお勧めします。これにより、データベース初期化パラメータへの最短アクセスが可能になります。また、この方法を選択せずにパラメータ・ファイルをプライベート・ディスクに置く場合は、必ず IFILE パラメータを含む初期化パラメータ・ファイルと IFILE パラメータで指定するファイルを両方とも各クラスタ・ノードのプライベート・ディスクにコピーしてください。

7.3.3.4 データベース認証

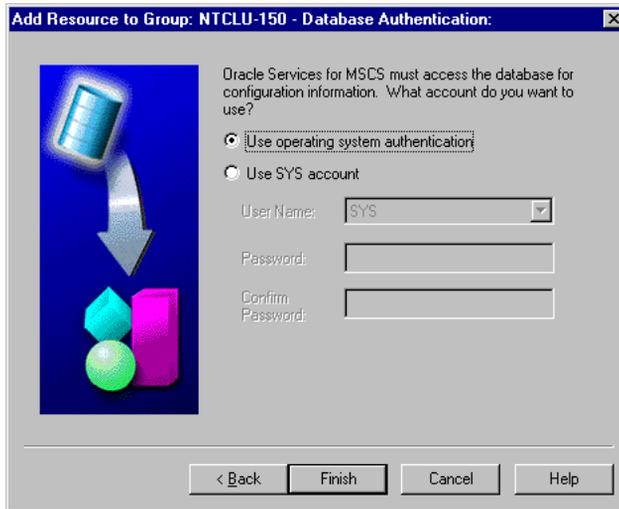
「データベースの認証」ページは、Oracle Services for MSCS のインストールに使用されたアカウントが、データベースに関連付けられた ORA_DBA グループまたは ORA_<SID>_DBA グループのいずれかの Windows オペレーティング・システム・グループにない場合に表示されます。Oracle Services for MSCS のインストールに使用されたアカウントが、ORA_DBA グループまたは ORA_<SID>_DBA グループにある場合は、オペレーティング・システム認証を使用してデータベースにアクセスできます。アカウントが ORA_DBA グループまたは ORA_<SID>_DBA グループにない場合は、SYS アカウントを使用してデータベースにアクセスする必要があります。

図 7-5 に示すように、Oracle Services for MSCS でデータベースおよびデータベース・インスタンスにアクセスする際に、オペレーティング・システム認証または SYS（あるいは INTERNAL）アカウントのどちらを使用するかを、このページから指定できます。

注意：「データベースの認証」ページは、Oracle Services for MSCS のインストールに使用されたアカウントが、すでにオペレーティング・システム認証によりデータベースにアクセス可能なグループのメンバーになっている場合は表示されません。

オペレーティング・システム認証を指定するには、「オペレーティング・システム認証を使用」を選択します。データベースの SYS アカウントを指定するには、「SYS アカウントを使用」を選択して SYS を選択し、アカウントのパスワードを入力および確認します。Oracle9i より前のリリースの Oracle サーバーで作成したデータベースの場合、「ユーザー名」ボックスで SYS アカウントのかわりに INTERNAL アカウントを選択できます。

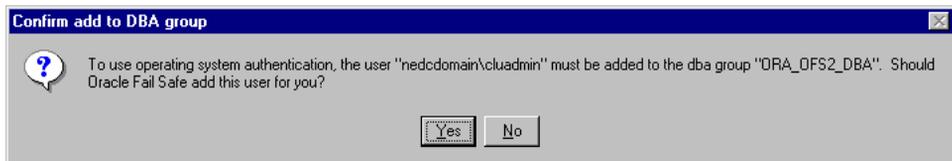
図 7-5 「データベースの認証」ウィザード・ページ



「オペレーティング・システム認証を使用」オプションを選択すると、Oracle Services for MSCS により、図 7-6 に示すような「DBA グループへ追加の確認」ウィンドウが開かれます。

図 7-6 のテキストは、Oracle Services for MSCS が nedcdomain\cluadmin アカウントにインストールされていることを示しています。「はい」を選択すると、Windows オペレーティング・システム・グループの ORA_OFS2_DBA に nedcdomain\cluadmin アカウントが追加されます。

図 7-6 「DBA グループへ追加の確認」ウィンドウ



7.3.3.5 データベース・パスワード

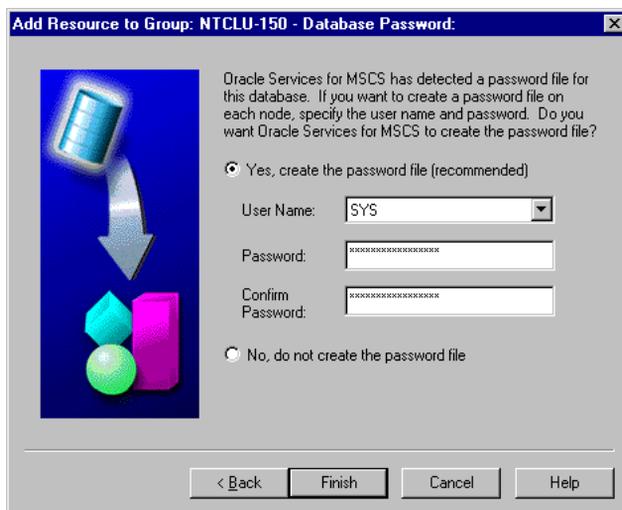
Oracle Services for MSCS によりスタンドアロン・データベースに関連付けられたパスワード・ファイルがあることが検出されると、フェイルセーフ・データベースのすべての可能所有者ノードでそのパスワード・ファイルを作成するかどうかを図 7-7 のように「DBA グループへ追加の確認」ウィザードで尋ねられます。

「はい、パスワード・ファイルを作成します」オプションを選択することをお勧めします。パスワード・ファイルはリモート操作を行う場合に必要となります。たとえば、**Recovery Manager** では、セキュリティ保護されていない Oracle Net 接続でターゲット・データベースにアクセスする場合にパスワードが必要です。

「いいえ、パスワード・ファイルを作成しません」オプションを選択すると、すべてのユーザーはオペレーティング・システム認証を使用するデータベースにアクセスする必要があります。さらに、ユーザーはデータベース管理のリモート操作を実行できません。

注意： Oracle Services for MSCS によりグループに追加するデータベースのパスワード・ファイルが検出されない場合、「データベース・パスワード」のページは表示されません。

図 7-7 「データベース・パスワード」ウィザード・ページ



Oracle Services for MSCS では、データベースのすべての可能所有者ノードでパスワード・ファイルを作成するかどうかによってデータベース初期化パラメータ・ファイルを次のように（必要に応じて）調整します。

- 「はい、パスワード・ファイルを作成します」を選択する場合
REMOTE_LOGIN_PASSWORDFILE パラメータを SHARED または EXCLUSIVE に設定します。
- 「いいえ、パスワード・ファイルを作成しません」を選択する場合
REMOTE_LOGIN_PASSWORDFILE パラメータを NONE に設定します。

注意： Oracle Services for MSCS 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。

データベースをグループに追加した後で SYS（または INTERNAL）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ります。データベースをグループに追加した後で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。

7.4 Oracle Net リスナー・リソースの作成および構成

シングルインスタンス・データベースをグループに追加すると、Oracle Fail Safe では、Oracle Net 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 以上の 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化すると、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イベントもログに取られます。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。

リリース 3.2.1 以上の Oracle Fail Safe では、データベースと、リスナーに関連付けられた IP アドレス（リスナーそのものではない）の間に依存性を作成します。この依存性は、クライアントがデータベースへ接続される前に IP アドレスがオフライン化された場合に、接続が中断されないように作成されたものです。

7.4.1 専用サーバー・モードでの共有ソケットの使用法

USE_SHARED_SOCKET パラメータを TRUE に設定すると、共有ソケットが使用できるようになります。このパラメータを TRUE に設定すると、ネットワーク・リスナーからデータベース・スレッドにクライアント接続用のソケット記述子が渡されます。その結果、クライアントはデータベース・スレッドへの新しい接続を確立する必要がなくなるためデータベース接続時間が改善されます。また、データベースの接続はすべてネットワーク・リスナーで使用するポート番号を共有するため、サード・パーティ製のプロキシ・サーバーの設定が簡単に行えます。

このパラメータは、TCP/IP 環境で専用サーバー・モードを使用する場合にかぎり有効です。

注意： この機能は、Oracle9i のリスナーおよびデータベースにのみ使用できません。Oracle9i より前の Oracle ホームで USE_SHARED_SOCKETS パラメータを TRUE に設定すると、Oracle Fail Safe グループのデータベースを追加または削除する際に次のエラーが返されます。

「Oracle 共有ソケットが <home-name> ホーム（ノード <node-name>）のリスナーに対して使用可能にされています。共有ソケットを使用不可にしてください。」

7.4.2 可用性の高いシングルインスタンス・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許可します。

シングルインスタンス・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・ノード上、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、Oracle Fail Safe で常に最新の構成でデータベース・インスタンスにアクセスできるようになります。

7.4.3 項では、データベースをグループに追加した後で、Oracle Fail Safe により listener.ora ファイルにエントリーを作成し、tnsnames.ora ファイルを更新して、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、クライアントがデータベースに接続できるようにする方法について説明します。

注意： Oracle Fail Safe では、TNS_ADMIN Windows 環境変数またはレジストリ・パラメータの使用をサポートしていません。Oracle Fail Safe では、<Oracle_Home>%network%admin ディレクトリにある Oracle Net ファイルの検索および更新は可能ですが、TNS_ADMIN Windows 環境変数やレジストリ・パラメータを指定してもこれらは無視されます。

7.4.3 データベースをグループに追加すると更新される Oracle Net 構成

次の項で説明するように、シングルインスタンス・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、listener.ora ファイルおよび sqlnet.ora ファイル内のデータベースの Oracle Net 構成が変更されます。

7.4.3.1 Oracle Fail Safe により行われる tnsnames.ora ファイルの更新内容

シングルインスタンス・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・サービス名エントリーは、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。

- Oracle Fail Safe によりネット・サービス名と、SERVICE_NAME パラメータまたは SID パラメータが調整されます。

サンプルのシングルインスタンス・データベースを作成、またはグループにシングルインスタンス・データベースを追加する際に、Oracle Net 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 により、ノード上でバージョンが最も新しいデータベースの Oracle ホーム内で、デフォルトのドメイン名が検索されます。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、ノード上で最もバージョンが新しいデータベースが、Oracle9i であると仮定します。Oracle Net ネット・サービス名に「MyDB」と指定し、Oracle9i 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。
- ノード上で最もバージョンが新しいデータベースの Oracle ホームに、デフォルトのドメイン名がない場合、ネット・サービス名には何も追加されません。たとえば、「MyDB」と指定した場合、ネット・サービス名も「MyDB」となります。

どちらの場合もネット・サービス名の決定後、Oracle Fail Safe により `tnsnames.ora` ファイルに同じ名前前のエントリがすでに存在するかどうかチェックされ、次のように `tnsnames.ora` ファイルが更新されます。

- ネット・サービス名のエントリがすでに存在する場合、Oracle Fail Safe により、仮想ホストに接続できるようにそのエントリが更新されます。Oracle Fail Safe では、(Oracle8i に追加された) `SERVICE_NAME` などの `CONNECT_DATA` パラメータは変更されません。
- エントリが存在しない場合、`tnsnames.ora` ファイルに新しいエントリが書き込まれます。この場合、ターゲット・データベースが Oracle8i 以上の場合でも `SID` は `CONNECT_DATA` パラメータで使用されます。(ユーザーは、サービス名ではなく `SID` を参照するネット・サービス名を使用して、Oracle8i 以上のデータベースに接続できます。)
- 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、(そのシングルインスタンス・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・ノード経由でシングルインスタンス Oracle データベースに対して作業を処理できるようにすることが必要な場合は、`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。

7.4.3.2 Oracle Fail Safe により行われる listener.ora ファイルの更新内容

シングルインスタンス・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. シングルインスタンス・データベースに関連付けられた仮想アドレスでリスニング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。

7.4.3.3 Oracle Fail Safe により行われる sqlnet.ora ファイルの更新内容

シングルインスタンス・データベースに対するオペレーティング・システム認証を選択済の状態、そのデータベースをグループに追加すると、パラメータを設定済でない場合は、Oracle Fail Safe によって、sqlnet.ora ファイルに SQLNET.AUTHENTICATION_SERVICES=(NTS) パラメータが追加されます。

7.4.4 可用性が高ま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法

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

```
(SID_DESC=
(SID_NAME=PLSExtProc)
(ORACLE_HOME = C:\¥Oracle¥Ora)
(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 Oracle Net リスナーの IPC アドレスが追加されます。

7.4.5 共有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、共有サーバー構成を使用する Oracle8、Oracle8i および Oracle9i シングルインスタンス・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。

注意： Oracle Fail Safe は、内部操作に専用サーバー接続を使用します。したがって、データベースを共有サーバー構成用に設定する際には、この専用サーバー接続が継続して機能するように、注意する必要があります。そのためには、各クラスタ・サーバー・ノード上の `tnsnames.ora` ファイルのデータベースに対する、ネット・サービス名エントリの接続データ部分に (`SERVER=DEDICATED`) パラメータを指定します。(共有サーバーが使用されている状態で `SERVER` パラメータが指定されていない場合、デフォルトではリスナーが共有サーバーを使用して接続を確立します。)

7.4.5.1 Oracle8 データベース用の共有サーバー

Oracle Fail Safe では、共有サーバー構成を使用するシングルインスタンス Oracle8 データベースをサポートします。ただし、共有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・シングルインスタンス・データベース、または現在グループ内のリソースであるシングルインスタンス・データベースを構成して、共有サーバーを使用できます。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. 共有サーバー構成を使用するシングルインスタンス・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 Oracle Net 構成ディレクトリで、`listener.ora` ファイルを検索します。(各リリースの `listener.ora` ファイルの場所は、[7.2.3 項](#)を参照してください。)

- 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。次に例を示します。

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

2. データベース初期化ファイル (`initofsf1.ora` など) 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・ファイルは、次のように共有インターコネクト上のディスクに存在する必要があるため注意してください。

```
H:¥OFSDDB¥OFS1¥PARAM¥initofsf1.ora
```

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してから再起動されます。

7.4.5.2 Oracle8i または Oracle9i データベースのための共有サーバー

Oracle8i または Oracle9i と Oracle Fail Safe を使用する環境で共有サーバー構成を使用するには、Oracle8i または Oracle9i データベースのデータベース・パラメータ・ファイルを変更する必要があります。

リスナー情報は、共有サーバー構成の LOCAL_LISTENER パラメータまたは DISPATCHERS パラメータのいずれかに指定できます。

注意： Oracle9i リリース 1 (9.0.1) から、MTS_DISPATCHERS パラメータは、DISPATCHERS パラメータという名前になりました。下位互換性により MTS_DISPATCHERS パラメータも使用できますが、Oracle9i の場合、DISPATCHERS パラメータを使用することをお勧めします。

共有サーバー構成で、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」操作中に共有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シングルインスタンス Oracle8i データベースまたは Oracle9i データベースは、共有サーバー・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共有サーバー構成を示します。

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

Oracle8i または Oracle9i 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、共有サーバー構成で DISPATCHERS 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、DISPATCHERS 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。7.2.5 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。

7.5 シングルインスタンス・データベースのセキュリティ要件

シングルインスタンス 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。

シングルインスタンスの**サンプル・データベース**を作成する場合や、グループにシングルインスタンス・データベースを追加する場合、Oracle Fail Safe ではオペレーティング・システム認証または SYS ユーザー・アカウントを使用してデータベースにアクセスする必要があります。ユーザーが SYS アカウントを使用してデータベースにアクセスする場合は、認証パスワード・ファイルを使用してデータベース初期化パラメータ・ファイル (init<database-name>.ora) 内の初期化パラメータ REMOTE_LOGIN_PASSWORDFILE を SHARED または EXCLUSIVE に設定します。ユーザーがオペレーティング・システム認証のみを使用してデータベースにアクセスする場合は、REMOTE_LOGIN_PASSWORDFILE パラメータを NONE 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、『Oracle9i データベース管理者ガイド』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

データベースのパスワード・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・インスタン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 アカウント以外アカウントが格納されている場合は、シングルインスタンス・フェイルセーフ・データベース用に、他のクラスタ・ノード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。

注意： INTERNAL アカウントは、Oracle9i より前の Oracle データベースでサポートされています。この項の SYS アカウントに関する説明は INTERNAL アカウントにも適用されます。

Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用してシングルインスタンス・データベースをグループに追加すると、Oracle Services for MSCS では、データベースが稼働するように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 SYS 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・インスタンスが稼働するように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、Oracle データベースの該当するリリースの管理者ガイドで、パスワード・ファイルの作成方法の手順を参照してください。

2. シングルインスタンス・データベースを含むグループを、そのデータベース・インスタンスが稼働するように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 以外のアカウントに権限を付与します。
4. データベースが稼働するように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・インスタンスが稼働するように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 SYSDBA アカウントのパスワードの変更

1 つ以上のデータベースをグループに追加した後で、それらデータベースの SYSDBA (SYS または INTERNAL) アカウントのパスワードを変更する場合は、Oracle Fail Safe Manager を使用してこの変更を行い、Is Alive ポーリングなどの操作を実行するときに Oracle Services for MSCS で正しいパスワードを使用してデータベースにアクセスできるようにします。

注意： INTERNAL アカウントは、Oracle9i より前の Oracle データベースでサポートされています。この項の SYS アカウントに関する説明は INTERNAL アカウントにも適用されます。

「データベース・パスワードの更新」ウィザード、またはフェイルセーフ・データベースのプロパティ・シートの「認証」タブを使用して、次のように SYSDBA パスワードを更新できます。

- 複数のデータベースに対する SYSDBA パスワードを変更するには、「データベース・パスワードの更新」ウィザードを使用することをお勧めします。Oracle Fail Safe Manager で「データベース・パスワードの更新」ウィザードにアクセスするには、「リソース」→「データベース・パスワードの更新」を選択します。

- 単一のデータベースに対する SYSDBA パスワードを変更するには、データベースの「認証」タブを使用することをお勧めします。ツリー・ビューから、パスワードを変更するフェイルセーフ・データベースを選択して、「認証」タブを選択します。

オペレーティング・システム認証が、パスワード・ファイルを使用せずに有効な場合、まずグループからデータベースを削除して、「リソースをグループに追加」ウィザードでデータベースをもう一度グループに追加しないと、Oracle Fail Safe Manager を使用した SYSDBA アカウント情報の変更はできません。データベースをもう一度グループに追加する場合、ウィザードの「データベースの認証」ページの「SYS アカウントを使用」を選択します。

次の情報については、オンライン・ヘルプを参照してください。

- 「データベース・パスワードの更新」ウィザードおよび「認証」プロパティ・ページの使用法に関する詳細
- リリース 3.3.1 より前の Oracle Services for MSCS でのこれらのツールの使用方法に関する情報
- Oracle Fail Safe Manager 以外のツールで SYSDBA パスワードがすでに変更されている場合の、これらのツールの使用方法に関する情報

7.5.3 Oracle Database Upgrade Assistant を使用したフェイルセーフ・データベースのアップグレード

この項では、Oracle Database Upgrade Assistant を使用して異なるリリース間でシングルインスタンス・フェイルセーフ・データベースをアップグレードする方法、あるいは異なる Oracle ホーム間でシングルインスタンス Oracle データベースを移動する方法について説明します。

アップグレードまたは新規 Oracle ホームに移動するシングルインスタンス・データベースのそれぞれに対して、次の手順を実行します。

1. シングルインスタンス・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、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. シングルインスタンス・データベースの移動先またはアップグレード先となる Oracle ホームから、Oracle Database Upgrade Assistant を実行します。
3. アップグレードするシングルインスタンス・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。データベースをアップグレードする際に、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・ファイルは Oracle Fail Safe による変換に適した場所に置かれます。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 Database Upgrade 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 Database Upgrade Assistant により、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 データベースにアップグレードする場合、Oracle Database Upgrade 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・ファイルを作成します (SID はデータベースのインスタンス名)。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. SQL*Plus を使用してデータベースに接続し、データベースを停止します。

```
SQL> SHUTDOWN
```
 - 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. SQL*Plus を使用してデータベースに接続し、次のコマンドを実行します。

```
SQL> STARTUP PFILE=init<SID>.ora MOUNT
SQL> ALTER DATABASE RENAME FILE
'<Oracle_Home>%database%mig<SID>.ora' TO
'cluster_disk%mig<SID>.ora';
SQL> SHUTDOWN
SQL> EXIT
```
6. Oracle Database Upgrade Assistant によって、グループ内のすべてのデータベースがアップグレードまたは新規 Oracle ホームに移動されてから、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」を選択します。
 - 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。

移動するグループのデータベースは、すべて同じ Oracle ホームに属している必要があります。Oracle Database Upgrade 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 ホームに移動する場合、そのグループ内のすべてのデータベースを新規 Oracle ホームに移動してください。

7.6 可用性を高めるための ctxsrv サーバーの構成

Oracle9i より前の Oracle データベース・サーバーの場合、ctxsrv サーバーはデータ操作言語 (DML) をバックグラウンドで処理してドキュメントの索引付け、検索、取出しおよび表示を行います。(Oracle9i から、標準の SQL または PL/SQL プロシージャによるドキュメントの索引付け、検索、取出しおよび表示が可能になりました。)

シングルインスタンス Oracle データベース・サーバーとともに 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。

1. 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
```

2. 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。
 - a. 「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。
 - b. 「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。
 - 必要に応じて、「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シングルインスタンス・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シングルインスタンス・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 (context.bat など) を入力します。
 - 「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 (D:\Orant\bin など) を入力します。
 - 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 環境でシングルインスタンス・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。

- クラスタを検出します。

Oracle Enterprise Manager で Oracle Fail Safe クラスタを検出するには、nmiconf.lst ファイルを編集して fs_discover.tcl をリストの最初のエン트리として追加する必要があります。fs_discover.tcl をファイルの最初にエントリしないと、Oracle Enterprise Manager で MSCS クラスタを検出できません。この操作は、すべてのクラスタ・ノードで実行する必要があります。複数の Oracle ホームから Oracle Intelligent Agent を使用している場合は、fs_discover.tcl エントリを各 Oracle ホームに追加します。nmiconf.lst ファイルは次のいずれかのディレクトリにあります。

- Oracle8 ホームの場合：

```
<Oracle_Home>%NET80%AGENT%CONFIG
```

- Oracle8i または Oracle9i ホームの場合：

```
<Oracle_Home>%NETWORK%AGENT%CONFIG
```

- 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。

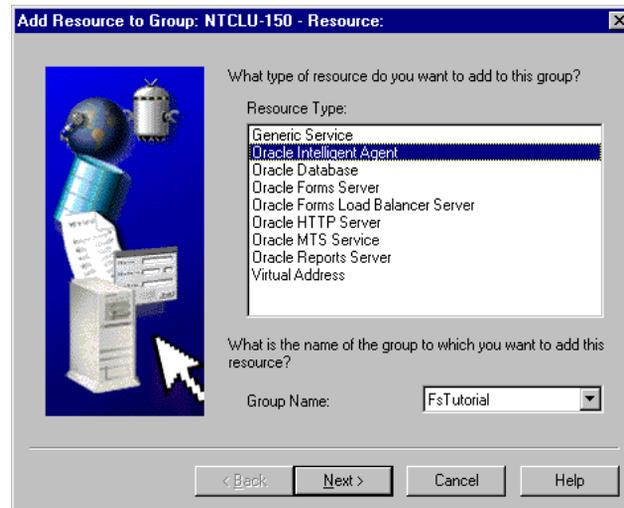
関連項目：

- 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 7.11.9 項を参照

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

グループにシングルインスタンス・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 7-8](#) のようなダイアログ・ボックスが表示されます。

図 7-8 「リソースをグループに追加」ウィザードの「リソース」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・リソ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・リソ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 Services for MSCS 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 シングルインスタンス・データベース・リカバリの最適化

Oracle Fail Safe によって可用性が高まるように構成された 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を実現でき
ます。Oracle のファスト・スタートおよび障害時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 Oracle データベース・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、ある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 Services for MSCS では、シングルインスタンス 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
シングルインスタンス・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： シングルインスタンス・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化およびオフライン化するときは、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リソースとしてみなし、再度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、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。

7.9 シングルインスタンス・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不可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. SQL*Plus 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。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。
3. 管理作業の終了後、SQL*Plus などのツールを使用してデータベースを停止します。
4. 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化します。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。新規データ・ファイルの追加によりグループが新しいディスクに依存する場合、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 透過的アプリケーション・フェイルオーバー（TAF）の構成

スタンドアロン・シングルインスタンス・データベースの場合、**データ・ファイル**（TAF）が発生すると、Oracle Net は別のリスナーに接続して障害の発生したデータベースへの再接続を確立します。このため、ユーザーは元の接続が維持されている場合と同じように、新しい接続を使用して作業を続行できます。スタンドアロン・シングルインスタンス・データベースの場合とシングルインスタンス・フェイルセーフ・データベースの場合では、TAF 機能による実際の処理は異なります。フェイルセーフ・データベースの場合、TAF が発生すると、Oracle Net はグループ・フェイルオーバーにより他のクラスタ・ノードに移された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、TAF のフェイルオーバーという語は、Oracle Net 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、TAF 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再接続が確立されます。

実装にはこのような違いがありますが、TAF を管理する方法は同じです。

Oracle Fail Safe で構成したデータベースに接続しているときに、TAF のメリットを活かすためには、**クライアント・アプリケーション**が次の条件を満たしている必要があります。

- Oracle OCI リリース 8.0.6 以上または Oracle ODBC リリース 8.0.6 以上を使用している。
- Oracle Net を介して Oracle データベースに接続している。

TAF では、グループ・フェイルオーバーの後、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。

注意： TAF 機能は、Oracle データベース・サーバー Enterprise Edition でのみサポートされています。

7.10.1 透過的アプリケーション・フェイルオーバーによりリストアされる機能

TAF では、アクティブなデータベース接続に関連付けられた、次のいずれかあるいはすべての要素が自動的にリストアされます。ただし、その他の要素を TAF により再接続するには、アプリケーション・コードにそれらが埋め込まれている必要があります。

- クライアント / サーバー・データベース接続

TAF では、同じ接続文字列または TAF の構成時に指定した代替接続文字列を使用して自動的に再接続を確立します。

- ユーザー・データベース・セッション

TAF では、障害発生前に使用されていたユーザー ID と同じ ID でユーザーが自動的にログインされます。複数のユーザーがデータベースに接続していた場合は、ユーザーがデータベース・コマンド処理を試行するときに TAF で自動的にそれらのユーザーがログインされます。TAF では、その他のセッション・プロパティは自動的にリストアされません。ただし、コールバック関数を起動するとこれらのプロパティをリストアできません。

- 実行済のコマンド

接続障害の発生時に、あるコマンドが完全に実行されてデータベースの状態が変更された場合、TAF はそのコマンドをリSENDしません。データベースを変更する可能性のあるコマンドに対して再接続が確立されると、アプリケーションに対してエラー・メッセージが発行されます。

- フェッチに使用されるオープン・カーソル

TAF では、アプリケーションで、フェイルオーバー後の行のフェッチをフェイルオーバー前のカーソル位置から開始できます。これを「選択フェイルオーバー」と呼びます。これは、同じスナップショットを使用して SELECT 文を再実行し、すでにフェッチされていた行を破棄し、最初にフェッチされていない行を取り出すことで実行されます。破棄された行が最初に返された行かどうかは TAF によって確認され、そうでない場合はエラー・メッセージが返されます。

- アクティブなトランザクション

アクティブなトランザクションはフェイルオーバー後に TAF で維持できないため、障害の発生時にロールバックされます。ROLLBACK コマンドが発行されるまでアプリケーションにエラー・メッセージが返されます。

- サーバー側プログラム変数

PL/SQL パッケージ状態のようなサーバー側プログラム変数は障害発生時に失われます。TAF でリカバリされません。これらはフェイルオーバー・コールバック関数からのコールにより初期化できます。

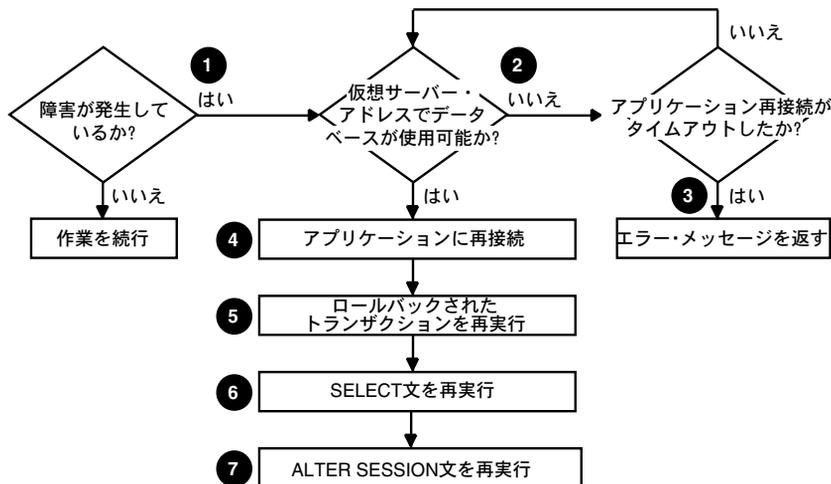
注意： アプリケーションでは、フェイルオーバー後のリカバリ機能の強化またはカスタマイズのため、Oracle Call Interface（OCI）のコールバック関数を使用できます。たとえば、コールバックを使用して中断された書き込み操作を処理できます。詳細は、『Oracle Call Interface プログラマーズ・ガイド』を参照してください。

7.10.2 透過的アプリケーション・フェイルオーバー実行時の処理

TAF 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 7-9 に、TAF に関連した処理を示します。

図 7-9 透過的アプリケーション・フェイルオーバーのフロー・チャート



次に、図 7-9 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. 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. 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時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。`tnsnames.ora` ファイルにパラメータを設定する方法の詳細は、[7.10.3 項](#)に記載されています。

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 文が再実行される時に行が取り出される順序は決まっていません。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 Call Interface 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。

7.10.3 透過的アプリケーション・フェイルオーバーのための tnsnames.ora ファイルの変更

クライアント・アプリケーションに対して TAF を使用可能にするには、Oracle Net の tnsnames.ora ファイルを更新する必要があります。Oracle8 以上の ODBC クライアントと OCI クライアントの両方に対して自動再接続を使用可能にする場合、7.10.3.1 項および 7.10.3.2 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かぎ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 HOST パラメータおよび CONNECT_DATA パラメータで行います。次の項を参照してください。

7.10.3.1 HOST パラメータの構成

グループ内のリソースであるシングルインスタンス・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、物理クラスター・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 パラメータに仮想サーバー・アドレス 138.2.26.155 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= 138.2.26.155) <-- 仮想サーバー・アドレス
              (PORT=1521))
    .
    .
    .
  )
)
```

アプリケーションで TAF 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・アドレスを指定する必要があります。

7.10.3.2 CONNECT_DATA パラメータの構成

TAF 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。次に例を示します。

```
(DESCRIPTION =
  (ADDRESS_LIST =
    .
    .
    .
  )
  (CONNECT_DATA=
    (SERVICE_NAME=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。次に、FAILOVER_MODE サブパラメータについて説明します。

- 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示します。
- METHOD サブパラメータは、フェイルオーバー・パフォーマンスを最適化します。この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 パラメータに 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。このサブパラメータは、Oracle Fail Safe 環境内では特に重要です。データベースがフェイルオーバー・ノードで再度オンライン化されるまでの時間を指定できます。

接続操作に関する詳細情報とその構文は、『Oracle9i Net Services 管理者ガイド』を参照してください。

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ません。	アプリケーション問合せを再実行しません。	TAF を試行せずにユーザーにエラー・メッセージを返します。これはデフォルトの処理です。
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します。	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ます。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ません。	障害が発生しない場合でも、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します。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 文の再実行に気が付きません。
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します。	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ません。	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します。

7.10.4 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 TAF を実演する、実行可能デモが提供されています。このプログラムおよび付属ドキュメント (readme.txt) は、<Oracle_Home>%fs%fsmgr%sample ディレクトリにあり、オンラインで使用できます。

7.11 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 が可用性の高いシングルインスタンス・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シングルインスタンス 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化する際に発生するエラーの処理

Oracle Fail Safe がシングルインスタンス・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。Oracle Fail Safe では、クラスタ上のすべてのシングルインスタンス・フェイルセーフ・データベースに同じスクリプトを使用しません。

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_Home>%fs%fsvr%scripts
```

シングルインスタンス・データベースをオンライン化できない場合、Oracle Fail Safe は次のように、スクリプトを実行するプロセスを生成し、エラー・コード、データベース名、データベース SID、TNS サービス名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てスクリプトを実行します。

```
FsDbError.bat <error-code> <database-name> <SID> <TNS service name>  
<parameter-file-spec>
```

次に例を示します。

```
FsDbError.bat ORA-01113 OracleDB OracleDB OracleDB.WORLD  
D:%Ora%admin%OracleDB%pfile%initOracleDB.ora
```

スクリプトが実行されているプロセスは、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、Windows イベント ログにイベントのログを取り、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。

スクリプトの成否にかかわらず、Oracle Fail Safe では、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シングルインスタンス・データベースのオンライン化の試行が継続されます。

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。

シングルインスタンス・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。
- Oracle 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シングルインスタンス・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、シングルインスタンス・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す。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、シングルインスタンス・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、ユーザーにかか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 データベースをグループに追加する際の問題

シングルインスタンス・データベースをグループへ追加する際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（6.13 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・データベースのグループへの追加を試行し Oracle Net 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（Oracle 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、7.11.8.4 項に記述されている方法に従って、Oracle Net 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。

- シングルインスタンス・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

シングルインスタンス・データベースが含まれるグループをオンライン化する際に問題が発生した場合は、次の操作を実行します。

- 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。

- Oracle Net のリスナー・ログを確認します。

Oracle Net 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。

- Oracle8i または Oracle9i 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 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

- シングルインスタンス・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・リソースが機能していないとみなします。

- Oracle Net 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る必要があります。

- 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シングルインスタンス・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。

- シングルインスタンス・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場合、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。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可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。

- ユーザーが SYS（または INTERNAL）アカウントを使用してデータベースにアクセスする場合は、データベース初期化パラメータ・ファイル（init<database-name>.ora）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。
- ユーザーがオペレーティング・システム認証のみを使用してデータベースにアクセスする場合は、データベース初期化パラメータ・ファイル内の初期化パラメータ REMOTE_LOGIN_PASSWORDFILE が NONE 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 プロセスが集中している操作の際のグループ・フェイルオーバー

プロセスが集中している操作（インポート操作など）では、Is Alive ポーリングが正常に実行されず、予期しないグループ・フェイルオーバーが発生する場合があります。このような場合には、FSCMD DISABLEISALIVE コマンドを発行して、データベースに対する Is Alive ポーリングを使用不可にできます。Is Alive ポーリングを使用不可にする場合は、再度使用可能にするまで、Oracle Fail Safe によりインスタンスの監視が中止されるので注意してください。Is Alive ポーリングを再度使用可能にするには、FSCMD ENABLEISALIVE コマンドを使用します。

これらの FSCMD コマンドは、プロセスが集中している操作の完了時に Is Alive ポーリングを再度使用可能にできるようにスクリプト内から発行することをお勧めします。

FSCMD コマンドの詳細は、[第 5 章](#)を参照してください。

7.11.6 データベース認証

Oracle Fail Safe でシングルインスタンス・データベースをオンライン化またはオフライン化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、次の操作を実行します。

- 「リソースをグループに追加」ウィザードの「認証」ページで「このアカウントを使用」を選択する場合は、データベース初期化パラメータ・ファイル (init<database-name>.ora) 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。

[7.5 項](#)に、データベース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 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できる必要があります。データベース・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。更新しないと、Oracle Fail Safe で Is Alive ポーリングを使用するデータベースの監視ができなくなります。この状態は、Windows イベント ビューアにより記録されます。

[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。

7.11.7 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 for Windows インストール・ガイド』を参照してください。
- サンプル・データベースの作成時に 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場合や、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した場合、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、削除操作を再度実行してください。

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。

7.11.8 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 Oracle Net 構成を調べる必要があります。

Oracle Fail Safe では、Oracle 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。

7.11.8.1 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、`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: Oracle Net: <name>

このメッセージ・コードは、Oracle Net 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 Net Assistant を使用してファイルを作成しなおします。
- 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、複数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: Oracle Net リスナーの Windows サービス <name> が開始できませんでした。

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せん。

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 つのリスナーにかぎり、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、または `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ある必要があります。

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、Oracle Net 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。

7.11.8.2 リスナー作成時の問題

Oracle Fail Safe Manager では、リスナー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\NETWORK\ADMIN で、リスナーが動作する仮想アドレスが ntclu-155 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\NETWORK\LOG\fs\ntclu-155.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 fs\ntclu-155 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。

7.11.8.3 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 ファイルを変更すると、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 Oracle Net 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。

7.11.8.4 ロールバック・ファイル

Oracle Net 構成ファイルの変更後、Oracle Fail Safe で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。

7.11.9 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 <username> 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、ユーザー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 SYS 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 SYS アカウントと一致していることを確認します。

7.11.10 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 (Oracle Net) 構成の有効性を検査します。

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

Oracle Fail Safe では、Oracle Forms Load Balancer Server をグループに追加するウィザードが提供され、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 では、可用性が高まるように Oracle Forms Server を構成するウィザードも提供されます。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 9 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	8.1 項
スタンドアロン Oracle Forms Load Balancer Server の検出	8.2 項
Oracle Forms Load Balancer Server のグループへの追加	8.3 項
Oracle Forms Load Balancer Server のセキュリティ要件	8.4 項
起動パラメータの変更	8.5 項
Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング	8.6 項

8.1 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、Common Gateway Interface (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。Forms の各実行要求を、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、Forms Server を実行するシステムとして負荷の最も小さいシステム名を指定して、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、Oracle Net 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 実行可能ファイルでは、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ことができます。

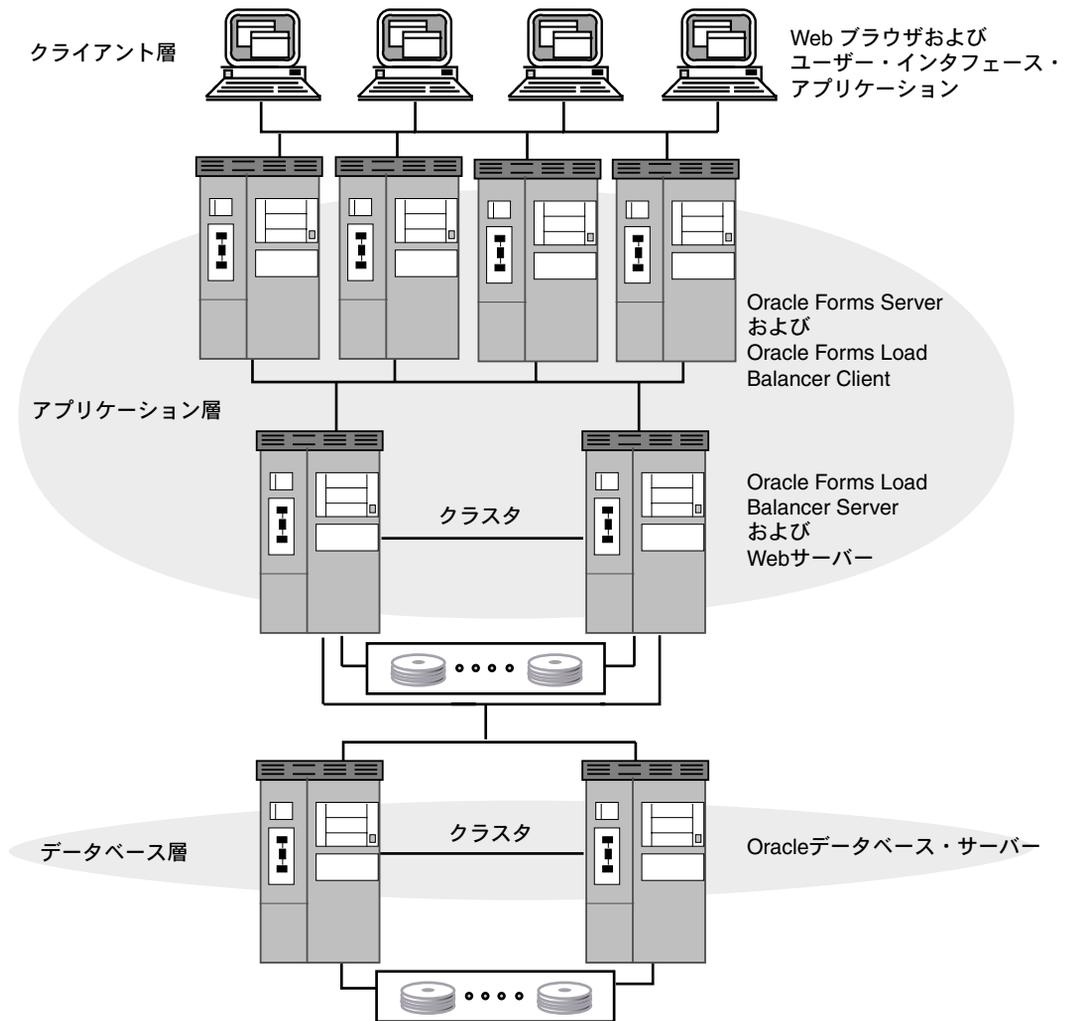
図 8-1 に、Oracle Forms Load Balancer Server と Oracle Web サーバーがデータベースとは別のクラスタに常駐する階層化実装を示します。Oracle Forms Server および Oracle Forms Load Balancer Client は、いくつかのクラスタ化されていないシステム上に常駐しています。

図 8-1 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 8-1 Forms Load Balancer Server を使用した Forms の 3 層アーキテクチャ



8.2 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 Services for MSCS 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、Oracle Forms Load Balancer Server エントリを探し、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。

8.3 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

Oracle HTTP Server と、Forms Server でアクセスするデータベースなども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、Oracle HTTP Server 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。

8.3.1 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。

8.3.2 構成手順

Oracle Fail Safe により、Oracle Forms Load Balancer Server の可用性を高めるため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 8-1 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 8-1 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	インストールの情報は、Oracle Forms のマニュアルを参照してください。
2	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
3	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 8-1 Oracle Forms Load Balancer Server の構成手順（続き）

手順	処置	説明
4	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ため、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。）
5	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。
6	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。
7	Oracle HTTP Server をフェイルオーバー用に構成	Oracle HTTP Server を作成してグループに追加し、Oracle Forms アプリケーションを Web 上で使用可能にします。Oracle HTTP Server を作成してグループに追加する方法の詳細は、第 11 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。

8.3.3 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 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 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。

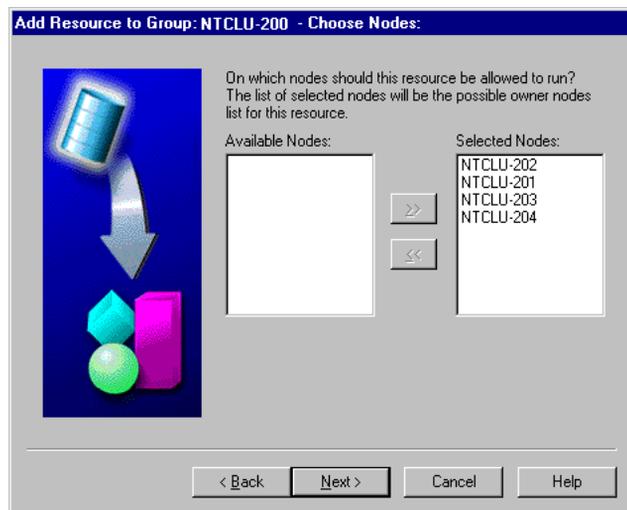
次の各項では、Oracle Forms Load Balancer Server をグループに追加するためのデータ要件を詳細に説明します。

8.3.3.1 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 8-2](#) 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。

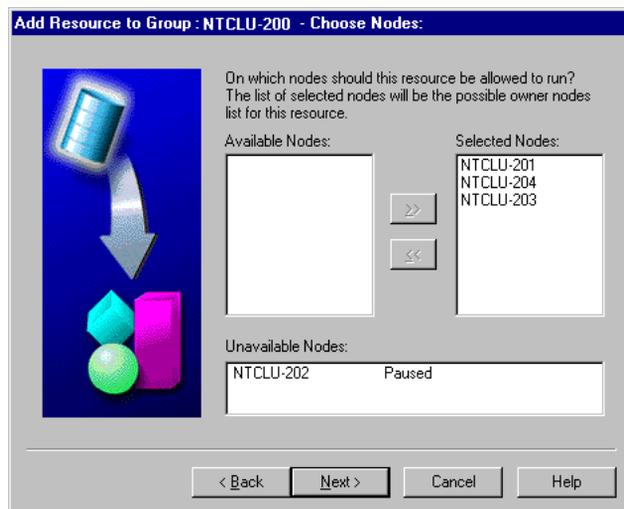
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

図 8-2 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。

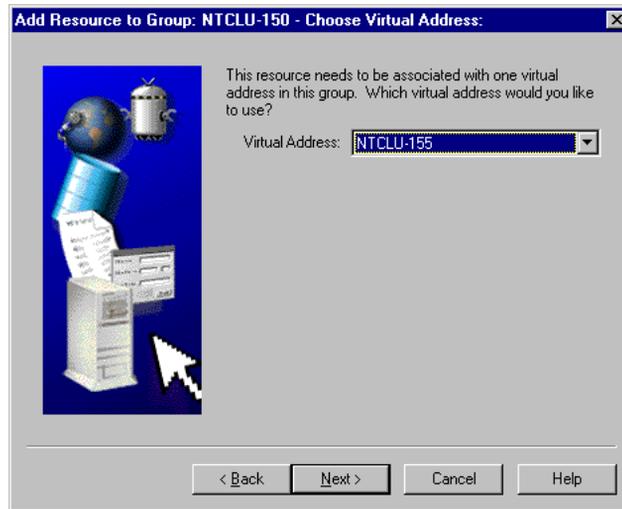
図 8-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



8.3.3.2 Oracle 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 8-4 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 8-4 「仮想アドレスの選択」ウィザード・ページ



8.3.3.3 Oracle Forms Load Balancer Server 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 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。

- 最大

これは、Oracle Forms Load Balancer Server にアクセスする Oracle Forms Load Balancer Client の最大数です。

必須の起動パラメータ変更の詳細は、[8.5 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 8-5 「Load Balancer Server の識別情報」ウィザード・ページ

Add Resource to Group: NTCLU-150 - Load Balancer Server Identity:

Group Name: FsTutorial

Which Load Balancer Server do you want to add to this group? You can select the name of an existing Load Balancer Server, or enter a unique name to have Oracle Fail Safe create a new Load Balancer Server.

Load Balancer Server Name: Test_LBS

What is the data port?
Data Port: 9010

What is the request port?
Request Port: 9020

What is the maximum number of clients?
Maximum: 1000

< Back Finish Cancel Help

8.4 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。

8.5 起動パラメータの変更

次の各項では、次の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、8.5.4 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。

8.5.1 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。

8.5.2 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。

8.5.3 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。

8.5.4 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、次のようにします。

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。

8.6 Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング

[フェイルセーフ・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、Oracle HTTP Server 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 では、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提供され、Oracle Forms アプリケーションの可用性を高めることもできます。複数の Oracle Forms Server を使用するソリューションでは、複数の Oracle Forms Server よりも 1 つの 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 8 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	9.1 項
スタンドアロン Oracle Forms Server の検出	9.2 項
Oracle Forms Server のグループへの追加	9.3 項
Oracle Forms Server のセキュリティ要件	9.4 項
Oracle Forms Server へのクライアント接続	9.5 項
Oracle Forms 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 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 Oracle HTTP Server、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 データベースなどを含めた、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。

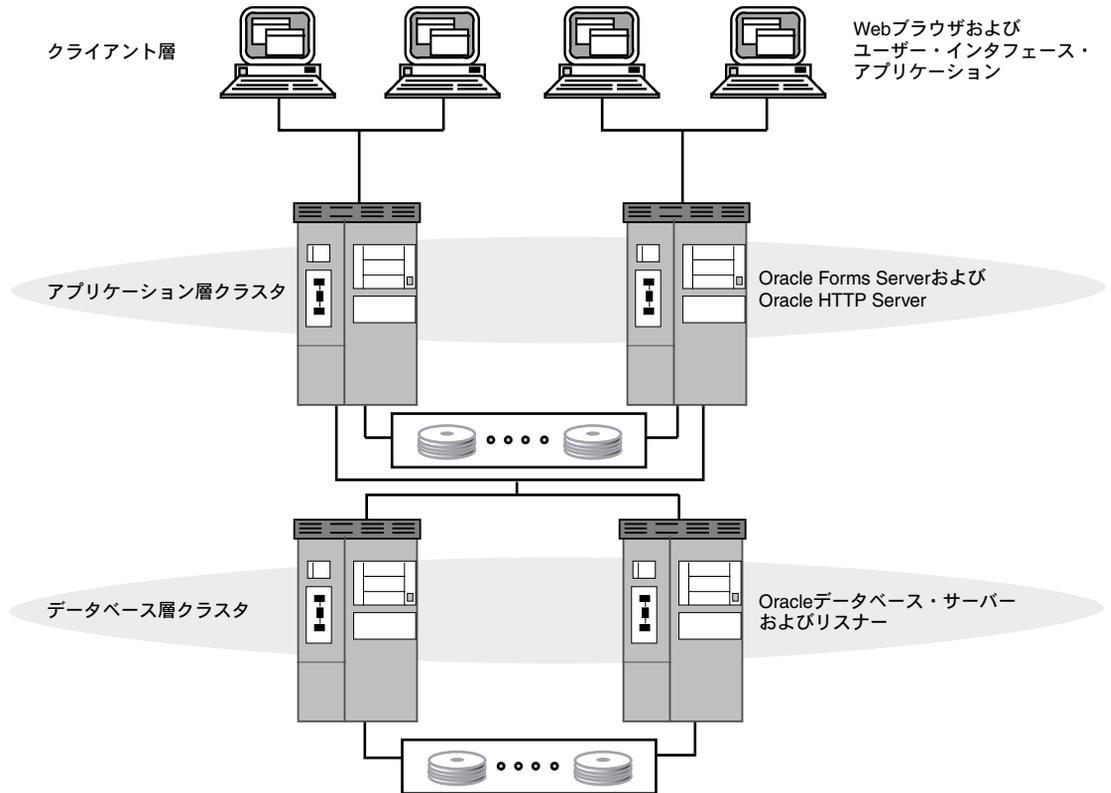
図 9-1 に、Oracle Forms Server と Oracle HTTP Server がデータベースとは別のクラスターに常駐する階層化実装を示します。

図 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： Oracle Forms Server、Oracle HTTP Server
- 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 9-1 Oracle Forms の 3 層アーキテクチャ



9.2 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 Services for MSCS 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、Oracle Forms Server エントリを探し、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 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 Oracle HTTP Server を追加する必要があります。Oracle HTTP Server は仮想アドレスに依存します。Oracle HTTP Server 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 Oracle HTTP Server 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。

9.3.1 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。Oracle Forms Server のホストとなるクラスタと同じクラスタまたは異なるクラスタのいずれかで、フォームのアクセス先となる Oracle データベース・サーバーの可用性が高まるように構成することで、可用性をさらに高めることができます。

9.3.2 構成手順

Oracle Fail Safe により、Oracle Forms Server の可用性を高めるため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 9-1 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」または「ヘルプ」→「チュートリアル」を選択します。

表 9-1 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	インストールの情報は、Oracle Forms のマニュアルを参照してください。
2	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タ上のクラスタ・ディスクに配置する必要があります。
3	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uggを使用して、このテストを実施します。
4	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
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セスを実行します。

表 9-1 Oracle Forms Server の構成手順 (続き)

手順	処置	説明
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ため、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。)
7	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、ポート番号の指定、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などが可能です。
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、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	Oracle HTTP Server をフェイルオーバー用に構成	Oracle HTTP Server を作成してグループに追加し、Oracle Forms アプリケーションを Web 上で使用可能にします。Oracle HTTP Server を作成してグループに追加する方法の詳細は、第 11 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全体で一意になるようにしてください。

9.3.3 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 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

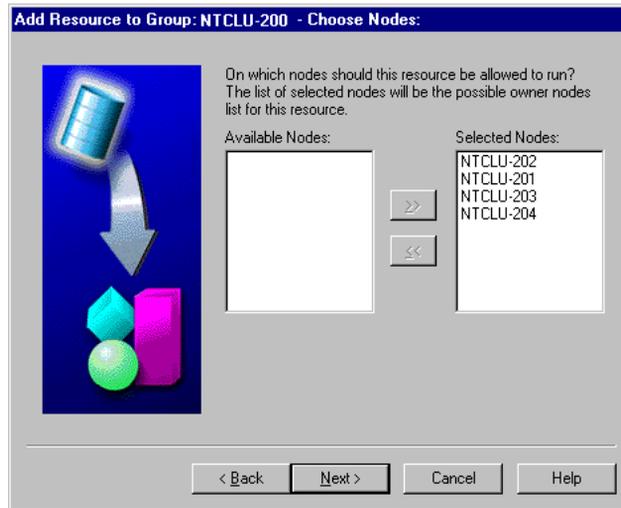
次の項では、構成情報の要件を詳細に説明します。

9.3.3.1 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 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 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。

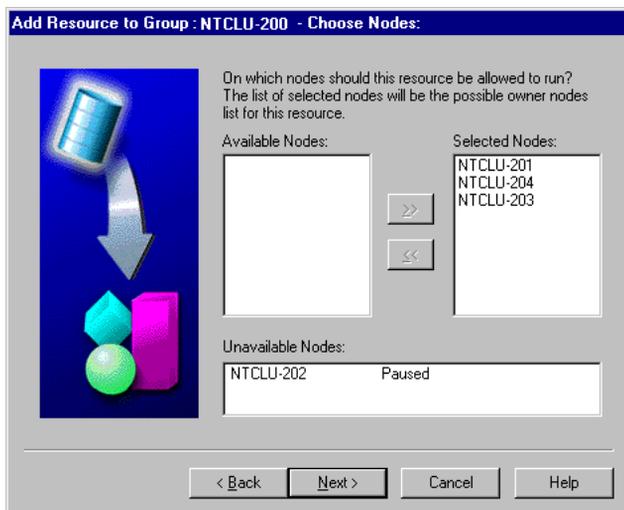
[2.6.6 項](#)では、「可能所有者ノード」リストの概念の詳細を説明します。

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 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



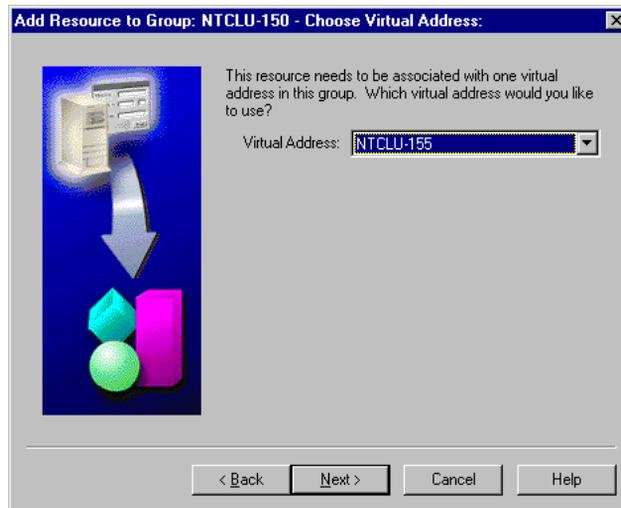
9.3.3.2 Forms Server の仮想アドレス

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 9-4 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 9-4 Oracle Forms の「仮想アドレスの選択」ウィザード・ページ



9.3.3.3 Forms Server の識別情報

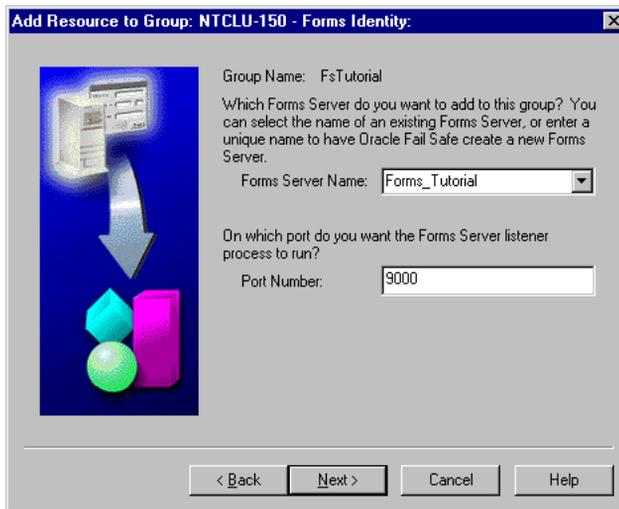
図 9-5 に示すように、Oracle Forms Server の識別情報は、Oracle Forms Server の名前と、Oracle Forms Server リスナー・プロセスが実行されるポートで構成されます。Forms Server 名とポート番号は、クラスタ内で一意になるようにします。ここに指定するポート番号は、アプリケーションの HTML ファイルまたは URL に指定したポート番号と同一にする必要があります。

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。

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 9-5 「Forms の識別情報」ウィザード・ページ



9.3.3.4 Forms のパラメータ

図 9-6 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、Secure Sockets Layer (SSL) 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。

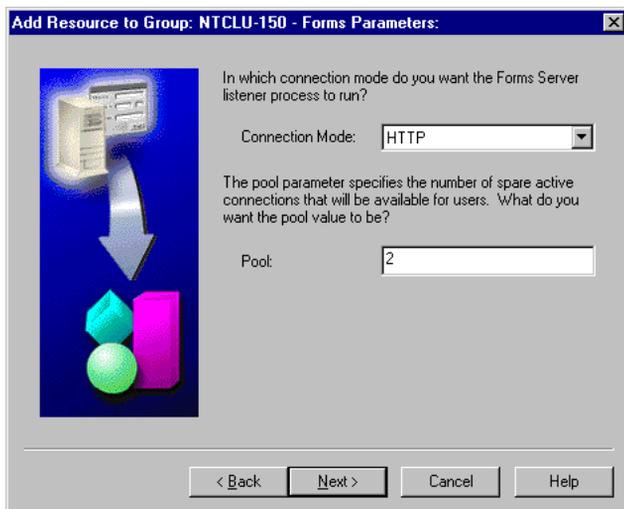
プール・パラメータでは、Oracle Forms Server で作成し、保持する必要のあるワーカー・プロセス (ifweb60.exe) の数を指定します。ワーカー・プロセスは、1 ユーザーに対して 1 つ割り当てられます。プール値を 4 に指定した場合、4 人のユーザーが遅延なく作業を開始できますが、5 人目のユーザーからは、ワーカー・プロセスを開始するたびに遅延が生じます。

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 9-6 「Forms のパラメータ」 ウィザード・ページ



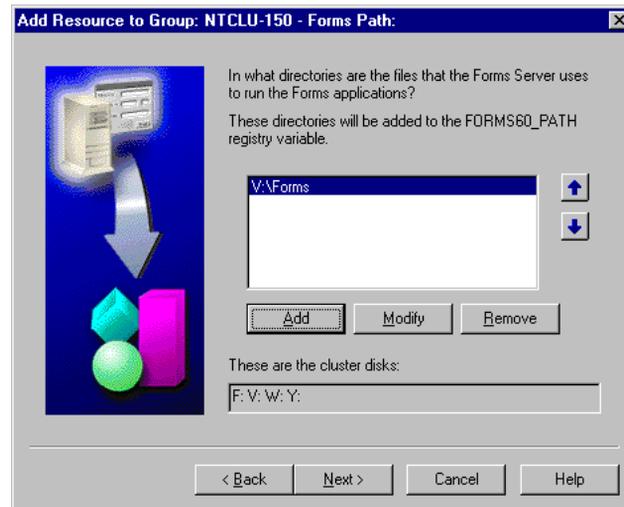
9.3.3.5 Oracle Forms アプリケーション・ファイルのパス

図 9-7 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・ページにおける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 9-7 「Forms パス」 ウィザード・ページ



9.3.3.6 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 9-8 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を使用する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 9-8 「Forms のデータベース」ウィザード・ページ

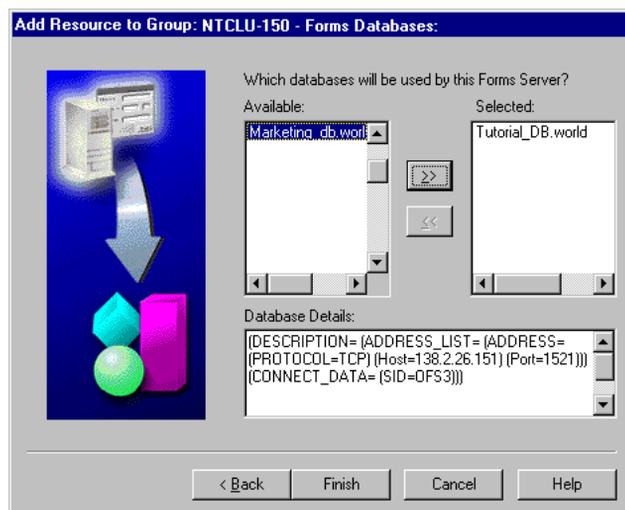
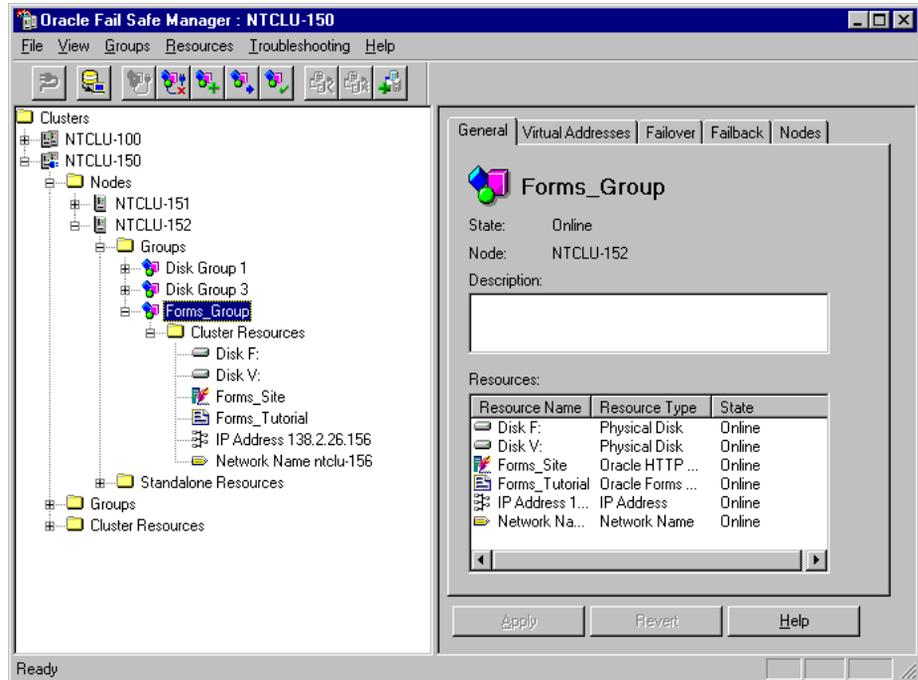


図 9-9 は Oracle Fail Safe Manager のツリー・ビューで、可用性が高まるように構成された Oracle Forms Server を示しています。このツリー・ビューには、Web でフォームを表示できるように Forms_Group に追加された Oracle HTTP Server (Forms_Site) も表示されています。

図 9-9 Web ベースの Forms を示すツリー・ビュー



9.4 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。

9.5 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、Oracle HTTP Server 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。

詳細の参照先は次のとおりです。

- 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 11 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 Oracle HTTP Server 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。

9.6 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。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。

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、これらのツールの説明を記載しています。

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

9.6.1 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、次の操作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・バーで「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。

9.6.2 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、Oracle HTTP Server の**仮想サーバー**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 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"
```

9.6.3 Java アプレットを実行できない

Java アプレットが実行されず、クラス例外エラーを受け取った場合は、次の操作を実行します。

- 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 Forms と JInitiator のソフトウェア互換性の詳細は、Oracle Forms のマニュアルを参照してください。)
- 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。次に例を示します。

```
<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">
```

9.6.4 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、次の操作を実行します。

- Web ブラウザで互換性のあるリリースの JInitiator ソフトウェアを使用していることを確認します。(Oracle Forms と JInitiator のソフトウェア互換性の詳細は、Oracle Forms のマニュアルを参照してください。)
- 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"
```

9.6.5 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、次の操作を実行します。

- 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。

9.6.6 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、次の操作を実行します。

- フォームを Forms ランタイムでローカルに実行します。
- 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。

フォームを Forms ランタイムでローカルに実行した後、再び Web ブラウザで実行しようとしたときに発生した問題をすべて修正します。

9.6.7 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
```


10

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、高い可用性が提供されます。(Oracle Reports を Web 上で使用できるようにするには、Oracle HTTP Server 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を得られます。

- Oracle Reports アプリケーションの停止時間が最短になります。
- Oracle Reports Server にスケジューリング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	10.1 項
スタンドアロン Oracle Reports Server の検出	10.2 項
Oracle Reports Server のグループへの追加	10.3 項
Oracle Reports Server へのクライアント接続	10.4 項
マスター / スレーブ実装のための構成後の手順	10.5 項
可用性の高い Oracle Reports Server へのユーザー・アクセスの制御	10.6 項
Oracle Reports のスケジューリング	10.7 項
Oracle Reports Server に関連する問題のトラブルシューティング	10.8 項

10.1 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、Oracle HTTP Server、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。

10.1.1 Web ベースの実装

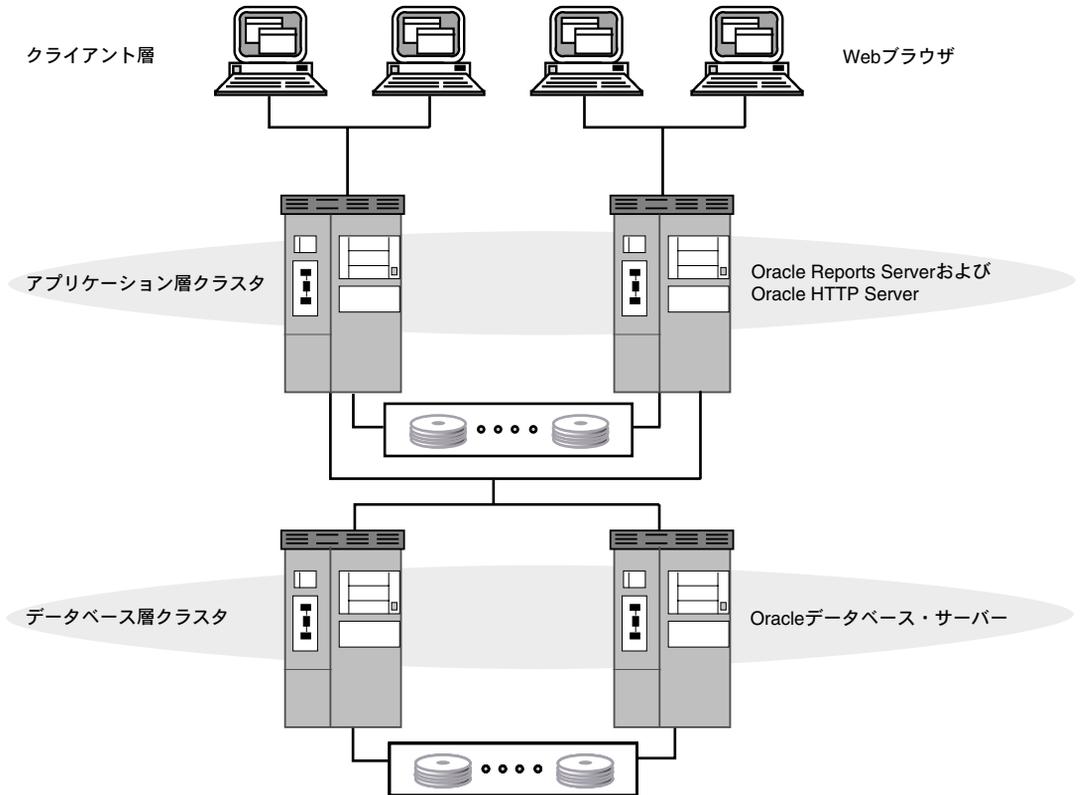
Oracle Reports Server を Reports Web カートリッジ、Reports Web CGI または Reports サブレットとともに使用して、Web ブラウザからレポートを実行できます。（Reports Web カートリッジ、Reports Web CGI および Reports サブレットの設定方法の詳細は、Oracle Reports のマニュアルを参照してください。）

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
Oracle HTTP Server と Oracle Reports Server の両方が単層として含まれる場合と、Oracle HTTP Server 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 10-1 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 Oracle HTTP Server 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 10-1 Oracle Reports の Web 実装のための 3 層アーキテクチャ



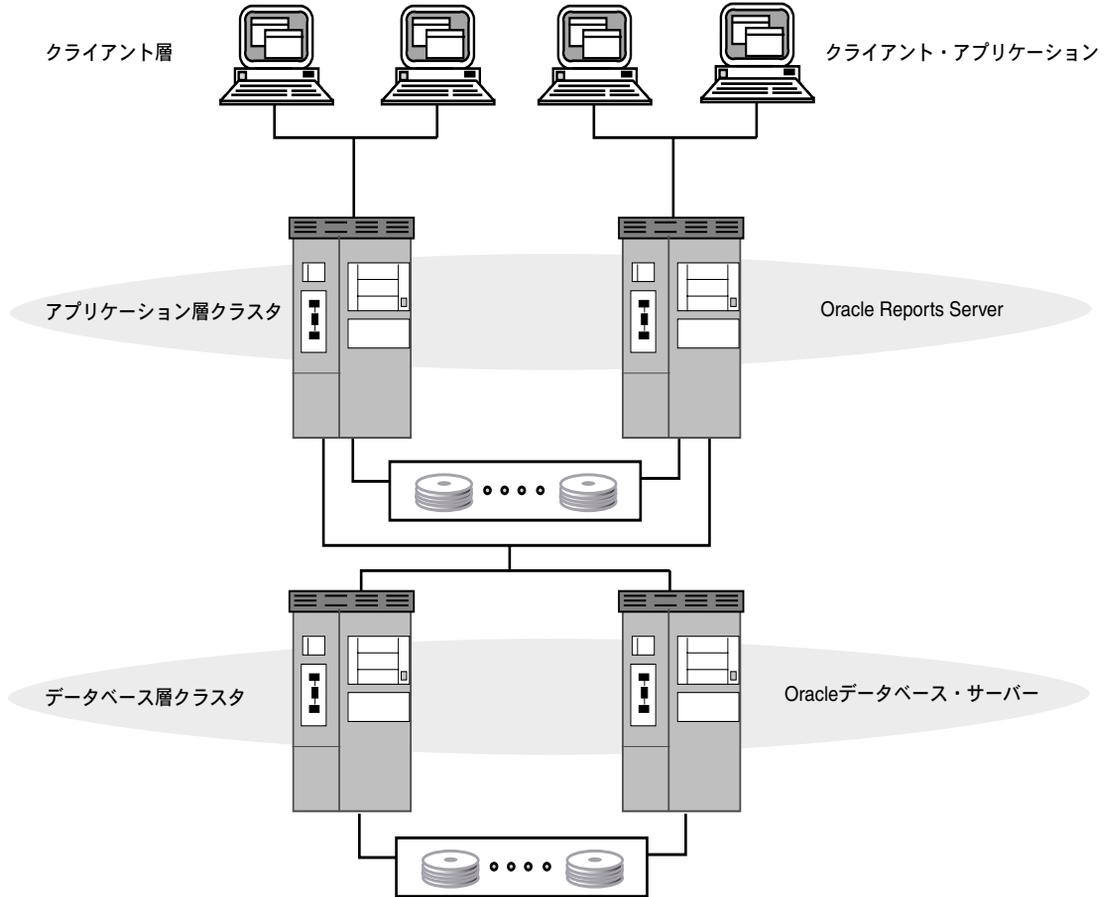
10.1.2 クライアント / サーバー実装

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 10-2 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 10-2 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



10.1.3 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可の場合、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可になります。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、[10.5 項](#)を参照してください。

10.2 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 Services for MSCS 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。

10.3 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リング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、Oracle HTTP Server をグループに追加することによっても可能です。Oracle HTTP Server 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。10.5 項に、構成後に実行する手順について説明しています。

10.3.1 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。

10.3.2 構成手順

Oracle Fail Safe では、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提供されます。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 10-1 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」または「ヘルプ」→「チュートリアル」を選択します。

表 10-1 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	インストールの情報は、Oracle Reports のマニュアルを参照してください。
2	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。
3	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。
4	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
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ため、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。)
7	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 10-1 Oracle Reports Server の構成手順（続き）

手順	処置	説明
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	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（10.4.1 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（10.5 項を参照）。
10	Oracle HTTP Server をフェイルオーバー用に構成	Oracle HTTP Server を作成してグループに追加し、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。Oracle HTTP Server を作成してグループに追加する方法の詳細は、第 11 章を参照してください。

10.3.3 Oracle Reports Server の構成データ

Oracle Fail Safe Manager では、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」ウィザードが提供されます。「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

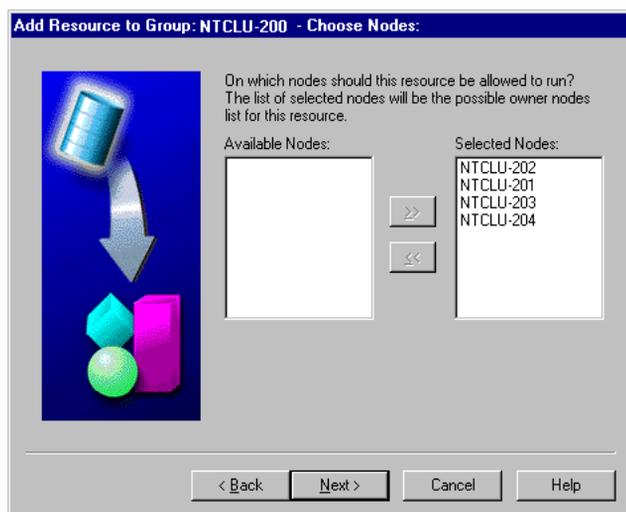
次の項で、これらの要件を詳細に説明します。

10.3.3.1 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 10-3](#) 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。

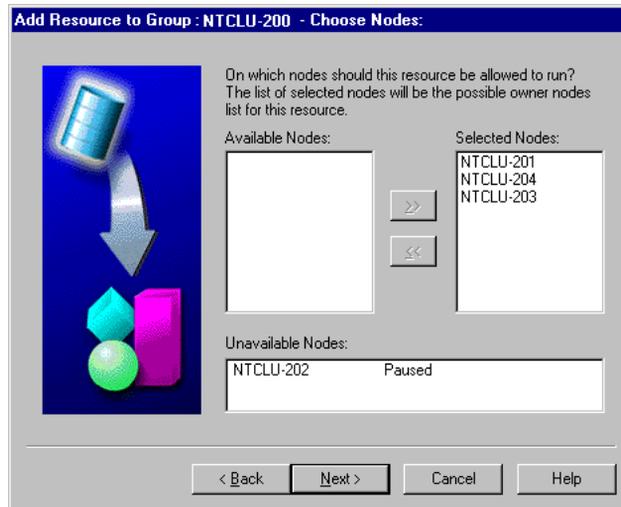
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

図 10-3 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 10-4](#) 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 10-4 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



10.3.3.2 仮想アドレス

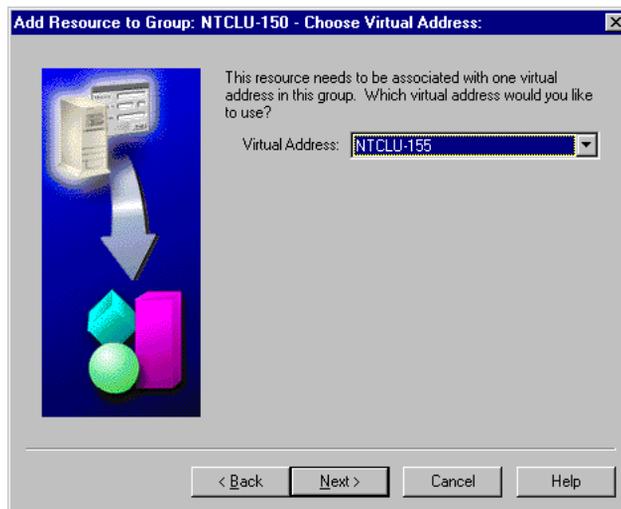
Oracle Reports Server をグループに追加するには、そのグループに少なくとも1つの仮想アドレスが含まれている必要があります。図 10-5 に示すように、Oracle Reports Server を追加するグループ内に複数の仮想アドレスが含まれている場合、Oracle Reports Server と関連付ける仮想アドレスを選択するように要求されます。Oracle Reports アプリケーションにクライアント・アプリケーションからアクセスする場合、アプリケーションから Oracle Reports Server への接続時に指定するホストはその仮想アドレスになります。Oracle 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 Oracle HTTP Server の仮想アドレスとなります。たとえば、Oracle Reports Server と Oracle HTTP Server 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、Oracle HTTP Server 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。

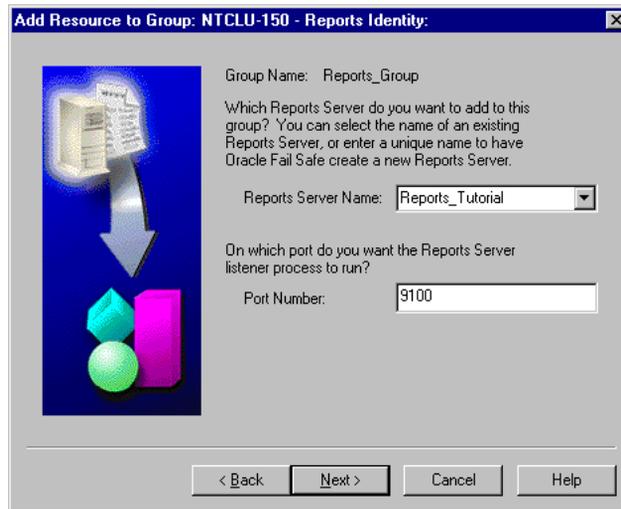
図 10-5 Oracle Reports の「仮想アドレスの選択」ウィザード・ページ



10.3.3.3 Oracle Reports Server の識別情報

図 10-6 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 10-6 「Reports の識別情報」ウィザード・ページ



10.3.3.4 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、これらの値はウィザードの「Reports のディスク」ページで入力されます。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>\Report60\Server\<reports-server-name>.ora 内にあります。

図 10-7 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

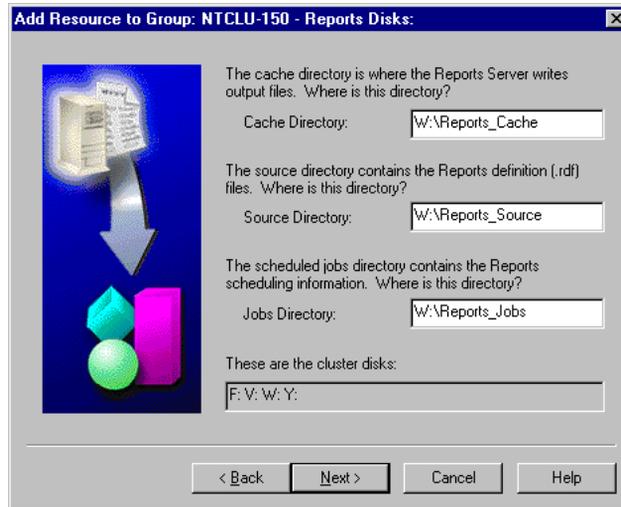
Oracle Reports Server 作業の実行スケジュールを保持する .dat ファイルを配置するディレクトリを指定します。このディレクトリ指定は、クラスタ・ディスクおよびディレクトリを指定します。ジョブをスケジュールしない場合でも、このディレクトリは必ず指定してください。

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かかわらず 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 10-7 「Reports のディスク」 ウィザード・ページ

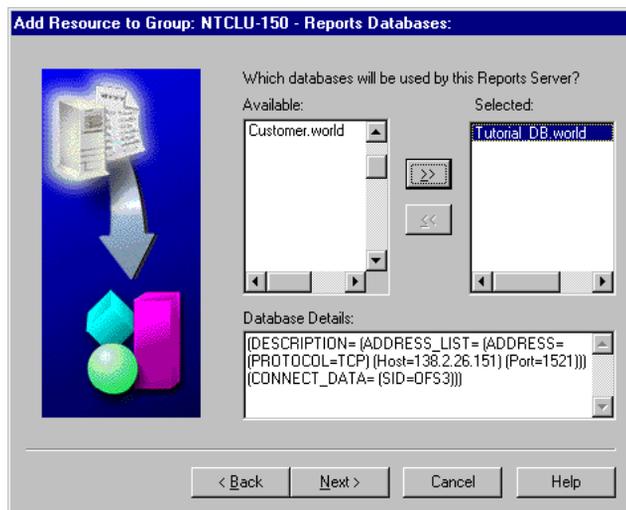


10.3.3.5 Oracle Reports Server によりアクセスされるデータベース

図 10-8 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を使用するかを認識する必要があります。Oracle Fail Safe では、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 tnsnames.ora ファイルを更新するので、Oracle Reports Server またはデータベースが稼働し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 10-8 「Reports のデータベース」ウィザード・ページ

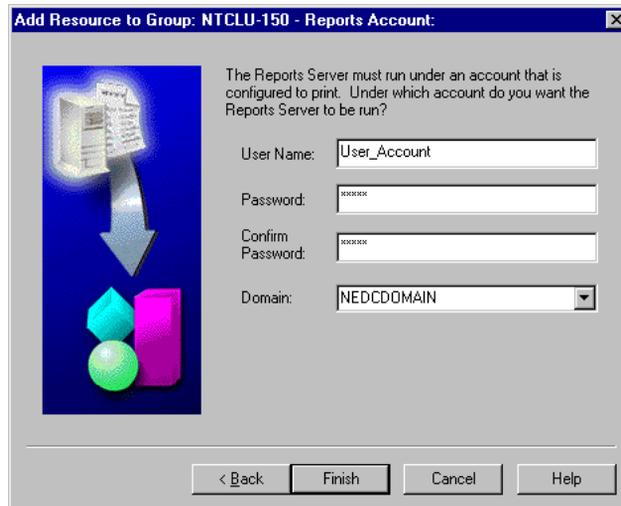


10.3.3.6 Oracle Reports Server の実行に使用されるアカウント

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 10-9 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。

図 10-9 「Reports のアカウント」ウィザード・ページ



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。

10.4 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、Oracle HTTP Server 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 リスナーがリスニング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 Oracle HTTP Server を追加したグループの仮想アドレス、および Oracle HTTP Server 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. 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 (Oracle HTTP Server 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。

10.4.1 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。

- 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。)
- 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
  )
```

10.5 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 (10.5.1 項)
- マスター / スレーブ間の通信を使用可能にする方法 (10.5.2 項)
- 既存の実装へのスレーブ・サーバーの追加 (10.5.3 項)

10.5.1 マスター / スレーブ例の概要

表 10-2 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 10-2 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 つのディレクトリが設定されています。この中央ファイル・サーバーは、グループへの Oracle Reports Server の追加時、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・ディレクトリを指定した際に指定した共有クラスタ・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。

10.5.2 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 を実行可能なシステム上で、`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システムの**ホスト名**ではなく、マスター・サーバーの仮想アドレスを `tnsnames.ora` ファイル内に指定してください。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。)

10.5.2.1 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` ディレクトリにある `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 ディレクトリにある 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 1 回オフライン化した後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンライン化します。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 (tnsname.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せん。
- 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。
- スレーブ・サーバーでは、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。

10.5.2.2 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 ディレクトリにある 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 ディレクトリにある 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。これにより、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・エンジンのみになります。
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>¥Report60¥Server ディレクトリにある sun-1.ora 構成ファイルを編集してください。

10.5.3 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 tnsname.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 10-2 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。

10.5.3.1 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 1 回オフライン化した後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンライン化します。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 (tnsname.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。

10.5.3.2 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 ディレクトリにある 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 ディレクトリにある 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。(これにより、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・エンジンのみになります。) 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。

10.6 可用性の高い Oracle Reports Server へのユーザー・アクセスの制御

Oracle Reports では、Oracle Reports へのユーザー・アクセス制御に Oracle Portal 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 Portal を使用します。Web 上にレポートを公開する場合は、Oracle Reports Server のマニュアル (Oracle9i Applications Server のドキュメント・ライブラリの一部) 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 Portal データベース・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 Portal データベース・インスタンスの可用性を向上させます。(Oracle Portal データベース・インスタンスをグループに追加します。)

2. (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。

```
SECURITYINSNAME="sec-rep"
```

この例で sec-rep は、Oracle Portal データベース・サーバー・インスタンスの追加先のグループの仮想アドレスです。

3. Oracle Reports Server を 1 回オフライン化した後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンライン化します。これを実行するまで、新しい構成は有効になりません。
4. (手順 2 で説明したように) Oracle Reports Server の可能所有者である他のクラスタ・ノードで report-server.ora 構成ファイルを編集します。

10.7 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リングできます。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 では、ジョブの実行時に Oracle Reports Server キュー・アクティビティのスナップショットを必ず撮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リングされている 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、スケジューリング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リングしており、再試行の回数が上限に達していない場合にかぎられます。

10.7.1 可用性の高い Oracle Reports Server でのジョブ・リポジトリの使用方法

Oracle Reports 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撮るようにデータベースを設定する機能を提供します。これにより、ユーザーは発行したすべてのレポートの状態を把握できます。また、管理者は Reports Server の同時ユーザーの数を把握できます。これは、環境内の使用状況を把握し、ライセンスへの準拠を確実にするという両面で役立ちます。

Oracle Reports Server では、ジョブ要求が発行されるたびに、データベースに最新のレポート・キューを転送します。この情報は RW_SERVER_QUEUE という表に追加され、次のデータが挿入されます。

- ジョブの名前
- ジョブの発行者
- ジョブに対して選択された出力フォーマット
- ジョブの現行の状態

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 のマニュアルの指示（レポート・キューを所有し、サーバー・キュー API の実行権限を持つスキーマの作成を含む）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. 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 のマニュアルに記述されている指示に従ってください。
3. 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 を繰り返します。
4. 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。

10.8 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。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 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、これらのツールの説明を記載しています。

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

10.8.1 Oracle HTTP Server で 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、次の操作を実行します。

- Oracle HTTP Server 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。

10.8.2 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。

10.8.3 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、次の操作を実行します。

- 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、太字の部分サーバー名を示します。

```
<A HREF=".../rwcgi60.exe?server=demoreportserver+report=...">
```

サーバー名は tnsnames.ora ファイルに定義する必要があります。

10.8.4 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、次の操作を実行します。

- 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 Oracle Reports 名を示します。

```
<A HREF="...?server=...+report=employees+userid...">
```

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。)

10.8.5 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、次の操作を実行します。

- レポートを Reports ランタイムでローカルに実行します。
- 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。

レポートをローカルで実行した後、再び Web ブラウザで実行しようとしたときに発生した問題をすべて修正します。

10.8.6 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、次の操作を実行します。

- 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 Oracle HTTP Server の構成

Oracle Fail Safe は、可用性の高い Oracle HTTP Server (powered by Apache) を構成する際に役立ちます。Oracle HTTP Server は、Oracle9i Application Server (Oracle9iAS) のコンポーネントです。

クラスタ・ノードで、停止または障害発生という事態が発生すると、Oracle HTTP Server 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、クライアントは、通常は数秒以内にサーバーの 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 Oracle HTTP Server 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。

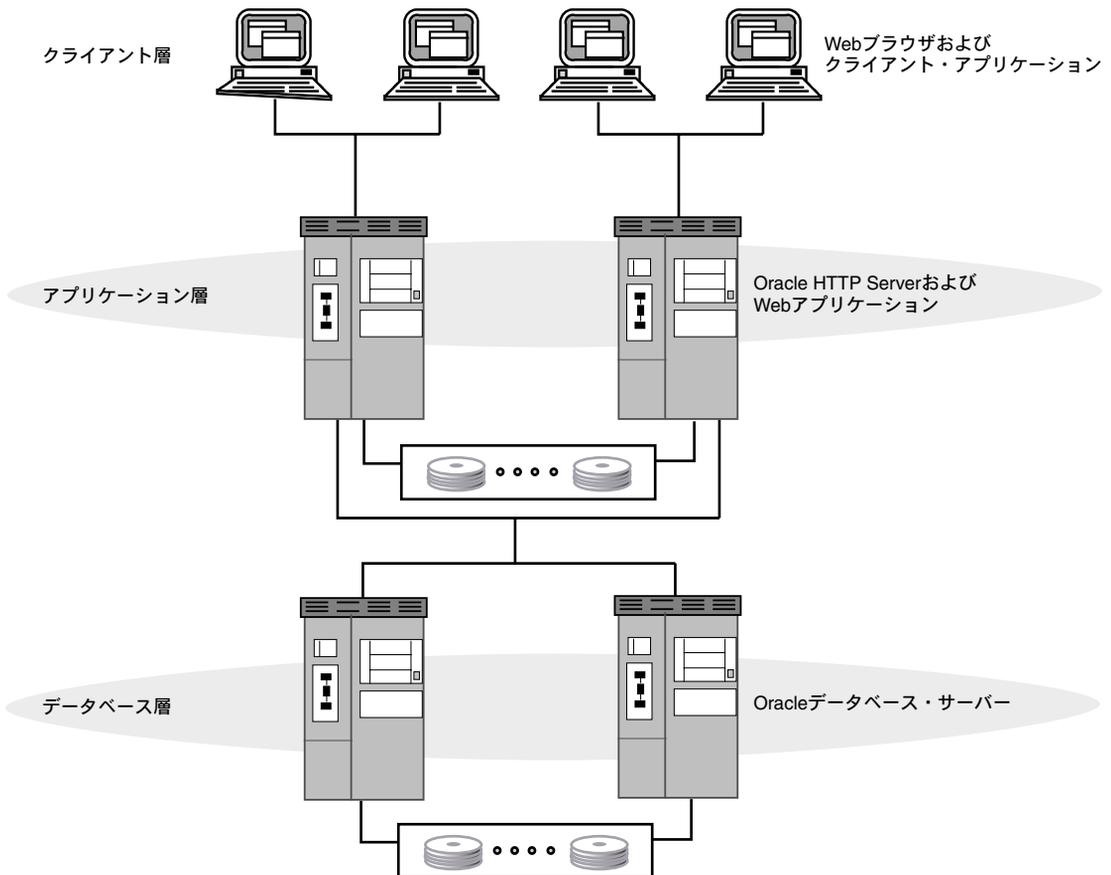
この章では、次の項目について説明します。

項目	参照
概要	11.1 項
スタンドアロン Oracle HTTP Server の検出	11.2 項
Oracle HTTP Server のグループへの追加	11.3 項
Oracle HTTP Server へのクライアント接続	11.4 項
Oracle HTTP Server のグループからの削除	11.5 項
Oracle HTTP Server のセキュリティ要件	11.6 項
Oracle HTTP Server に関連する問題のトラブルシューティング	11.7 項

11.1 概要

図 11-1 に、アプリケーション層に可用性が高まるように Oracle HTTP Server および Web アプリケーションを構成し、データベース層にシングルインスタンス 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ーを示します。

図 11-1 可用性の高い Web 構成のための 3 層アーキテクチャ



多数の冗長周辺機器がロード・バランス Web サーバーを介してアクセスされるアーキテクチャによって、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、Oracle HTTP Server と次のような Web アプリケーションの可用性が向上します。

- Oracle Applications Concurrent Manager
- Oracle Forms
- Oracle Reports
- 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
- Oracle HTTP Server 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、**Common Gateway Interface (CGI)**、Perl、Active Server Pages (ASP)、Visual Basic または JavaScript などを使用する Web アプリケーションを、Oracle Fail Safe を使用して可用性を高めるように構成できます。

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。

11.2 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 Services for MSCS 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、各エントリに関連付けられたイメージ名を探し、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。

11.3 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

11.3.1 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。

11.3.2 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 11-1 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」または「ヘルプ」→「チュートリアル」を選択します。

表 11-1 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ー・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	インストールの情報は、Oracle HTTP Server のマニュアルを参照してください。
2	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ットのクラスター・ディスク上に配置する必要があります。 ¹
3	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
4	クラスターの検証	「トラブルシューティング」→「クラスターの検証」を選択して、クラスター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。
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ため、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 11-1 Oracle HTTP Server の構成手順 (続き)

手順	処置	説明
6	Oracle HTTP 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。
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。

¹ 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)。

11.3.3 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 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ー・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 (セキュリティ保護) 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。

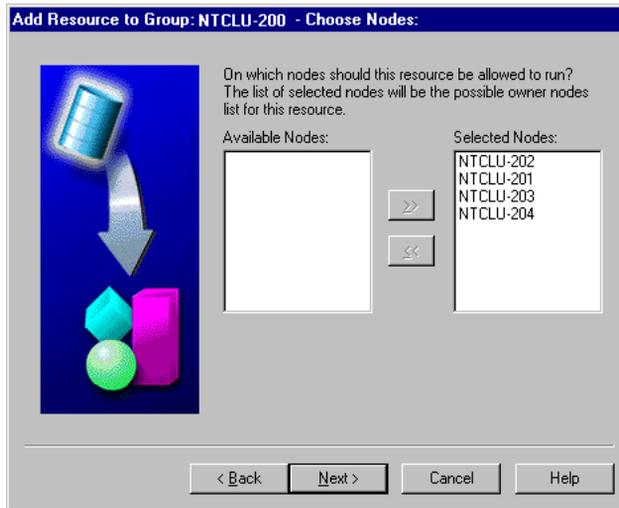
次の項で、これらの要件を詳細に説明します。

11.3.3.1 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 11-2](#) 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。

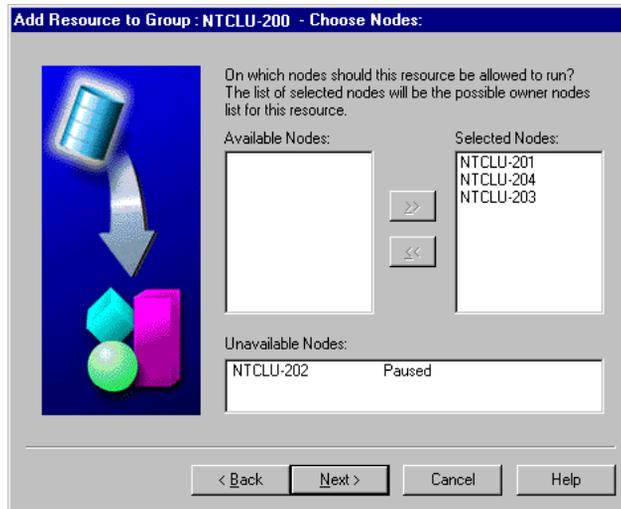
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

図 11-2 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 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かを指定するように求められます。このような場合、[図 11-3](#) 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 11-3 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



11.3.3.2 Oracle HTTP Server の識別情報

[図 11-4](#) 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 がオンライン化またはオフライン化するときに、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 のバージョンおよび Oracle ホーム
Oracle HTTP Server がインストールされた Oracle ホームのバージョン番号と名前。
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。

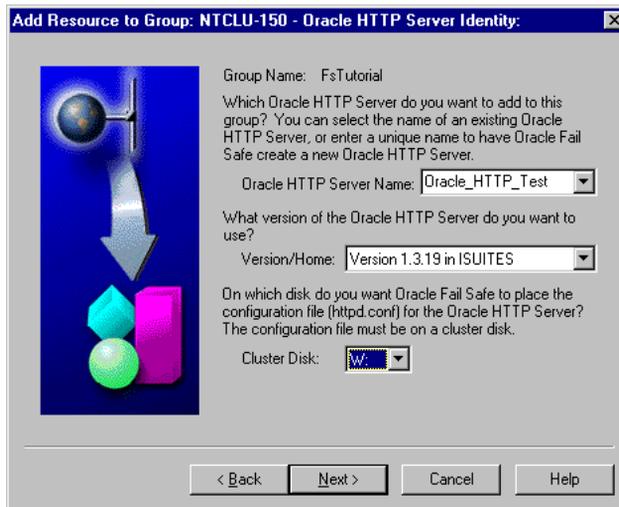
- クラスタ・ディスク

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。

注意： Oracle Fail Safe Manager を Oracle Fail Safe Server のリリース 3.1.2 以下とともに使用する場合は、クラスタ・ディスク名および 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 W:¥ を選択すると、Oracle Fail Safe では構成ファイルを W:¥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 W:¥Apache¥htdocs となります。

Oracle Fail Safe Manager を、リリース 3.2.1 以上の Oracle Services for MSCS とともに使用する場合は、次に示すウィザード・ページでドキュメント・ルート・ディレクトリを明示的に指定します。

図 11-4 「Oracle HTTP Server の識別情報」ウィザード・ページ



11.3.3.3 Oracle HTTP Server のディレクトリ

図 11-5 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 HTTP Server のドキュメント・ルート・ディレクトリ、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。

- ドキュメント・ルート

ドキュメント・ルートは、ドキュメントを処理するディレクトリを示します。

既存の Oracle HTTP Server のリソースをグループに追加する場合、「ドキュメント・ルート」フィールドに、サービスで使用されている `httpd.conf` ファイルの現在の設定が表示されます。

Oracle HTTP Server のリソースを新規に作成する場合は、このフィールドにはドキュメント・ルート指定のための構文が表示されます。

ドキュメント・ルートには、クラスタ・ディスクあるいはクラスタ化されていないディスクのどちらでも指定できます。

注意： Oracle Fail Safe Manager で接続するサーバー・ソフトウェアが Oracle Fail Safe Server リリース 3.1.2 以下の場合、ウィザードにはドキュメント・ルートに関する入力項目がありません。ドキュメント・ルート・ファイルに使用するディスクおよびディレクトリを決定する方法の詳細は、[11.3.3.2 項](#)を参照してください。

- 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。ディレクトリ索引ファイル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、**仮想サーバー**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\Forms60\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。図 11-6 に示すように、「追加」をクリックすると、Oracle Fail Safe では「仮想ディレクトリのマップ」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 11-5 「Oracle HTTP Server のディレクトリ」ウィザード・ページ

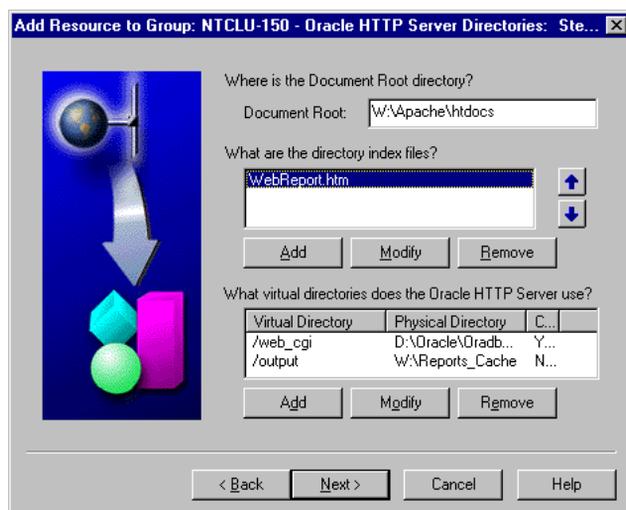
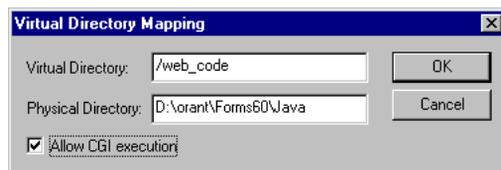


図 11-6 「仮想ディレクトリのマップ」ダイアログ・ボックス



11.3.3.4 Oracle HTTP Server の仮想アドレス

図 11-7 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、Oracle HTTP Server 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（セキュリティ保護）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。

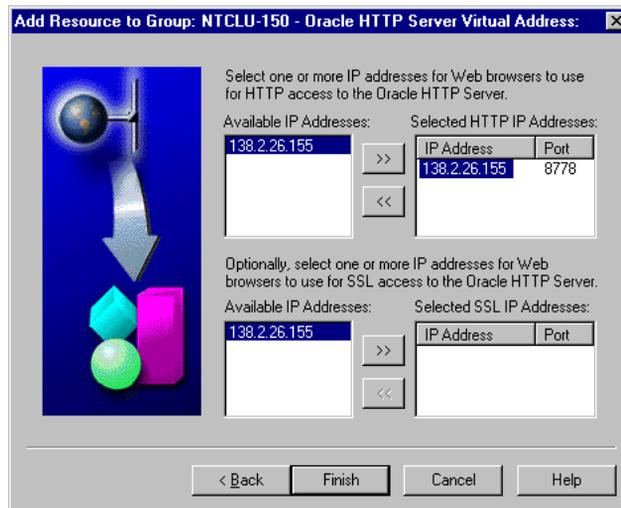
ウィザードに表示される IP アドレスは、仮想アドレスです。アドレスを「使用可能な IP アドレス」リストから「選択済の HTTP IP アドレス」リストあるいは「選択済の SSL IP アドレス」リストへ移動すると、Oracle Fail Safe ではウィンドウが開き、Oracle HTTP Server で選択済アドレスをリスニングするポートを指定するように要求されます。

（「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。次に例を示します。

```
<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 11-7 「Oracle HTTP Server の仮想アドレス」 ウィザード・ページ



11.4 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
```

11.5 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。

11.6 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。

11.7 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。

11.7.1 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、次の操作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。

```
<cluster-disk>:%Apache%conf%httpd_<resource-name>.conf
```

この例では、<cluster-disk> は、「リソースをグループに追加」ウィザードの「Oracle HTTP Server の識別情報」ページで指定したクラスタ・ディスクです。

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 <cluster-disk>:%Apache%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。

11.7.2 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、次の操作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。
- Forms ランタイムや Reports ランタイムなどのツールを使用してドキュメントまたはその他のファイル（フォーム、レポートなど）への直接アクセスを試行し、検出された問題を解決します。

問題を解決した後、Oracle HTTP Server でドキュメントまたはその他のファイルへのアクセスを再試行します。

11.7.3 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、次の操作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。

11.7.4 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、次の操作を実行します。

- 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 を構成する際に役立ちます。クラスタ・ノードで、停止または障害発生という事態が発生すると、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します。

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、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 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。

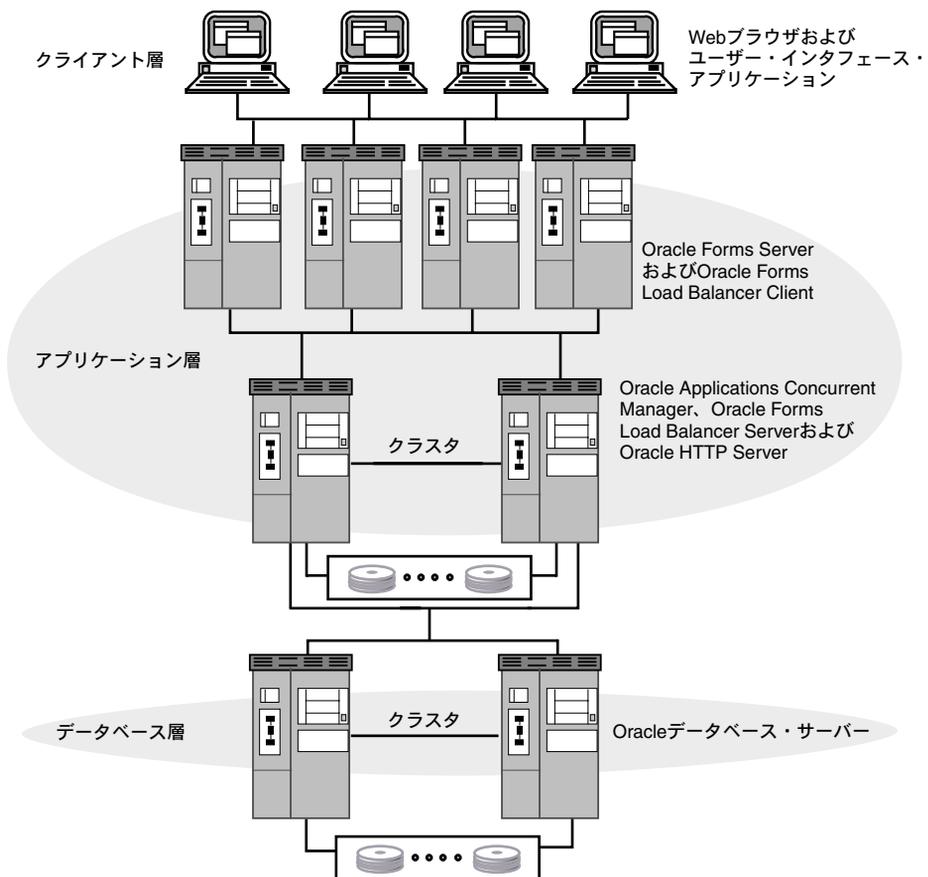
項目	参照
概要	12.1 項
スタンドアロン Oracle Applications Concurrent Manager の検出	12.2 項
Oracle Applications Concurrent Manager のグループへの追加	12.3 項
Oracle Applications Concurrent Manager のセキュリティ要件	12.4 項
Oracle Applications Concurrent Manager に関連する問題のトラブルシューティング	12.5 項

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、Oracle HTTP Server など）の可用性も向上させる必要があります。

12.1 概要

図 12-1 は、可用性を高めるためにアプリケーション層で Oracle Applications Concurrent Manager、Oracle HTTP Server および Oracle Forms Load Balancer Server が構成され、データベース層で Oracle データベースが構成され、クライアント層に Web ブラウザおよびユーザー・インタフェース・アプリケーションがインストールされた 3 層アーキテクチャを持つ、2 ノード・クラスタを示しています。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 がアプリケーション層にインストールされています。

図 12-1 Oracle Applications Concurrent Manager の 3 層アーキテクチャ



12.2 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 Services for MSCS では内部 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、ccmsvc.exe の関連イメージ名のエントリを探し、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。

12.3 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をリスニングする Oracle 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 Oracle HTTP Server (オプション)

12.3.1 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。(Oracle Fail Safe でも、Oracle Applications Concurrent Manager と可用性の高いデータベースで使用される Oracle Net ネットワーク・リスナーの間に、適切な依存性が設定されます。)

- 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\11.5.0\<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。

12.3.2 構成手順

Oracle Fail Safe を使用して内部 Oracle Applications Concurrent Manager の可用性を高めるために必要な手順は、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 12-1 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 12-1 Oracle Applications 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は、Oracle Applications 11i のマニュアルを参照してください。
2	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
3	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。
4	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ため、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。）
5	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。
6	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。
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、アプリケーション全体の可用性を向上させます。

12.3.3 Oracle Applications Concurrent Manager の構成データ

Oracle Fail Safe Manager では、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」ウィザードが提供されます。「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 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 Applications Concurrent Manager と関連付ける仮想アドレス
- 内部 Oracle Applications Concurrent Manager の名前、アプリケーション構成 ID、ログおよび出力ファイルを格納する共有クラスタ・ディスク上のディレクトリ
- 内部 Oracle Applications Concurrent Manager を実行する Windows 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 Windows 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適切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。

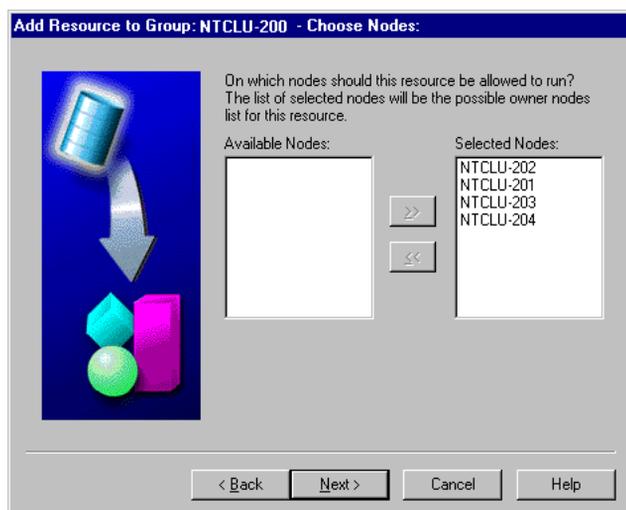
次の各項では、内部 Oracle Applications Concurrent Manager をグループに追加するためのデータ要件を詳細に説明します。

12.3.3.1 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 12-2](#) 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。

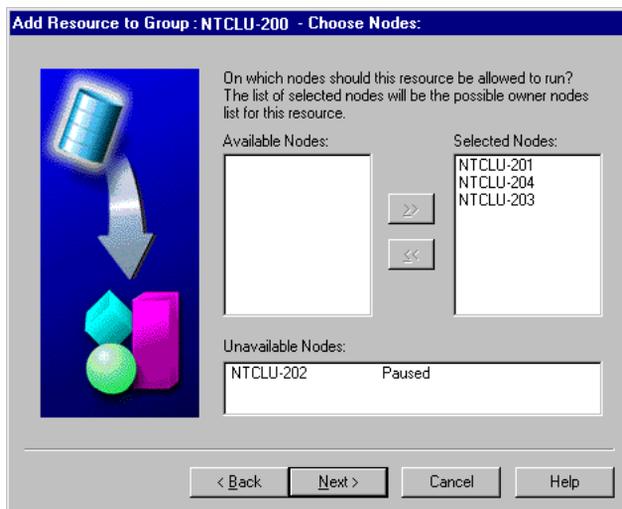
[2.6.6 項](#)では、「可能所有者ノード」リストの概念の詳細を説明します。

図 12-2 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かを指定するように求められます。このような場合、図 12-3 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 12-3 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

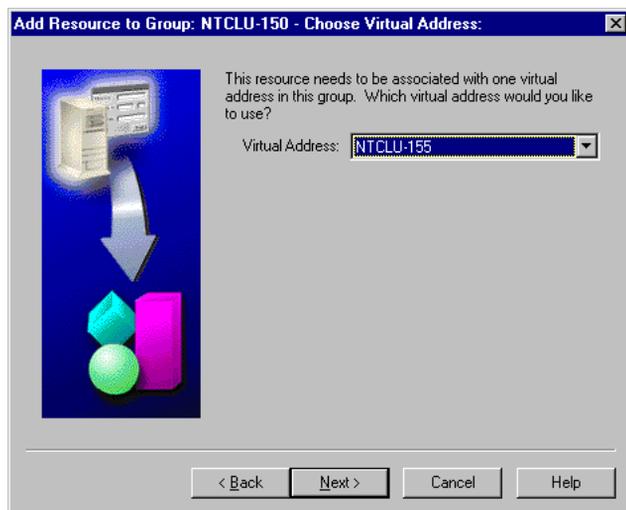


12.3.3.2 Oracle Applications Concurrent Manager 仮想アドレス

内部 Oracle Applications Concurrent Manager を追加する前に、グループは少なくとも1つの仮想アドレスを持っている必要があります。図 12-4 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 12-4 Oracle Applications Concurrent Manager の「仮想アドレスの選択」ウィザード・ページ



12.3.3.3 Concurrent Manager 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、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:¥Apps_Logs と指定した場合、ログ・ファイルは、X:¥Apps_Logs¥log に置かれ、出力ファイルは X:¥Apps_Logs¥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>¥APPLCSF
```

図 12-5 「Concurrent Manager の識別情報」ウィザード・ページ

Add Resource to Group: NTCLU-150 - Concurrent Manager Identity:

Group Name: apps_oas_prod

Which Concurrent Manager do you want to add to this group? You can select the name of an existing Concurrent Manager, or enter a unique name to have Oracle Fail Safe create a new Concurrent Manager.

Concurrent Manager Name:

What is the configuration ID for this Concurrent Manager?

Configuration ID:

In what directory do you want the log and output files to be stored? These files must be stored on a shared disk.

Directory:

< Back Next > Cancel Help

12.3.3.4 Concurrent Manager の認証

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、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、Windows 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。
- 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。
- 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 12-6 「Concurrent Manager の認証」ウィザード・ページ

Add Resource to Group: NTCLU-150 - Concurrent Manager Authentication:

The Concurrent Manager must run under an administrator's account. Under which account do you want the Concurrent Manager to run?

User Name:

Password:

Confirm Password:

Domain:

< Back Next > Cancel Help

12.3.3.5 Concurrent Manager のデータベース

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、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。

- 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、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。

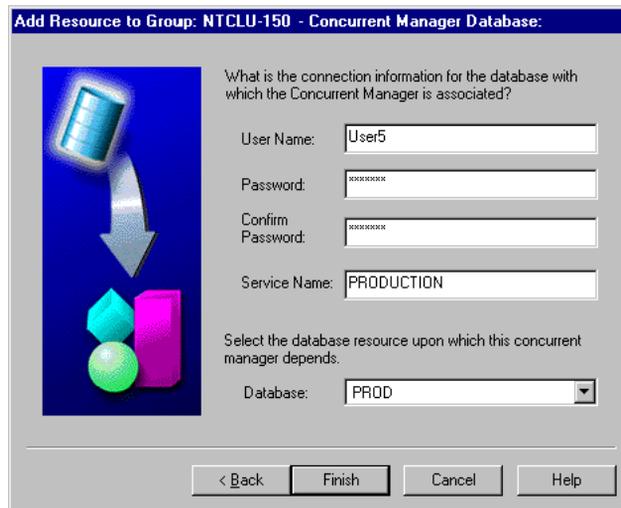
- 内部 Oracle Applications Concurrent Manager の追加先であるグループに Concurrent Manager の接続先のデータベースが含まれている場合
- データベース・リソース名がデータベース・サービス名とは異なる場合

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 12-7 「Concurrent Manager のデータベース」 ウィザード・ページ



Add Resource to Group: NTCLU-150 - Concurrent Manager Database:

What is the connection information for the database with which the Concurrent Manager is associated?

User Name:

Password:

Confirm Password:

Service Name:

Select the database resource upon which this concurrent manager depends.

Database:

< Back Finish Cancel Help

12.4 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ー・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。

12.5 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 Windows 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と、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。

12.5.1 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・ノードに存在すること

12.5.2 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、次の操作を実行します。

- 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 (使用不可、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・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・システムに多大な負荷がかかり、内部 Oracle Applications Concurrent Manager があたかも使用不可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、次のようにリソース・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可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。

12.5.3 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。

12.5.4 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 Services for Microsoft Transaction Server (Oracle MTS Service と呼びます) を構成する際に役立ちます。クラスタ・ノードが停止または障害発生という状態になると、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。

注意： 可用性が高まるよう Oracle Services for Microsoft Transaction Server を構成するためのサポートは、Oracle データベース・サーバー 8.1.6 および 8.1.7 で提供されています。Oracle9i 以降、Oracle データベース・サーバーが Microsoft Transaction Server とともにトランザクションを処理する場合に、Oracle MTS Service は使用されません。Oracle データベース・サーバーが Microsoft Transaction Server のトランザクションに参加する際に必要なコードおよびロジックは、すべて MTS アプリケーション・プロセスに組み込まれています。このため、この章に記載する説明は Oracle サーバー 8.1.6 および 8.1.7 のみに該当します。

この章では、次の項目について説明します。

項目	参照
概要	13.1 項
スタンドアロン Oracle MTS Service の検出	13.2 項
Oracle MTS Service のグループへの追加	13.3 項
Oracle MTS Service のセキュリティ要件	13.4 項
Oracle MTS Service に関連する問題のトラブルシューティング	13.5 項

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。

Oracle データベースに接続する COM コンポーネントは、特別なデータベース統合を必要とせず、Microsoft Transaction Server 内に配布できます (サポートされるバージョンのリストは、『Oracle Fail Safe for Windows 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 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。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。

13.1 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。

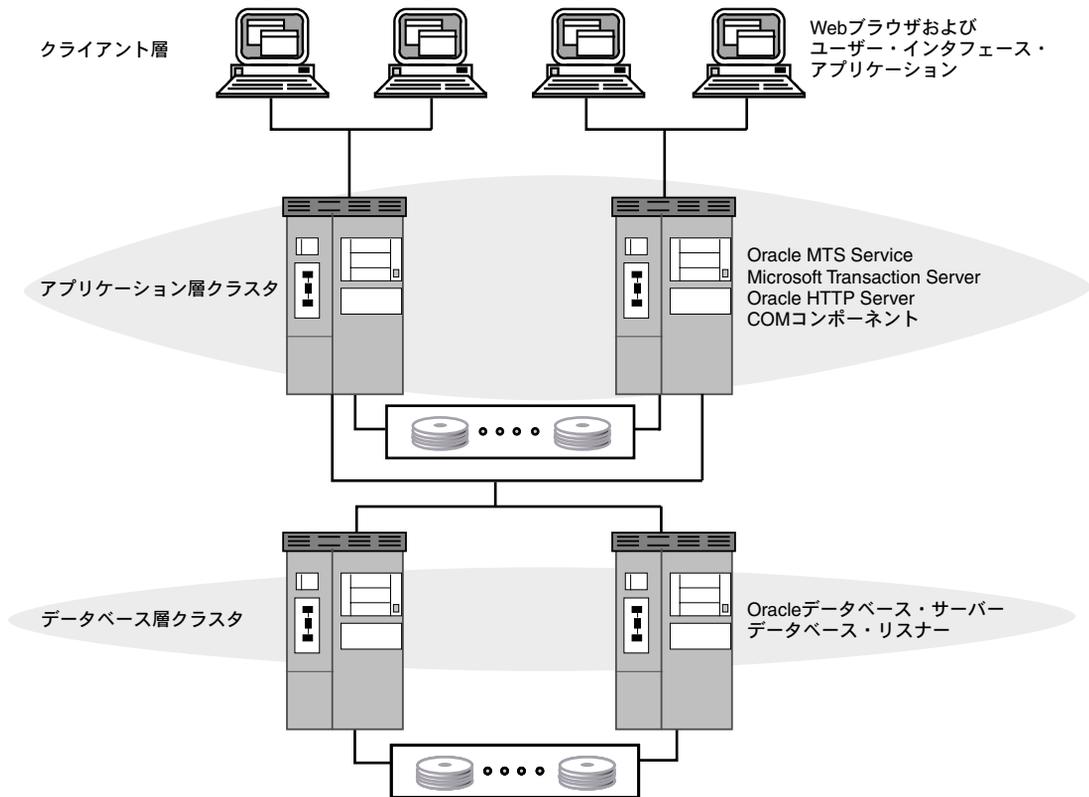
図 13-1 に、Microsoft Transaction Server、Oracle MTS Service、Oracle HTTP Server および COM コンポーネントがデータベースとは別のクラスタに常駐する階層化実装を示します。

図 13-1 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、Oracle HTTP Server 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 13-1 可用性の高い Oracle MTS Service のための 3 層アーキテクチャ



13.2 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 Services for MSCS 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、Oracle MTS Service エントリを探し、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。

13.3 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 Distributed Transaction Coordinator (MS DTC) 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 Distributed Transaction Coordinator
- 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 Distributed Transaction Coordinator 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 Distributed Transaction Coordinator 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 Distributed Transaction Coordinator を含むグループに追加される必要があります。

13.3.1 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 Distributed Transaction Coordinator (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 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 for Windows 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。

13.3.2 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 13-1 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 13-1 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 Distributed Transaction Coordinator (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	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。

表 13-1 Oracle MTS Service を構成する手順 (続き)

手順	処置	説明
4	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
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	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。
7	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	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。(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。)
8	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。
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、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 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。

- Oracle MTS Service 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)
- 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 のログオ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の不適切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。

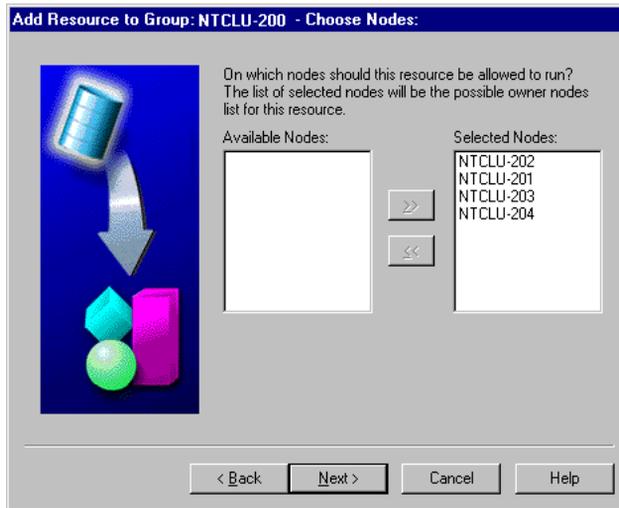
次の各項では、Oracle MTS Service をグループに追加するためのデータ要件を詳細に説明します。

13.3.3.1 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 13-2](#) 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。

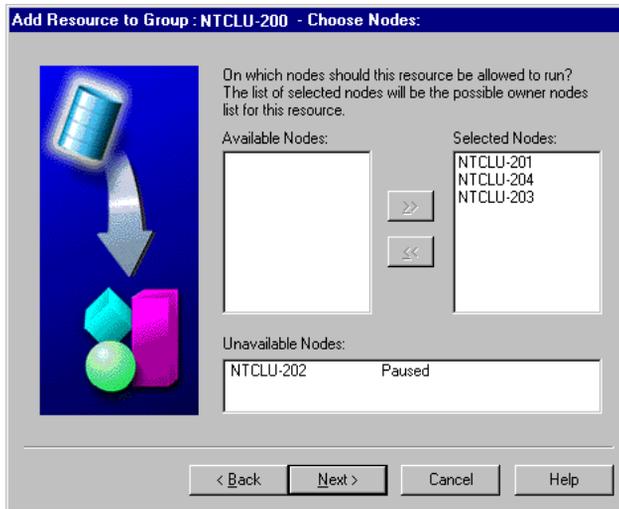
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

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 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 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。

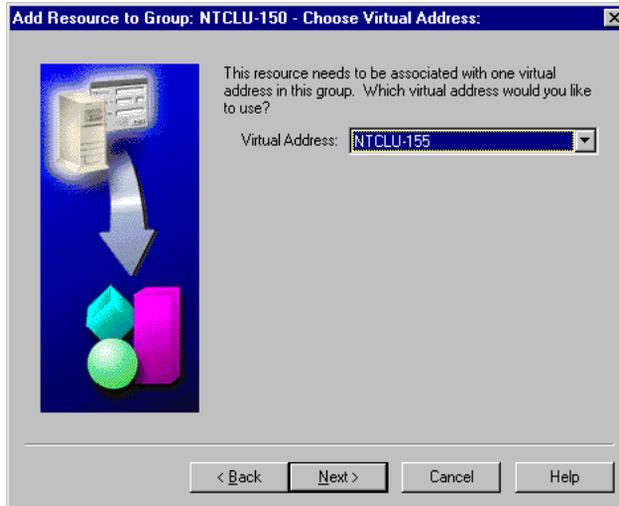
図 13-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



13.3.3.2 Oracle 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 13-4 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 13-4 Oracle MTS Service の「仮想アドレス」ウィザード・ページ



13.3.3.3 Oracle MTS Service 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、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

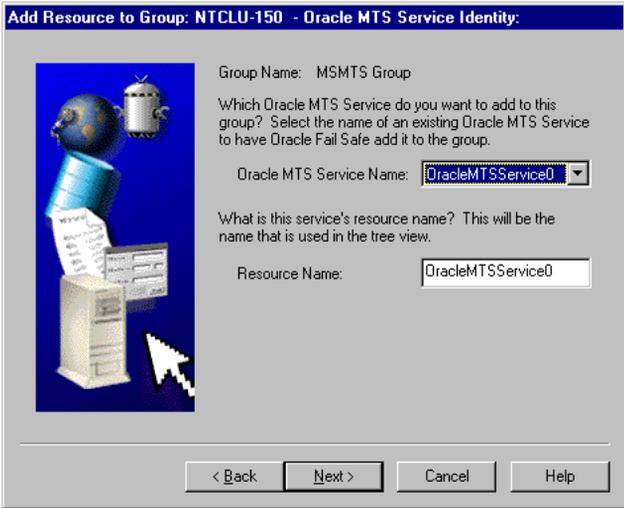
これは、クラスタ上の既存の Oracle MTS Service の名前です。提供される名前のリストは、Oracle Fail Safe でクラスタ上のスタンダアロン Oracle MTS Service として識別されている名前のリストです。

- 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 13-5 「Oracle MTS Service の識別情報」ウィザード・ページ



Add Resource to Group: NTCLU-150 - Oracle MTS Service Identity:

Group Name: MSMTS Group

Which Oracle MTS Service do you want to add to this group? Select the name of an existing Oracle MTS Service to have Oracle Fail Safe add it to the group.

Oracle MTS Service Name: OracleMTService0

What is this service's resource name? This will be the name that is used in the tree view.

Resource Name: OracleMTService0

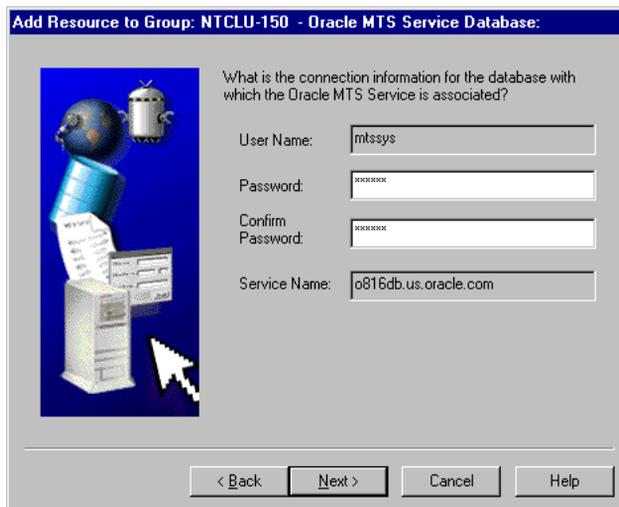
< Back Next > Cancel Help

13.3.3.4 Oracle MTS Service データベース

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、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。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 は、常に特定のデータベースに対して作成されます。

図 13-6 「Oracle MTS Service のデータベース」ウィザード・ページ



Add Resource to Group: NTCLU-150 - Oracle MTS Service Database:

What is the connection information for the database with which the Oracle MTS Service is associated?

User Name:

Password:

Confirm Password:

Service Name:

< Back Next > Cancel Help

13.3.3.5 Oracle MTS Service の認証

「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 13-7](#) 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「LocalSystem」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 13-7 「Oracle MTS Service の認証」ウィザード・ページ

Add Resource to Group: NTCLU-150 - Oracle MTS Service Authentication:

Specify a Windows account with which the service can log on.

User Name:

Password:

Confirm Password:

< Back Finish Cancel Help

13.4 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

13.5 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に関する一般的な情報は、Oracle8i for Windows のマニュアルの一部である『Oracle 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 の可能所有者であるノードから、Oracle MTS Service を誤って削除した場合。「グループの検証」操作により、Oracle MTS Service 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と、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 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 *resource-name* の構成に失敗しました。

FS-10618: Windows サービス *service-name* 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。

13.5.2 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化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。

13.5.3 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 によりサポートされる 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。

- 所有する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 シングルインスタンス・データベース、Oracle Forms Server、Oracle Forms Load Balancer Server、Oracle Reports Server、Oracle Intelligent Agent、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。(14.3 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	14.1 項
スタンドアロン汎用サービスの検出	14.2 項
汎用サービスのグループへの追加	14.3 項
汎用サービスのセキュリティ要件	14.4 項
サンプル汎用サービスの構成	14.5 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	14.6 項

14.1 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、「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、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、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や、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。

14.1.1 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みのため、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 で、あるサービスのためのウィザードがカスタマイズされていない場合、「リソースをグループに追加」ウィザード内の汎用サービス・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。Oracle Fail Safe Manager のツリー・ビューでクラスタを選択して「リソース」タブをクリックすると、Oracle Fail Safe でウィザードがカスタマイズされているサービスがわかります。ウィザードがカスタマイズされているリソースがこのプロパティ・ページに一覧表示されます。

14.1.2 可用性を高める構成にしない汎用リソース

可用性を高めるために汎用サービスを構成すると、そのサービスは一度に1つのクラスタ・ノード上でしか実行できなくなるので、そうした構成を行う場合は注意してください。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。

14.2 スタンドアロン汎用サービスの検出

Oracle Services for MSCS は、Windows サービス マネージャで検索することによって汎用リソースを検出します。検出プロセスの際、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。

14.3 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。

14.3.1 構成手順

表 14-1 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。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 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	汎用サービスでデータ・ファイルが必要な場合、共有記憶装置インターコネクタ上のクラスタ・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、「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」を選択します。
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せん。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、14.3.2.5.2 項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。

14.3.2 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）
- 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

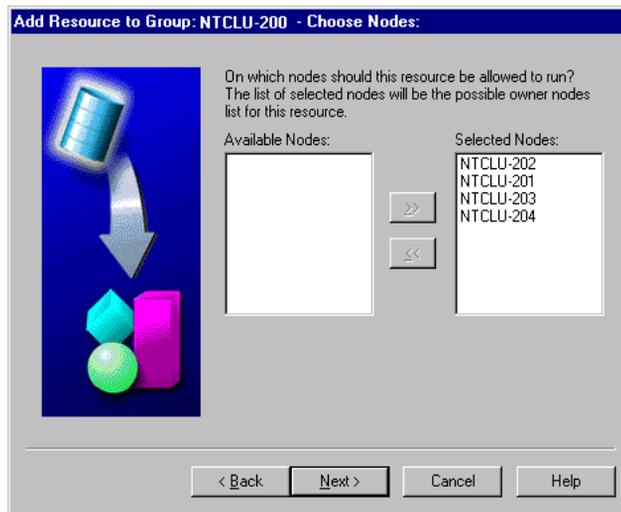
他のリソースを可用性が高まるように構成する場合と異なり、汎用サービスを追加する前には仮想アドレスをグループに追加するように要求されません。汎用サービスの使用方法に応じて、仮想アドレスが必要かどうかを自分で判断する必要があります。次の各項では、仮想アドレスをグループに追加する必要性を判断する際に考慮すべき問題と、汎用リソースをグループに追加するために必要な構成情報について検討します。

14.3.2.1 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 14-1](#)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。

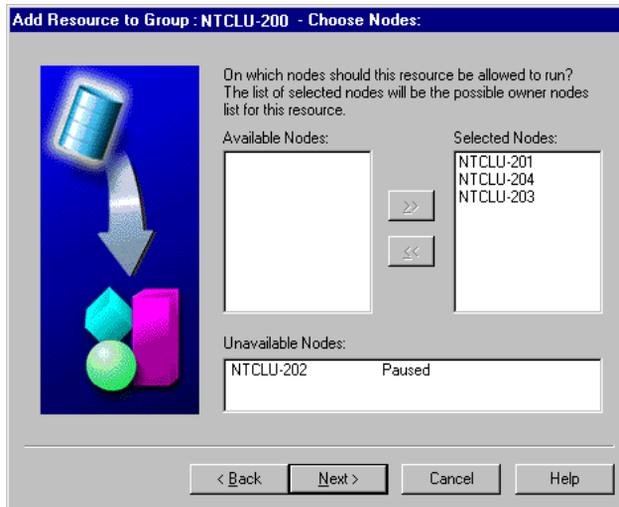
2.6.6 項では、「可能所有者ノード」リストの概念の詳細を説明します。

図 14-1 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 14-2 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 14-2 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



14.3.2.2 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。汎用サービスが複数のノードに置かれている場合、それらのいずれか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。汎用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となる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、スペースを含めた文字を 256 文字まで指定できます。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インストールする必要があります。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れます。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 14-3 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。「サービス名」ボックスに既存のサービスを入力する場合、「状態」ボックスにそのサービスの状態が表示されます。「サービス名」ボックスに新規サービスを入力する場合、「状態」ボックスには何も表示されません。Oracle Fail Safe Manager により表示される状態は、確認のためのものです。既存の汎用リソースは、実行されているか停止されているかにかかわらず、グループに追加できます。

図 14-3 「汎用サービスの識別情報」ウィザード・ページ

14.3.2.3 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

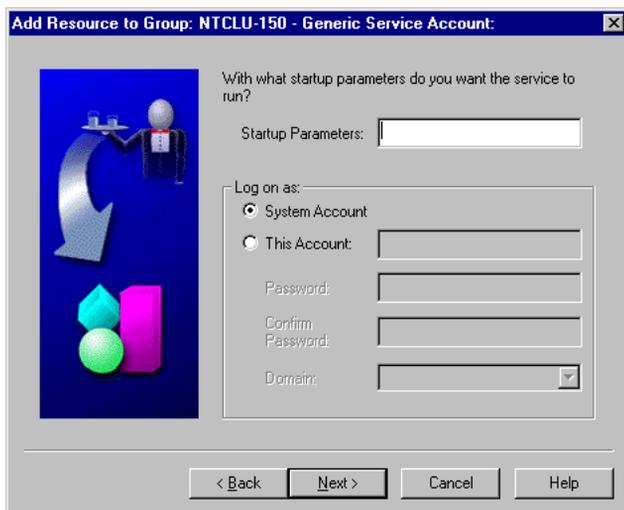
(LocalSystem) としてログオンする場合、サービスは、ローカル・システム上のすべてのファイルにアクセスできますが、ネットワーク全体のファイルにはアクセスできません。サービスがユーザー・アカウントとしてログオンする場合、そのサービスが持つ権限に応じて、ローカル・システム上のファイルとネットワークにわたるファイルの両方にアクセスできます。たとえば、Oracle Fail Safe 自体は、すべてのクラスタ・ノードのファイルにアクセスできる必要があるため、(Oracle Fail Safe をインストールしたときに指定した) ユーザー・アカウントで実行されます。

「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 Services for MSCS 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「使用不可」)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 14-4 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 14-4 「汎用サービスのアカウント」ウィザード・ページ



14.3.2.4 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。

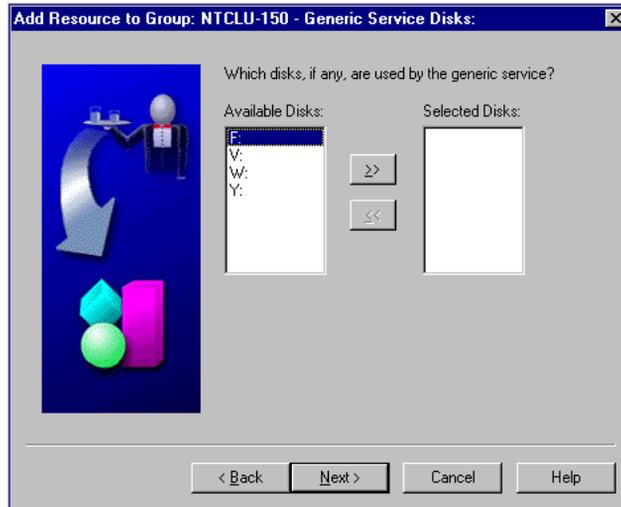
通常、サービス実行可能ファイルは各クラスタ・ノードのプライベート・ディスク上にインストールし、データ・ファイルは共有クラスタ・ディスク上に配置します。実行可能ファイルの配置場所の詳細は、[14.3.2.2 項](#)を参照してください。

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 14-5 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 14-5 「汎用サービスのディスク」ウィザード・ページ

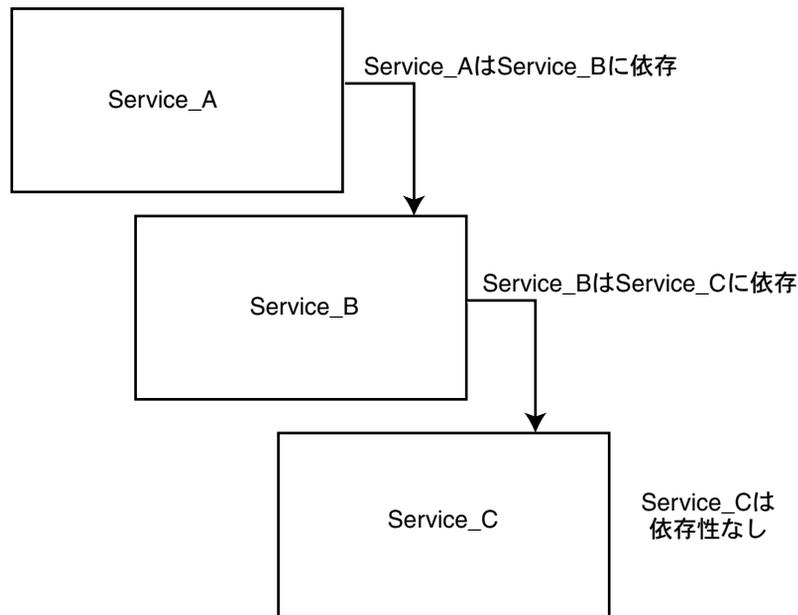


14.3.2.5 汎用サービスの依存性

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。

14.3.2.5.1 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するとします。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 14-6 のように、この依存性の連鎖はツリー構造で図示できます。

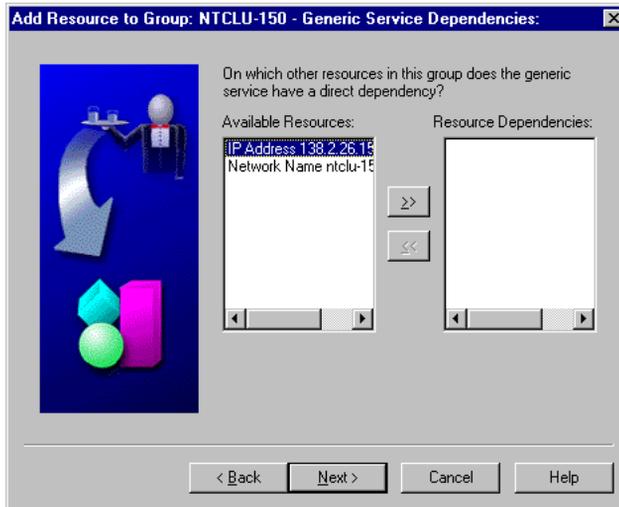
図 14-6 依存性ツリー



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 14-7 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 14-7 「汎用サービスの依存性」ウィザード・ページ



14.3.2.5.2 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が生じる可能性があります。

14.3.2.6 汎用サービスのレジストリ・キー

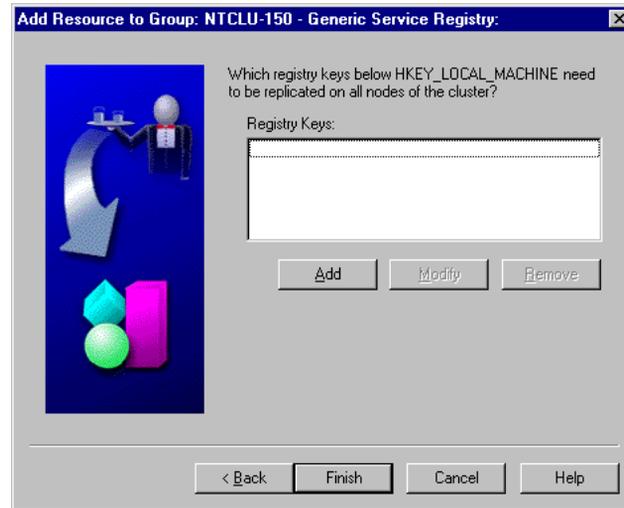
汎用サービスで情報を格納するために Windows レジストリのエントリを使用する場合は、「リソースをグループに追加」ウィザードでこれらのエントリを指定できます。ウィザードでエントリを指定すると、汎用サービスの可能所有者であるクラスタ・ノード間で、サービスの Windows レジストリ・エントリの一貫性が常に保たれます。これは、フェイルオーバー時に、汎用サービスの可能所有者である全クラスタ・ノードで、サービスが正常に稼働するために重要です。

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 14-8 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 14-8 「汎用サービスのレジストリ」ウィザード・ページ



14.4 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を確認されます。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を確認されます。有効でない場合、エラーが返されます。

14.5 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 14-2 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 14-2 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%fsivr%bin%FsSamplesvc.exe に置かれます。
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します。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 表示名： Fail Safe Sample Service サービス名： FsSampleService イメージ名： <Oracle_Home>%fs%fsivr%bin%FsSamplesvc.exe この他のデータを「リソースをグループに追加」ウィザードで入力する必要はありません。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、「fssvclient」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、fssvclient 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。

14.6 汎用サービスに関する問題のトラブルシューティング

「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。

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。

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かぎり、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 以上の 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、TAF 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。

注意： TAF 機能は、Oracle データベース・サーバー Enterprise Edition でのみサポートされています。

Oracle Fail Safe ソフトウェアに含まれている TAFDemoGUI.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実演するデモ・プログラムです。このプログラムには、OCI デモおよび 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) Enterprise Edition 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
ホスト名および IP アドレスの登録	B.1 項
クラスタ内の正しい名前解決の有効性検査	B.2 項
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 項

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かかわらず、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 のいずれか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、全システムに対して `ping` を実行し、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して、ホスト自体からホスト名に対して `ping` を実行した場合も、他の任意のシステムから `ping` を実行した場合も、同一の IP アドレスが返されることを確認します。

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て、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

クラスタ・システム 1、クラスタ・システム 2 およびクライアント・システムの各システム上で、4 つのホスト名すべてに対して `ping` を実行します。たとえば、クラスタ・システム 1 で次を実行します。

```
ping ClusterHost1
ping ClusterHost2
ping ClusterAlias
ping ClientHost
```

ping を実行した各ホスト名に対して、ping テストでは同一の 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、クラスタ・システム 1、クラスタ・システム 2 およびクライアント・システム上での 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

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、クラスタ・システム 1 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

クラスタ・システム 1 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 5 以上に推奨されるソリューション

Windows NT 4.0 Service Pack 5 以上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。

この操作の目的は、IP アドレスがホスト名の既知のアドレスにマップされるような順序を選択することです。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.3.2 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
問題の報告	C.1 項
バージョン情報の検出	C.2 項
Oracle Fail Safe の問題のトレース	C.3 項
トレース・ファイルとアラート・ファイルの保管	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、あらかじめ次の情報をご用意ください。

- 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 およびその他のソフトウェア製品の完全なリリース番号。
- 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（リリース番号を含む）。
- 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可。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 = 機能はかぎられているが、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。

ファイルにエラー出力のログを取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。

「ヘルプ」 → 「Oracle Fail Safe Manager のバージョン情報」

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 のレジストリ・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 Services for MSCS 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 Services for MSCS 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. 複数の Oracle Services for MSCS トレース・フラグを設定する場合は、手順 3 ～ 7 を繰り返します。
9. MSCS クラスタ アドミニストレータを使用して Oracle Services for MSCS を停止します。
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 Services for MSCS を停止します。
11. MSCS クラスタ アドミニストレータを使用して Oracle Services for MSCS 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。次に例を示します。 C:\%fsr_tracelog
FSS_TRACE_OUTPUT	パスおよびファイル名	Oracle Services for MSCS 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。次に例を示します。 C:\%fss_tracelog

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のログを取ります。
	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、情報のログを取ります。
	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、情報のログを取ります。
	DEL_SAMPLE	「サンプル・データベースの削除」操作に関する情報のログを取ります。CR_SAMPLE 操作と似ています。
	GR_VERIFY	「グループの検証」操作に関する情報のログを取ります。
	LOCAL_TRACE	ローカル・トレースを使用可能にして、あるクラスタ・ノードのトレース出力を、そのノードの FSS_TRACE_OUTPUT ファイルに書き込むように指定します。このフラグを指定しない場合、すべてのクラスタ・ノードのトレース出力は Oracle Services for MSCS を実行するノード (クラスタ・グループが常駐するノード) の FSS_TRACE_OUTPUT ファイルに書き込まれます。 トレースする情報のタイプを指定するために、1 つ以上の追加の FSS_TRACE_FLAG 文字列を指定する必要があります。LOCAL_TRACE 文字列のみを指定した場合、トレース出力は作成されません。
	SQLNET	Oracle Services for MSCS が実行した Oracle Net 構成に関連する詳しい内部情報を生成します。Oracle Net 構成の変更が必要な操作が実行されるたびに、情報のログが取られます。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」操作に関する情報のログを取ります。

C.4 トレース・ファイルとアラート・ファイルの保管

トレース・ファイルとアラート・ファイルは、クラスタ・ディスクまたはプライベート・ディスクに保管できます。

- 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・ファイルで使用されるディスク（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移動されていると、完全な履歴情報を得るには各クラスタ・ノードのファイルを合せて表示することが必要な場合もあります。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。

24 × 365

1日24時間、1年365日のビジネス体制。

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。

CGI

「[Common Gateway Interface \(CGI\)](#)」を参照。

Common Gateway Interface (CGI)

サーバー上で実行されるプログラムによって（通常 Web ブラウザを使用して）ユーザーとの対話を可能にする、Web サーバーの一機能。CGI スクリプトによって、動的な Web ページや Web ページの構成要素の作成や、ユーザーの入力の読み込みとそれに応じた出力などの、ユーザーとの対話が可能になる。一般的な使用例としては、ユーザーがオンラインで入力して提出する対話式フォームがある。CGI スクリプトで使用される共通言語は、Perl、JavaScript および Java などである。

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 Server 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。

インスタンス (instance)

システム・グローバル領域 (SGA) と 1 つ以上の Oracle データベース・サーバー・プロセスの組合せ。データベース・サーバーでデータベースが起動すると、Oracle は SGA を割り当て、1 つ以上の Oracle プロセスを開始する。インスタンスのメモリーおよびプロセスは、関連付けられたデータベースのデータを効率良く管理し、データベース・ユーザーに提供する。各インスタンスには一意の Oracle システム識別子 (SID)、インスタンス名、インスタンス番号、ロールバック・セグメントおよびスレッド ID がある。

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 node)

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。

- 指定し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)

指定したリソースのリソース 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ノードのセット。

可用性 (availability)

システムまたはリソースが、必要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 (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。

クォーラム (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、[クォーラム・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。クォーラムは、クラスタを確立する際にも使用される。「[クォーラム・リソース](#)」も参照。

クォーラム・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、クォー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、クォーラム・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。「[クォーラム](#)」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つ以上の独立したコンピューティング・システムのグループ。

クラスタ構成 (cluster configuration)

「[マスター/スレーブ実装](#)」を参照。

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 (group failover)

クラスタ・リソースのグループをあるノードでオフライン化し、そのリソースを別のノードでオンライン化するプロセス。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）のいずれかとなる。

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。

グループ・フェイルバック・ポリシー (group fallback policy)

クラスタ・リソースをフェイルオーバー・ノードから優先所有者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。

計画外グループ・フェイルオーバー (unplanned group 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。「[計画的グループ・フェイルオーバー](#)」も参照。

計画的グループ・フェイルオーバー (planned group 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化し、もう一方のノードでオンライン化する処理。たとえば、『Oracle Fail Safe for Windows インストール・ガイド』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（別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。「[計画外グループ・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。

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・ナッシング構成 (shared-nothing configuration)

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。

冗長周辺機器 (redundant component)

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タ周辺機器。

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)

優先所有者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「[優先所有者ノード](#)」も参照。

セカンダリ・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。

停止時間 (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時間の割合。停止時間は、装置が必要とされる合計時間のうち、アクセス不可である時間の割合として測定される。

データ・ファイル (data file)

表や索引などの論理データベース構造のコンテンツを含むファイル。1つ以上のデータ・ファイルは、表領域と呼ばれる論理記憶域を形成する。あるデータ・ファイルは1つの表領域および1つのデータベースのみに関連付けられる。

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 より前のリリースでは、ネット・サービス名はサービス名エントリと呼ばれていた。

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化し、そのリソースをもう一方のノードでオンライン化するプロセス。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) のいずれかとなる。

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・リソースをあるノードから別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、試行回数の最大値。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化する。「[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)

使用不可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)

優先所有者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・ノードから優先所有者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)

「[グループ・フェイルバック・ポリシー](#)」を参照。

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、あるノードの使用可否を別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。

マスター / スレーブ実装 (master/slave implementation)

1つをマスターとしてその他をスレーブとして指定する、複数の Oracle Reports Server を使用した構成。ユーザーは、マスター Oracle Reports Server にレポート要求を送る。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・サーバーに転送する。いずれかのスレーブが使用できない場合、マスター Oracle Reports Server はそのスレーブには要求を送らない。Oracle Reports のマニュアルでは、これを「クラスタ構成」と呼ぶ。

ミッション・クリティカル・アプリケーション (mission-critical application)

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。

優先所有者ノード (preferred owner 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化およびオフライン化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・インストール手法。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、各ノードが順番にアップグレードされて再起動されるため、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化している間、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

A

- abort モード
 - シングルインスタンス・データベースのオフライン化, 5-5
- Active Server Pages (ASP)
 - Web アプリケーション, 11-3
- Apache
 - Oracle HTTP Server powered by「Oracle HTTP Server」を参照

B

- Bequeath プロトコル・アダプタ
 - Oracle Fail Safe のシングルインスタンス・データベースへの接続, 7-20
- BYPASS 値, 7-20

C

- CGI (Common Gateway Interface)
 - Oracle Forms Load Balancer Server, 8-2
 - Oracle HTTP Server, 11-11
 - rwcgi60.exe 実行可能ファイルの検索, 10-28
 - Web アプリケーション, 11-3
- CLUSTER_MGR 文字列, C-5
- CLUSTER 修飾子
 - FSCMD コマンド, 5-4
- common gateway interface
 - 「CGI」を参照
- Concurrent Manager
 - 「Oracle Applications Concurrent Manager」を参照
- CONNECT_DATA パラメータ, 7-42, 7-43
- DELAY パラメータ, 7-43
- FAILOVER_MODE パラメータ, 7-43

- METHOD サブパラメータ, 7-43
- RETRIES パラメータ, 7-43
- TYPE パラメータ, 7-43
- CR_DBRES 文字列, C-5
- CR_SAMPLE 文字列, C-5
- ctxsrv サーバー
 - 可用性を高めるための構成, 7-33
 - 汎用アプリケーションとしての構成, 7-33
- ctxsrv サーバーのグループへの追加, 7-33

D

- Data Guard
 - 「Oracle Data Guard」を参照, 3-10
- Database Configuration Assistant (DBCA), 7-3
 - Oracle Net 構成の作成, 7-3
 - ネット・サービス名エントリ, 7-3
- DB_NAME パラメータ, 7-16
- DBA_AUTHORIZATION パラメータ, 7-20, 7-29
- DBCA
 - 「Database Configuration Assistant」を参照
- DCOM, 13-2
- DEL_SAMPLE 文字列, C-5
- DELAY パラメータ
 - CONNECT_DATA パラメータ, 7-43
- DISABLEISALIVE パラメータ, 5-3, 7-50
- DISPATCHERS パラメータ
 - 完全なリスナー情報の指定, 7-28
 - 情報の指定, 7-28
- DNS サーバー
 - IP アドレスおよびホスト名の登録, B-2
- DOMAIN 修飾子
 - FSCMD コマンド, 5-5
- DUMPCLUSTER パラメータ, 5-3

Dynamic Link Library (DLL)
FsResOdb.dll, 2-8
FsResOdbEx.DLL, 2-8
IP アドレス, 2-8
Is Alive ポーリング間隔でのシングルインスタンス・データベース・アクセス, 7-48
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

E

ENABLEISALIVE パラメータ, 5-3, 7-50

F

FAILOVER_MODE パラメータ, 7-44
CONNECT_DATA 文字列, 7-43
METHOD キーワード, 7-43
TYPE キーワード, 7-43
FORMS60_HTTP_NEGOTIATE_DOWN 環境変数, 9-11
FORMS60_PATH レジストリ変数, 9-19
Oracle Forms Server, 9-13
formsweb.cfg ファイル, 8-11, 8-13
MetricServerPort パラメータ, 8-11
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, 9-19
Forms の .fmx ファイル, 9-19
FSCMD コマンド, 1-5
DISABLEISALIVE パラメータ, 5-3, 7-50
DUMPCLUSTER パラメータ, 5-3
ENABLEISALIVE パラメータ, 5-3, 7-50
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, 3-6
構文, 5-2
使用上の注意, 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 スクリプト, 7-45
FSR_TRACE_OUTPUT 値, C-4
FsResOdb.dll ファイル
機能, 2-8
FsResOdbEx.DLL ファイル
機能, 2-8
FSS_TRACE_FLAGS 値, C-5
FSS_TRACE_OUTPUT 値, C-4

G

GR_VERIFY 文字列, C-5

H

Hosts ファイル
IP アドレスおよびホスト名の登録, B-2
HOST パラメータ
tnsnames.ora ファイル, 7-42
HTML ファイル, 9-17
JInitiator タグ, 9-18
module= 構文, 9-19
Oracle Forms アプリケーション, 9-17
HTTPS 接続モード, 9-11
HTTP 接続モード, 9-11

I

immediate モード
シングルインスタンス・データベースのオフライン化, 5-5
INTERNAL アカウント, 7-55
パスワードの変更, 7-30
IP アドレス
クラスタへの割当て, 2-12
正しい名前解決の有効性検査, 11-16, B-2
登録, B-2
必要数の決定, 2-12
複数のアドレスのトラブルシューティング, B-3
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, 7-53
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

再度の使用可能設定, 5-3
使用可能にする, 7-50
使用不可にする, 5-3, 7-50
バックアップ操作中の影響, 7-37

J

Java

アプレット・キャッシュをオフにする, 9-18
アプレットの再ロード, 9-18
アプレットの実行時の問題, 9-18
アプレットのダウンロード時の問題, 9-17
仮想マシンを使用した一貫性の保持, 11-3

JavaScript

Web アプリケーション, 11-3

JInitiator

Oracle Forms に関するソフトウェア互換性, 9-18
使用の推奨, 11-3

L

listener.ora ファイル, 4-5

Database Configuration Assistant による更新, 7-3
EXTPROC 接頭辞, 7-24
アーカイブ, 7-54
仮想サーバーの構成で発生する問題, 7-52
構成データの確認, 7-48
サンプル定義, 7-52
複数の構成, 7-6
変更, 7-52
例, 7-53
ロールバック・ファイル, 7-53

Load Balancer Server

「Oracle Forms Load Balancer Server」を参照

Local Area Network, 1-8

LOCAL_LISTENER パラメータ

グループの情報の更新, 7-28
シングルインスタンス・データベース・パラメータ・ファイルへの書き込み, 7-28
シングルインスタンス・データベース・パラメータ・ファイルへの追加, 7-7
シングルインスタンス・データベースをグループから削除した後の削除, 7-28
リスナーの指定, 7-28

LocalSystem アカウント

汎用サービス, 14-10

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, 7-54

M

METHOD サブパラメータ

CONNECT_DATA 文字列, 7-43

MetricServerPort パラメータ

formsweb.cfg ファイル, 8-11

Microsoft Distributed Transaction Coordinator

「MS DTC」を参照

Microsoft Transaction Server

説明, 13-2

「Oracle MTS Service」も参照

Microsoft Windows NT Enterprise Edition 4.0

登録に関する注意事項, B-1

MOVEGROUP パラメータ, 5-3

MS DTC

可用性を高めるための構成, 13-5

構成の検証, 13-5

MSCS

Is Alive ポーリング, 2-8

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, 1-2

保留タイムアウト値, 7-49, 12-17

リソース・タイプ, 2-8

MSCS クラスタ アドミニストレータ

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

MTS_DISPATCHERS パラメータ

「DISPATCHERS パラメータ」を参照

N

NODE 修飾子

FSCMD コマンド, 5-4

normal モード

シングルインスタンス・データベースのオフライン化, 5-5

O

OCI

クライアント

自動再接続を使用可能にする, 7-42

フェイルオーバー・プロセス, 7-40

ODBC クライアント

自動再接続を使用可能にする, 7-42

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_DB

Windows オペレーティング・システム・グループ,
7-17

ORA_sid_DB

Windows オペレーティング・システム・グループ,
7-17

Oracle Applications Concurrent Manager

オフライン化

保留タイムアウト, 12-18

仮想アドレス, 12-10

可能所有者ノード, 12-8

構成 ID, 12-5, 12-11

削除された Windows サービスの再作成, 12-16

スタンドアロンの検出, 12-4

データベース, 12-14

名前, 12-11

不安定な状態の解決, 12-17

「リソース」も参照

Oracle Applications Concurrent Manager のグループへ

の追加, 12-4

アカウント, 12-13

仮想アドレス, 12-10

可能所有者ノード, 12-8

構成 ID, 12-11

前提条件, 12-5

データベース, 12-14

手順, 12-5

マネージャ名, 12-11

Oracle Data Guard

Oracle Fail Safe との使用, 3-10

Oracle Database Upgrade Assistant

シングルインスタンス・データベースのアップグ
レード, 7-31

Oracle Enterprise Manager

アクティブ / アクティブ構成での使用, 3-6

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

クラスタの検出, 7-34

グループの検出, 7-34

統合, 7-34

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

Oracle Fail Safe Manager, 1-2

Oracle Fail Safe Server の前のリリースとの互換性,
4-9

ウィザード, 1-5

概要, 1-5

クラスタ別名, 2-12

検証ツール, 1-5

ツリー・ビュー, 1-5

Oracle Fail Safe Server

Oracle Fail Safe Manager のリリースとの互換性,
4-9

「Oracle Services for MSCS」も参照

Oracle Forms

CGI-bin 実行可能ファイル, 8-3

Forms ランタイム・エンジンの起動, 9-19

JInitiator の使用, 9-18

Web サーバーとの構成, 11-3

アプリケーションの JInitiator タグ, 9-18

Oracle Forms Load Balancer Client

localhost, 8-13

Oracle Forms Load Balancer Server との関連, 8-2

仮想アドレス, 8-13

起動パラメータ, 8-13

データ・ポート, 8-13

データ・ホストのデフォルト値, 8-13

Oracle Forms Load Balancer Server

CGI 実行可能ファイル, 8-2

formsweb.cfg ファイル, 8-11

Oracle Forms Load Balancer Client との関連, 8-2

Oracle Forms Server との関連, 8-2

概要, 8-2

仮想アドレス, 8-10

可能所有者ノード, 8-8

起動パラメータ, 8-13

起動パラメータの変更, 8-13, 8-14

- クライアントの最大数, 8-11
- クラスタの3層構造, 8-3
- グループへの追加, 8-5
- グループへの追加手順, 8-6
- 検出, 8-5
- 構成手順, 8-6
- スタンドアロンの検出, 8-5
- セキュリティ要件, 8-12
- データ・ポート, 8-10
- トラブルシューティング, 8-14
- トレース・レベル, 8-11
- 名前, 8-10
- リクエスト・ポート, 8-11
- 「リソース」も参照
- Oracle Forms Load Balancer Server のグループへの追加, 8-6
 - 仮想アドレス, 8-10, 9-9
 - 可能所有者ノード, 8-8
 - サーバー名, 8-10
 - 前提条件, 8-6
 - データ・ポート, 8-10
 - 手順, 8-6
 - リクエスト・ポート, 8-11
- Oracle Forms Server, 9-16
 - FORMS60_PATH レジストリ変数, 9-13
 - Oracle Fail Safe Manager ツリー・ビューでの外観, 9-15
 - Oracle Forms Load Balancer Server との関連, 8-2
 - アクセスされる Oracle データベースの指定, 9-14
 - アプリケーション・ファイルの場所, 9-13
 - 仮想アドレス, 9-4, 9-9
 - 可能所有者ノード, 9-7
 - 可用性の向上に必要な情報, 9-6
 - 可用性を高めるための構成, 9-4
 - クライアント接続, 9-16
 - クラスタ内に設定されるアプリケーション, 9-4
 - グループへの追加手順, 9-5
 - グループへの追加に対する前提条件, 9-4
 - 検出, 9-3
 - 構成手順, 9-5
 - セキュリティ要件, 9-15
 - データ・ファイル, 9-13
 - データベースへのアクセス, 9-14
 - トラブルシューティング, 9-16
 - アプリケーションが見つからない, 9-19
 - サーバーにアクセスできない, 9-17
 - 名前, 9-10
 - パス, 9-13
 - プール・パラメータ, 9-12
 - ポート, 9-10
 - 「リソース」も参照
- Oracle Forms Server のグループへの追加, 9-5
 - FORMS60_PATH, 9-13
 - tnsnames.ora ファイル, 9-14
 - アプリケーション・ファイルの場所, 9-13
 - 可能所有者ノード, 9-7
 - サーバー名, 9-10
 - 接続モード・パラメータ, 9-11
 - 前提条件, 9-4
 - データベース, 9-14
 - 手順, 9-5
 - ポート番号, 9-10
- Oracle HTTP Server
 - HTTP アクセス, 11-12
 - Oracle ホーム, 11-4
 - SSL アクセス, 11-12
 - Web サイトに接続できない, 11-16
 - 仮想アドレス, 11-12
 - 仮想ディレクトリ, 11-10
 - 「可能所有者ノード」リスト, 11-7
 - 可用性を高めるための構成, 11-4
 - クライアント接続, 11-13
 - クラスタ・ディスク, 11-9
 - グループからの削除, 11-14
 - グループへの追加, 11-6
 - グループへの追加に対する前提条件, 11-4
 - 構成ファイル, 11-9, 11-14
 - スタンドアロンの検出, 11-4
 - セキュリティ要件, 11-15
 - ディレクトリ索引ファイル, 11-10
 - ドキュメント・ルート・ディレクトリ, 11-9
 - トラブルシューティング, 11-15
 - 名前, 11-8
 - ユーザーがアクセスできない, 11-16
 - 「リソース」も参照
- Oracle HTTP Server のグループへの追加, 11-4
 - 仮想アドレス, 11-12
 - 仮想ディレクトリ, 11-10
 - 可能所有者ノード, 11-7
 - サーバー名, 11-8
 - サーバー・ルート, 11-9
 - 前提条件, 11-4
 - ディレクトリ索引ファイル, 11-10
 - 手順, 11-5

- Oracle Intelligent Agent
 - 仮想アドレス, 7-36
 - グループへの追加, 7-35
 - 通信の管理, 2-6
 - デフォルト, 7-36
- Oracle Intelligent Agent のグループへの追加, 7-35
 - 前提条件, 7-35
- Oracle MTS Service
 - 仮想アドレス, 13-11
 - 可能所有者ノード, 13-9
 - 検出, 13-4
 - サービス名, 13-12
 - 削除した場合の再作成, 13-16
 - スタンドアロンの検出, 13-4
 - データベース, 13-13
 - 名前, 13-12
- Oracle MTS Service のグループへの追加, 13-5
 - アカウント認証, 13-14
 - 仮想アドレス, 13-11
 - 可能所有者ノード, 13-9
 - サービス名, 13-12
 - 前提条件, 13-5
 - データベース, 13-13
 - 手順, 13-6
- Oracle Net Assistant
 - ネットワーク・ファイルの再作成のための使用, 7-52
- Oracle Net ネットワーク
 - tnsnames.ora ファイルの変更, 7-23
 - アーカイブされた構成ファイル, 7-54
 - グループ内のリソース, 2-6
 - 構成情報のトレース, C-5
 - 構成のトラブルシューティング, 7-48
 - シングルインスタンス・データベースの構成, 7-22
 - ロールバック・ファイル, 7-54
- Oracle Net リスナー
 - 依存性, 7-20
 - 構成, 7-20
 - 作成, 7-20
 - 動作, 7-4
- Oracle Portal
 - Oracle Reports Server でのユーザー・アクセスの制御, 10-25
- Oracle Reports Server
 - Oracle Portal によるユーザー・アクセスの制御, 10-25
 - persistfile パラメータ, 10-14
 - .rdf ファイル, 10-14, 10-29
 - tnsnames.ora ファイル, 10-19
 - tnsnames.ora ファイルの例, 10-19
 - Web サーバー仮想ディレクトリのレジストリ変数, 10-30
 - Web サーバーとの構成, 11-3
 - Web ベースの構成, 10-2
 - アーキテクチャ, 10-2
 - アクセスされるデータベース, 10-16
 - 仮想アドレス, 10-12
 - 仮想アドレスの判断, 10-19
 - 可能所有者ノード, 10-10
 - 可用性を高めるための構成, 10-7
 - 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, 10-23
 - キャッシュ・ディレクトリ, 10-14
 - 仮想ディレクトリによる指定, 10-30
 - レジストリ変数, 10-30
 - キュー・アクティビティの表示, 10-27
 - クライアント・アクセスのための tnsnames.ora ファイルの更新, 10-19
 - クライアント / サーバー実装, 10-2, 10-4
 - クライアント接続, 10-18
 - クラスタ構成, 10-6
 - 検出, 10-6
 - 識別情報, 10-13
 - 実行可能ファイルの場所, 10-7
 - 実行に使用されるアカウント, 10-17
 - 出力ファイルの場所, 10-7
 - ジョブ・ディレクトリ, 10-14
 - スレーブ・サーバーに対するマスター・サーバーの指定, 10-23
 - 層, 10-2
 - ソース・ディレクトリ, 10-14
 - クラスタ・ディスク, 10-29
 - 通常ファイル, 10-26
 - データ・ファイルの場所, 10-7
 - トラブルシューティング, 10-28
 - 名前, 10-13
 - 複数, 3-8
 - プリンタ, 10-17
 - ポート, 10-13
 - マスター / スレーブ実装, 10-2, 10-6
 - マスター / スレーブ実装の構成, 10-20
 - マスターに対するスレーブの指定, 10-21
 - マスターに対する高い可用性の提供, 10-1
 - ランタイム・エンジン, 10-29

- レポートの実行, 10-18
- ロード・バランス, 10-6
- 「リソース」も参照
- Oracle Reports Server のグループへの追加, 10-8
- persistfile パラメータ, 10-15
- tnsnames.ora ファイル, 10-16
- 仮想アドレス, 10-12
- 可能所有者ノード, 10-10
- キャッシュ・ディレクトリ, 10-14
- サーバー名, 10-13
- ジョブ・ディレクトリ, 10-14
- 前提条件, 10-7
- ソース・ディレクトリ, 10-14
- データベース, 10-16
- 手順, 10-8
- プリンタ・アクセス用のアカウント, 10-17
- ポート, 10-13
- Oracle Services for Microsoft Transaction Server
 - 「Oracle MTS Service」を参照
- Oracle Services for MSCS
 - Windows セキュリティ情報の更新, 4-7
 - アカウント, 4-7
 - セキュリティ設定ツール, 4-7
 - 説明, 1-2
- 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 シングルインスタンス・データベース
 - Oracle8i へのアップグレード, 7-4
 - Oracle9i へのアップグレード, 7-4
- Oracle データベース・リソース・タイプ, 2-8
- Oracle ホーム
 - Oracle Forms Load Balancer Server の構成, 8-6
 - Oracle Forms の構成, 9-4
 - Oracle HTTP Server のパス, 11-4
 - グループ, 4-9
 - 名前, 6-3
 - 複数, 4-9

P

- Perl
 - Web アプリケーション, 11-3
- persistfile パラメータ, 10-15
 - Oracle Reports Server, 10-14
- PFILE
 - 「初期化パラメータ・ファイル」を参照
- ping
 - クラスタ・ノード間のネットワーク通信の検証, 11-16, B-2
- 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, 1-8
- 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, 7-50
 - 設定, 7-49
 - パスワード・ファイルを使用しない場合, 7-20
 - パスワード・ファイルを使用する場合, 7-20
- REPORTS60_CGIMAP レジストリ変数, 10-18
- REPORTS60_PHYSICAL_MAP レジストリ変数, 10-30
- REPORTS60_VIRTUAL_MAP レジストリ変数, 10-30
- RETRIES パラメータ
 - CONNECT_DATA パラメータ, 7-43
- rwcgi60.exe ファイル
 - 検索時の問題, 10-28
 - ダウンロード, 10-28

S

- SCSI 共有記憶装置インターコネクタ, 1-8
- 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, 7-44
- SERVER=DEDICATED パラメータ, 7-25
- SERVICE_NAME パラメータ
 - Database Configuration Assistant による追加, 7-3

SID

CONNECT_DATA 文字列内のキーワード, 7-43
Database Configuration Assistant により追加される
名前, 7-3

tnsnames.ora ファイルのエントリ, 4-5
シングルインスタンス・データベースのための入
力, 7-52
複数のリスナー, 7-6
リスナー間で重複した名前, 7-53

SID_DESC パラメータ, 7-3

Oracle8i へのアップグレード, 7-4
Oracle9i へのアップグレード, 7-4
新規シングルインスタンス・データベース用の追
加, 7-3
シングルインスタンス・データベースをグループに
追加した後の更新, 7-24

Socket 接続モード, 9-11

SPFILE

「初期化パラメータ・ファイル」を参照

SQL*Plus

管理作業の実行, 7-37

SQLNET.AUTHENTICATION_SERVICES パラメータ シングルインスタンス・データベースをグループに 追加した後の更新, 7-24

sqlnet.ora ファイル

シングルインスタンス・データベースをグループに
追加した際に行われる更新, 7-24
デフォルト・ドメイン値の更新, 7-3

SQLNET 文字列, C-5

SYSDBA アカウント

パスワードの変更, 7-30

SYSDBA 権限, 7-29

SYSOPER 権限, 7-29

SYS アカウント, 7-29, 7-55

パスワードの変更, 7-30

T

TAFDemoGUI.exe 実行可能プログラム

デモ, A-1

TNS_ADMIN Windows 環境変数, 7-22

TNS_ADMIN Windows レジストリ・パラメータ, 7-22

tnsnames.ora ファイル, 4-5

CONNECT_DATA 文字列, 7-42, 7-43
Database Configuration Assistant による更新, 7-3

EXTPROC_CONNECTION_DATA ネット・サービ
ス名エントリ, 7-25

HOST パラメータ, 7-42

Oracle Forms Server, 9-14

Oracle Net 構成データの確認, 7-48

Oracle Reports Server, 10-16, 10-19

Oracle Reports Server のマスター / スレーブ実装, 10-21

Oracle Reports Server の例, 10-19

SERVER=DEDICATED パラメータ, 7-25

アーカイブ, 7-54

仮想サーバーの構成で発生する問題, 7-52

可用性の高いシングルインスタンス・データ ベースのための変更, 7-21

更新, 7-21

シングルインスタンス Oracle データベース・サー バー, 7-10

シングルインスタンス・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, 7-3

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

ロールバック・ファイル, 7-53

TNS リスナー

ネット・サービス名の更新, 7-6

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, 7-43

U

UNIX システム

階層化構成, 3-10

URL

Oracle HTTP Server Web サイト用の指定, 11-13

Oracle Reports Server の通信のトラブルシューティ
ング, 10-29

USE_SHARED_SOCKETS パラメータ, 7-21

USER 修飾子

FSCMD コマンド, 5-6

V

VERIFY_DB 文字列, C-5

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, 11-3

W

Web アプリケーション

Oracle Fail Safe によるサポート, 11-3
標準, 11-3

Web サイト

Oracle HTTP Server に接続できない, 11-16

Web ブラウザ

Java アプレットをダウンロードできない, 9-17
JInitiator を使用した一貫性の保持, 11-3
Oracle Reports の .rdf ファイルへのアクセス, 10-29
Oracle Reports の再ロード, 10-18
フェイルオーバー後の Oracle Forms の再ロード,
9-4

Web ページ

作業負荷のパーティション化構成, 3-7

Wide Area Network, 1-8

Windows 2000 Advanced Server, 1-8

デフォルトの Oracle Intelligent Agent の構成, B-4

Windows NT Enterprise Edition, 1-8

Windows NT ユーザー・アカウント

セキュリティ設定, 4-6

Windows オペレーティング・システム・グループ

ORA_DBA, 7-17

ORA_sid_DBA, 7-17

Windows 環境変数

TNS_ADMIN, 7-22

Windows クラスタ

「クラスタ」を参照

Windows レジストリ・エディタ, B-2

Windows レジストリ・キー

「レジストリ・キー」を参照

Windows レジストリ・パラメータ

TNS_ADMIN, 7-22

WINS

IP アドレスおよびホスト名の登録, B-2

あ

アーカイブされたファイル, 7-54

アカウント

INTERNAL

パスワードの変更, 7-30

Oracle Applications Concurrent Manager を実行する, 12-13

Oracle MTS Service に必要な権限, 13-15

Oracle MTS Service に必要なシングルインスタンス・データベース・ロール, 13-15

Oracle MTS Service の実行に使用, 13-14

Oracle Reports Server の実行に使用, 10-17

Oracle Services for MSCS アカウントのドメイン名の変更, 4-7

Oracle Services for MSCS の実行に使用, 4-7

SYS

パスワードの変更, 7-30

SYSDBA

パスワードの変更, 7-30

権限および許可, 4-5

シングルインスタンス Oracle データベースの管理,
7-29

データベース・パスワード・ファイルへの追加,
7-29

汎用サービスの実行に使用される, 14-9, 14-10

アクティブ / アクティブ構成, 3-4

アクティブ / パッシブ構成, 3-2

アプリケーション

フェイルオーバー, 1-7, 2-31

概念, 7-40

フロー・チャート, 7-40

フェイルオーバー後の再接続, 7-42, 7-43

フェイルオーバー後の自動再接続, 1-7

ミッション・クリティカル, 3-6

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・ログ

起動問題のトラブルシューティング, 6-12

い

依存性

定義, 14-13

汎用サービス, 14-12

リスナー, 7-20

リソース, 2-7

一時表

フェイルオーバー, 7-9

「一般」プロパティ・ページ

リソース, 2-23

イベント

情報を格納, 7-36

イメージ名

汎用サービス, 14-8

インスタンス名

可用性の高いシングルインスタンス・データベース, 7-15

「SID」を参照

インストール

Oracle Fail Safe, 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

レスポンス・ファイルを使用, 1-4

インターコネクト

共有記憶装置, 2-2, 7-47

プライベート・ノード間接続, 2-2

う

ウィザード, 1-5

入力項目, 4-3

え

エラー

FS-10066, 7-52

FS-10070, 7-52

FS-10101, 7-54

FS-10270, 7-51

FS-10618, 13-17

FS-10778, 13-17

ORA-01031, 7-55

Oracle Net のリスナー・ログへの書き込み, 7-48

トレース・ファイル内の情報の追跡, C-3

不十分な権限に対して返される, 7-55

ユーザー認証, 7-54

レポート, C-2

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

オペレーティング・システム認証, 7-17

オラクル社カスタマ・サポート・センター

連絡, C-2

か

階層化構成, 3-8

UNIX の使用, 3-10

外部プロシージャ

構成, 7-24

カスタマ・サポート・センター

連絡, C-2

カスタム・リソース・タイプ, 2-8

仮想アドレス, 1-3

Oracle Applications Concurrent Manager, 12-10

Oracle Forms Load Balancer Client, 8-13

Oracle Forms Load Balancer Server, 8-10

Oracle Forms Server, 9-4, 9-9

Oracle HTTP Server, 11-12

Oracle Intelligent Agent, 7-36

Oracle Intelligent Agent での使用, 7-36

Oracle MTS Service, 13-11

Oracle Reports Server, 10-12

Oracle Reports Server のマスター / スレーブ実装,
10-21

クラスタ別名, 2-12

グループ, 7-11

グループで複数の仮想アドレスを使用する理由,
4-10

グループへの追加, 2-10

構成の問題のトラブルシューティング, 7-52

識別, 2-10

シングルインスタンス Oracle データベース, 7-14

定義, 2-10

登録, B-2

内部クラスタ通信, 7-23

汎用サービス, 14-14

複数, 2-11, 4-10

仮想サーバー, 1-3

tnsnames.ora ファイルでの指定, 7-42

アドレスの登録, B-2

検出, 7-34

構成, 2-11

構成時の tnsnames.ora ファイルに対する更新, 7-21

構成の問題, 7-52

定義, 2-11

ネットワーク構成, 7-21

複数, 4-10

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, 10-28
 - Oracle HTTP Server, 11-10
 - キャッシュ・ディレクトリ, 10-30
 - 指定, 9-17
- 仮想ディレクトリ設定
 - Java アプレットのダウンロード, 9-17
- 「可能所有者ノード」リスト
 - Oracle Applications Concurrent Manager, 12-8
 - Oracle Forms Load Balancer Server, 8-8
 - Oracle Forms Server, 9-7
 - Oracle HTTP Server, 11-7
 - Oracle MTS Service, 13-9
 - Oracle Reports Server, 10-10
 - グループ, 2-23
 - 「グループの検証」コマンド, 2-22, 2-23, 4-11
 - 指定, 2-23
 - シングルインスタンス Oracle データベース, 7-12
 - 調整, 2-23
 - ノードの除外, 2-22
 - 汎用サービス用, 14-6
 - フェイルオーバー・ノードの決定, 2-27
 - フェイルオーバーへの影響, 2-27
 - 変更, 2-23
 - リソース, 2-22, 2-23
 - リソースへの影響, 2-22
- 「可能所有者ノード」リストの指定, 2-23
- 可用性
 - RAID テクノロジー, 2-3
- 環境変数
 - FORMS60_HTTP_NEGOTIATE_DOWN, 9-11
- 管理
 - Oracle Enterprise Manager との統合, 7-34
 - クライアント・システム上の tnsnames.ora ファイル, 7-10, 7-23
 - クラスタ・セキュリティ, 4-5
 - クラスタ別名の使用, 2-12
- 管理者権限
 - FSCMD コマンドによる指定, 5-6
 - クラスタへのログイン, 4-8
 - セキュリティの設定, 4-6
- 管理操作
 - シングルインスタンス・データベース, 7-37

き

- 起動タイプ
 - 汎用サービス, 14-10
- 起動パラメータ
 - Oracle Forms Load Balancer Client, 8-13
 - Oracle Forms Load Balancer Server, 8-13
 - Oracle Forms Load Balancer Server の変更, 8-14
 - 汎用サービス, 14-9
- キャッシュ
 - Java アプレット, 9-18
- キャッシュ・ディレクトリ
 - Oracle Reports Server, 10-14
- 共有記憶装置インターコネクト, 1-3, 2-2
 - SCSI, 1-8
 - ファイバ・チャネル, 1-8
- 共有サーバー
 - Oracle8i データベースのための構成, 7-28
 - Oracle8 データベースのための構成, 7-25
 - Oracle9i データベースのための構成, 7-28
 - SERVER=DEDICATED パラメータ, 7-25
- 共有ソケット
 - エラー・メッセージ, 7-21
 - サード・パーティ製のプロキシ・サーバー, 7-21
 - 使用, 7-21
 - データベースのグループへの追加, 7-21
- 許可
 - 不十分なため返されるエラー ORA-01031, 7-55

く

- クォーラム・ディスク
 - クラスタ・リカバリ, 2-5
 - 情報のダンプ, 6-10
- クォーラム・リソース
 - メタデータ, 2-5
- クライアント
 - アドレスの登録, B-2
 - クラスタの接続, 2-31
 - フェイルオーバー後の Oracle Forms Server への再接続, 9-4
 - フェイルオーバー後の Oracle Reports Server への再接続, 10-18
 - フェイルオーバー後の Web サーバーへの再接続, 11-1
 - フェイルオーバー後の Web サイトの再ロード, 11-1

- フェイルオーバー後のシングルインスタンス・データベースへの再接続, 7-38
- 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, 4-10
- 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, 10-1
- クライアント接続
 - Oracle Forms Server, 9-16
 - Oracle HTTP Server, 11-13
 - Oracle Reports Server, 10-18
 - シングルインスタンス Oracle データベース・サーバー, 7-21
- クラスタ, 1-2
 - Oracle Enterprise Manager での検出, 7-34
 - 階層化構成, 9-2
 - 概要, 2-2
 - 各種の構成, 2-3
 - クラスタ別名, 2-12
 - 構成, 2-2
 - 構成の検証, 5-3, 6-3
 - 新規ノードの追加, 4-11
 - 接続, 2-12
 - 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, 1-8
 - 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, 1-3, 2-2
 - パスワード・ファイルの同期化, 7-29
 - メタデータ, 2-5
 - メンバー, 2-2
- クラスタ・グループ, 2-12
 - クラスタ別名ネットワーク名, 2-12
 - リソース, 2-12
- クラスタ・ディスク
 - Oracle Forms Server, 9-4
 - Oracle HTTP Server, 11-5, 11-9
 - Oracle Reports Server, 10-7
 - Oracle シングルインスタンス・データベース, 7-9
 - RAID ハードウェア, 1-8
 - 冗長性, 2-3
 - ジョブおよびイベントの情報を格納, 7-36
 - 汎用サービス, 14-11
 - 「ディスク」も参照
- クラスタ内通信, 1-8
- クラスタ・ノード, 2-2
 - 既存クラスタへの追加, 4-11
 - 共有サーバー構成, 7-25
 - 定義, 1-2
- 「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
- クラスタ別名
 - Oracle Fail Safe Manager, 2-12
 - アドレスの登録, B-2
 - 使用, 2-12
 - 定義, 2-12
- クラスタへの IP アドレス割当て, 2-12
- クラスタへの接続
 - ドメイン・ユーザー・アカウント, 4-8
- クラスタ・リカバリ
 - クォーラム・ディスク, 2-5
- クラスタ・リソース
 - 定義, 2-5
- 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, 7-38
- クラスタワイド操作
 - 「グループの検証」操作, 6-6
 - 「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, 9-18
- グループ
 - DBA パスワードの不一致, 7-50
 - Oracle Applications Concurrent Manager の追加, 12-6
 - Oracle Enterprise Manager での検出, 7-34
 - Oracle Forms Load Balancer Server の追加, 8-6
 - Oracle Forms Server の追加, 9-5
 - Oracle HTTP Server の追加, 11-5
 - Oracle Intelligent Agent の追加, 7-35
 - Oracle MTS Service の追加, 13-6
 - Oracle Reports Server の追加, 10-8
 - 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, 7-11
 - 仮想アドレスの追加, 2-10

- クラスタ内のすべてのグループの検証, 5-3
- 作成, 2-6
- 情報のダンプ, 6-10
- 所有権, 2-6
- シングルインスタンス・データベースの追加, 7-10
- シングルインスタンス・データベースを追加するための権限, 7-55
- 帯域幅追加, 2-11
- 定義, 2-6
- ネットワーク名の修正, 7-52
- 「ノード」プロパティ・ページ, 2-23
- 汎用サービスの追加, 14-4
- フェイルオーバー, 2-6
- フェイルオーバー期間, 2-25
- フェイルオーバーしきい値, 2-24
- フェイルオーバー・ポリシー, 2-24
- 複数仮想アドレス, 4-10
- 「優先所有者ノード」リスト, 2-27
- 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, 2-30
- グループの計画的フェイルオーバー, 2-17
- 「グループの検証」操作
 - FSCMD コマンド, 5-3
 - Oracle Forms Server の検証, 9-17
 - Oracle HTTP Server の検証, 11-16
 - 概要, 6-5
 - 「可能所有者ノード」リスト, 2-22, 2-23, 4-11
 - クラスタ・レジストリ内の情報の更新, 7-38
 - 情報のロギング, C-5
 - 説明, 6-2
 - 用途, 6-5
 - リソース依存性の更新, 7-46

け

- 計画外フェイルオーバー, 2-14
- 計画的フェイルオーバー
 - 静的ロード・バランス, 2-17
- 計画的メンテナンス, 2-17
- 権限
 - 各クラスタ・ノードでの付与, 7-29
 - 汎用サービス, 14-15
 - 必須, 4-5
- 検出

- Oracle Enterprise Manager, 7-34
- サンプルのスタンドアロン・シングルインスタンス・データベース, 7-2
- スタンドアロン Oracle Applications Concurrent Manager, 12-4
- スタンドアロン Oracle Forms Load Balancer Server, 8-5
- スタンドアロン Oracle Forms Server, 9-3
- スタンドアロン Oracle HTTP Server, 11-4
- スタンドアロン Oracle MTS Service, 13-4
- スタンドアロン Oracle Reports Server, 10-6
- スタンドアロン・シングルインスタンス・データベース, 7-2
- スタンドアロン汎用サービス, 14-3
- リソース, 4-8
- 検証ツール, 1-5

こ

- 構成, 3-2 ~ 3-10
 - アクティブ / アクティブ, 3-4
 - アクティブ / パッシブ, 3-2
 - ウィザード入力項目の使用, 4-3
 - 階層化, 3-8
 - カスタマイズ, 3-2
 - グループの問題のトラブルシューティング, 12-17, 13-17
 - 検証, 6-5, 6-7
 - 作業負荷のパーティション化, 3-6
 - 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, 1-8
 - 複数仮想アドレス, 4-10
- 構成 ID
 - Oracle Applications Concurrent Manager, 12-5, 12-11
- 構成のカスタマイズ, 3-2
- 構成ファイル
 - Oracle HTTP Server, 11-9

さ

- サード・パーティ製のプロキシ・サーバー
 - 共有ソケットの使用, 7-21
- サーバー・ノード
 - FSS_TRACE_FLAGS 値および文字列, C-5

- アドレスの登録, B-2
- クライアント接続の損失, 7-55
- クラスタ・ポーリングの障害, 7-50
- 再起動の試行, 2-31
- トレース・フラグ, C-5
- 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, 14-15
- サービス名
 - Oracle MTS Service, 13-12
 - 汎用サービス, 14-8
 - 「ネット・サービス名」も参照
- 再接続
 - アプリケーション, 7-43
 - クライアント, 7-42
- 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, 12-17
- 作業負荷のパーティション化構成, 3-7
- 削除
 - Oracle HTTP Server をグループから, 11-14
- サンプル・データベース
 - 必要な権限, 7-55
 - 問題のトラブルシューティング, 7-51
- サンプル・データベースの削除
 - エラー, 7-51
 - 情報の獲得, C-5
- サンプル・データベースの作成
 - 情報のトレース, C-5

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, 10-17
- 実行
 - スタンドアロン Oracle HTTP Server, 11-14
- 周辺機器
 - 冗長, 11-3
- 出力ファイル
 - リスナー, 7-54
- 障害
 - 階層化構成での除去, 8-3, 9-2, 13-3
 - 計画外, 2-14

- 防止のためのリソースの検証, 6-5, 6-7
- 障害耐久力, 3-10
- 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, 7-28
 - PFILE, 7-16
 - REMOTE_LOGIN_PASSWORDFILE 初期化パラメータ, 7-20, 7-29, 7-50
 - SPFILE のエクスポート, 7-16
 - SPFILE を指定する際の制限事項, 7-16
 - 可用性の高いシングルインスタンス・データベース, 7-16
 - 配置, 7-16
 - 場所, 7-16
- ジョブ
 - アクセス問題のトラブルシューティング, 7-55
 - 情報を格納, 7-36
 - スケジューリングした Oracle Reports, 10-14
- ジョブ・ディレクトリ
 - Oracle Reports Server, 10-14
- 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
- シングルインスタンス Oracle データベースのグループへの追加, 7-9
 - Oracle Net リスナー, 7-21
 - tnsnames.ora ファイル, 7-10
 - インスタンス名, 7-15
 - 仮想アドレス, 7-14
 - 可能所有者ノード, 7-12
 - サービス名, 7-15
 - 前提条件, 7-9
 - データベース名, 7-16
 - 手順, 7-10
 - 認証情報, 7-17
 - パスワード, 7-19
- シングルインスタンス・サンプル・データベースの作成
 - 障害, 7-51
- シングルインスタンス・データベース
 - アップグレード, 7-31
 - インスタンス名, 7-15
 - オンライン化の障害のエラー処理, 7-45
 - 「可能所有者ノード」リスト, 7-12
 - 管理操作, 7-37

- 共有サーバーを使用するためのデータベース・リソースの構成, 7-25
- クライアントによる接続の損失, 7-55
- グループへの追加に対する前提条件, 7-9
- サービス名, 7-15
- 初期化パラメータ・ファイル, 7-16
- フェイルオーバー後の再接続, 7-42
- シングルインスタンス・データベースのアップグレード, 7-31

す

- スクリプト
 - FSCMD コマンドの使用, 5-2, 5-7
 - FsDbError.bat スクリプト, 7-45
 - アクティブ / アクティブ構成の FSCMD コマンド, 3-6
 - シングルインスタンス・データベースをオンライン化できない場合のエラー処理, 7-45
 - スケジューリングしたジョブ
 - Oracle Reports Server, 10-14, 10-26
 - スタンドアロン・データベース
 - Oracle8i と共有サーバー, 7-7
 - Oracle9i と共有サーバー, 7-7
 - 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, 7-3
 - リスナーの検索, 7-6
 - 「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
 - 新規データベースに対する実行, 7-3
 - 説明, 6-2
 - トレース, C-5
- 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
 - スタンバイ・データベース, 3-10
 - スループット
 - 最大化, 3-7
 - スレーブ・サーバー
 - マスター Oracle Reports Server に対する指定, 10-21

せ

- 静的ロード・バランス, 2-17
- セカンダリ・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, 4-4
- セキュリティ
 - FSCMD コマンドによる指定, 5-5
 - Oracle Services for MSCS のセキュリティ設定ツール, 4-7
 - SYSDBA アカウントのパスワードの変更, 7-30
 - シングルインスタンス・データベースの SYSDBA パスワードの変更, 7-30
 - ノード上のパスワード・ファイルの同期化, 7-29
 - 複数仮想アドレスによる向上, 2-11
 - 要件
 - Oracle Applications Concurrent Manager, 12-15
 - Oracle Forms Load Balancer Server, 8-12
 - Oracle Forms Server, 9-15
 - Oracle HTTP Server, 11-15
 - Oracle MTS Service, 13-15
 - シングルインスタンス・データベース, 7-29
 - 汎用サービス, 14-10, 14-15
- 接続時間
 - 改善, 7-21
- 接続モード
 - HTTP, 9-11
 - HTTPS, 9-11
 - Socket, 9-11
- 専用サーバー・モード
 - USE_SHARED_SOCKETS リスナー・パラメータ, 7-21
- 専用ネットワーク・リンク, 3-7

そ

- ソース・ディレクトリ
 - Oracle Reports Server のための指定, 10-14
- ソフトウェア構成, 1-8

た

- 帯域幅
 - ノード間通信の要件, 3-7
 - 複数仮想アドレスによる向上, 2-11

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, 2-18

つ

- 通常ファイル
 - Oracle Reports Server, 10-26
 - フェイルオーバー, 10-26
- 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の損失, 7-55
- ツリー・ビュー, 1-5
 - クラスタの作成, 2-12

て

- 停止
 - データベースのモード, 5-4
- ディスク
 - 構成, 2-4
 - 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, 2-23
- ディレクトリ索引ファイル
 - Oracle HTTP Server, 11-10
- データ・ファイル
 - .fmx ファイル, 9-19
 - Oracle Forms Server, 9-13, 9-19
 - Oracle Reports Server, 10-14
 - Oracle Reports をスケジューリングする .dat ファイル, 10-14
 - 新規の追加, 7-38
- データベース
 - DBA パスワードの変更, 7-50
 - EXCLUSIVE アクセス, 7-29
 - Is Alive ポーリング, 7-50
 - Oracle Database Upgrade Assistant を使用したアップグレード, 7-31
 - Oracle Forms Server によりアクセスされる, 9-14
 - Oracle Reports Server によりアクセスされる, 10-16
 - Oracle7
 - アップグレード, 7-4, 7-32
 - Oracle8i, 7-28
 - SHARED アクセス, 7-29
 - 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, 5-5, 7-27, 7-37
 - FSCMD コマンドの使用, 7-37
 - グループを停止してから再起動, 7-27
 - コールド・バックアップ操作, 7-37
 - 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, 7-50
 - 階層化構成, 9-2
 - 仮想アドレス, 7-14
 - 共有サーバーと MSCS クラスタ・ノード, 7-25
 - 共有サーバーの使用, 7-25
 - 共有サーバーを使用する Oracle8, 7-25
 - 共有サーバーを使用するためのデータベース・リソースの構成, 7-25
 - クライアントによるシングルインスタンス・データベース接続の損失, 7-55
 - グループへの追加
 - 共有ソケット, 7-21
 - グループ・リソース, 2-6
 - 検証, 6-5, 6-7
 - 構成データ, 7-11
 - 構成の手順, 7-10
 - コールド・バックアップ操作のための SQL*Plus の使用, 7-37
 - 識別情報, 7-15
 - 初期化ファイル内の LOCAL_LISTENER パラメータ, 7-28
 - 新規データファイルの追加, 7-38
 - スタンドアロン用のリスナー, 7-6
 - セカンダリ・ノード上の停止テスト, 4-4
 - セキュリティ, 7-29
 - 接続時間の改善, 7-21
 - 専用ネットワーク・リンクを通じたアクセス, 3-7
 - チェックポイント取得, 2-18
 - 透過的アプリケーション・フェイルオーバー・デモ, 7-44, A-1
 - 名前, 7-16
 - ネット・サービス名エントリ, 7-3
 - パスワード・ファイルの同期化, 7-29
 - バックアップ操作, 7-37
 - 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シングルインスタンス・データベースへの再接続, 7-42
リスナーへの依存性の削除, 7-20
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, 7-23
「リソース」も参照
データベース・アカウント
 INTERNAL, 7-55
 SYS, 7-29, 7-55
 パスワードの変更, 7-30
データベース管理者
 DBA_AUTHORIZATION パラメータ, 7-20, 7-29
 Internal としての接続, 7-50
 パスワードの変更, 7-50
「データベース・パスワードの更新」ウィザード, 7-30
データベース・パスワード・ファイル
 データベースへのアカウントの追加, 7-29
データベース・リカバリ
 シングルインスタンス・データベースの最適化,
 7-36
データ・ポート
 Oracle Forms Load Balancer Client, 8-13
 Oracle Forms Load Balancer Server, 8-10
データ・ホスト
 Oracle Forms Load Balancer Client のデフォルト値,
 8-13
デモ
 透過的アプリケーション・フェイルオーバー,
 7-44, A-1

と

透過的アプリケーション・フェイルオーバー, 7-38
 デモ・アプリケーション, 7-44, A-1
動的検出
 Oracle Enterprise Manager, 7-35
登録
 FORMS60_PATH 変数, 9-19
 Oracle Reports Server の変数, 10-30
 ジョブおよびイベント, 7-34
 正しい名前解決を確認する ping の実行, 11-16,
 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-2
 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, 11-9
ドメイン・アカウント
 Oracle Fail Safe Manager, 4-5
 Oracle Reports Server, 4-5
 Oracle Services for MSCS, 4-5
 デフォルト値の更新, 7-3
ドメイン名
 Oracle Services for MSCS アカウントの変更, 4-7
トラブルシューティング, 9-16
 IP アドレスおよびホスト名の登録, 6-3, B-3
 Oracle Applications Concurrent Manager, 12-16
 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, 8-14
 Oracle Forms Server, 9-16
 Oracle HTTP Server, 11-15
 Oracle MTS Service, 13-16
 Oracle Reports Server, 10-28
 仮想アドレスの構成, 7-52
 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, 12-17, 13-17
 グループをオンライン化できない, 7-48, 12-17,
 13-17
 検証ツール, 6-2
 シングルインスタンス・データベースに接続できな
 いユーザー, 7-20
 データベース, 7-44
 ネットワーク・アダプタの順序, 6-3, B-3
 汎用サービス, 14-17
 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, 1-7
 フェイルオーバー後のロールバック, 7-44
トレース
 Oracle Forms Load Balancer Server, 8-11
 使用可能にする, C-3
 フラグ, C-4
トレース・レベル
 Oracle Forms Load Balancer Server, 8-11

な

内部クラスタ通信
 仮想アドレスの構成, 7-23

に

認証

- DBA 認証のトラブルシューティング, 7-50
- 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, 7-29
- SYSDBA ロール, 7-29
- SYSDBA 権限, 7-29
- データベースでの INTERNAL の使用, 7-29
- データベースでの SYS の使用, 7-29
- 「バッチ ジョブでログオン」アクセス権限, 7-55
- 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, 7-42, 7-43
- HOST パラメータ, 7-42

ネット・サービス名

- 仮想サーバーの構成時, 7-52
- 可用性の高いシングルインスタンス・データベース, 7-15
- 新規シングルインスタンス・データベースのエントリ, 7-3
- シングルインスタンス・データベース用の追加, 7-3
- シングルインスタンス・データベースをグループに追加した後のエントリ, 7-22

定義

- アーカイブされた構成ファイルでの参照, 7-54
- デフォルト・ドメイン名による追加, 7-22
- デフォルトのドメイン名, 7-22

ネットワーク

- tnsnames.ora ファイルのプロトコル情報, 4-5
- アクティブ / アクティブ構成でのプライベートの使用, 3-7
- 仮想サーバーの構成時に検出される問題, 7-52
- 構成, B-1
 - 可用性の高いシングルインスタンス・データベース, 7-21
 - 「グループの検証」による検証, 7-46
 - 更新, 7-3
 - 複数のリスナーを伴うノードでの Oracle Net, 7-6
- 構成情報のトレース, C-5

- 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, 7-3
- ネットワーク名
 - クラスタ別名, 2-12
 - グループ用の修正, 7-52

の

ノード

- 可能所有者, 2-22
- 「可能所有者ノード」リストからの除外, 2-22
- 既存クラスタへの追加, 4-11
- クラスタ, 1-3
- グループのフェイルオーバー先, 2-27
- グループを所有する優先順位, 2-27
- 定義, 2-2
- 優先, 2-29

ノード障害

- 影響, 2-31
- フェイルオーバー, 2-16
- 「ノード」プロパティ・ページ
- グループ, 2-23

ノード名

- 汎用サービス, 14-7

は

ハードウェア

- RAID, 1-8
- 構成, 1-8

ハートビート接続

- 定義, 1-8

パスワード

- DBA アカウントのための変更, 7-50
- REMOTE_LOGIN_PASSWORDFILE パラメータでの問題, 7-50
- SYSDBA アカウントのための変更, 7-30
- 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, 7-29

破損

- tnsnames.ora および listener.ora ファイル, 7-52
- 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, 3-5
- データベースの最適化, 3-7
- パブリック・インターコネクト, 1-8
- パラメータ
 - tnsnames.ora ファイルの SERVER=DEDICATED, 7-25
 - USE_SHARED_SOCKETS リスナー・パラメータ, 7-21
- パラメータ・ファイル
 - 「初期化パラメータ・ファイル」を参照
- 汎用サービス
 - LocalSystem アカウント, 14-10
 - Oracle Fail Safe によりサポートされる他のサービスと比較, 14-2
 - Oracle Fail Safe を使用して構成する利点, 14-2
 - Windows 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, 14-9
 - Windows レジストリ・キー, 14-14
 - アカウントの指定, 14-9
 - 依存性, 14-12
 - イメージ名, 14-8
 - 仮想アドレス, 14-14
 - 可能所有者ノード, 14-6
 - 可用性を高める構成にしない汎用サービス, 14-3
 - 可用性を高めるための構成, 14-3, 14-5
 - 可用性を高めるための構成手順, 14-4
 - 起動タイプ, 14-10
 - 起動パラメータ, 14-9
 - グループへの追加, 14-4
 - 構成手順, 14-5
 - サービス名, 14-8
 - 作成およびグループへの追加, 14-5
 - 実行可能ファイルの配置場所, 14-8
 - スタンドアロンの検出, 14-3
 - セキュリティ要件, 14-15
 - 定義, 2-8, 14-1
 - データ・ファイルの配置場所, 14-11
 - トラブルシューティング, 14-17
 - ノード名, 14-7
 - 汎用サービスの識別情報, 14-7
 - 表示名, 14-8
 - リソース・タイプ, 2-8
 - 「リソース」も参照
- 汎用サービスのグループへの追加, 14-3
 - アカウント, 14-9
 - 依存性, 14-12

- イメージ名, 14-8
- 仮想アドレス, 14-14
- 可能所有者ノード, 14-6
- 起動パラメータ, 14-9
- クラスタ・ディスク, 14-11
- サービス名, 14-8
- サンプル・サービス, 14-16
- 手順, 14-4
- ノード名, 14-7
- 表示名, 14-8
- レジストリ・キー, 14-14

ひ

- 表示名
 - 汎用サービス, 14-8

ふ

- ファイバ・チャネル
 - 共有記憶装置インターコネクト, 1-8
- ファイル
 - アーカイブ, 7-54
 - ロールバック, 7-54
- ファイルの破損
 - tnsnames.ora および listener.ora ファイル, 7-52
- フィジカル・スタンバイ・データベース, 3-10
- プール・パラメータ
 - Oracle Forms Server, 9-12
- フェイルオーバー, 2-13
- Oracle Intelligent Agent リソース, 7-36
- Web サーバー, 11-3
- アクティブ / アクティブ構成, 3-4
- アプリケーション, 1-7
- アプリケーションの自動再接続, 1-7
- 「可能所有者ノード」リストによる影響, 2-27
- クライアント・アプリケーション, 2-31
- グループ, 2-6
- グループの移動先のノード, 2-27
- 「グループの検証」操作による検証, 6-5, 7-46
- 計画外, 2-13
- 計画的, 2-13, 2-17
- 「検証」コマンドによるチェック, 4-3
- 最高速, 3-4
- 単位, 2-6
- 通常ファイル, 10-26
- 定義, 2-2

- データベースのチェックポイント取得, 2-18
- トラブルシューティング, 7-50
- トランザクションの再実行, 1-7
- ノード障害, 2-16
- バックアップ操作中の使用不可, 7-37
- 反復を制限する, 2-24
- 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, 7-44
- 「優先所有者ノード」リストによる影響, 2-27
- リソース再起動ポリシーによる影響, 2-26
- リソース障害, 2-14
- ロード・バランス, 2-13
- フェイルオーバー期間, 2-25
- フェイルオーバーしきい値, 2-24
- フェイルオーバー・ポリシー
 - グループ, 2-24
- フェイルバック
 - グループの移動操作による影響, 2-30
 - 「優先所有者ノード」リストによる影響, 2-30
- 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, 2-29
 - 情報のダンプ, 6-10
- 複数 Oracle ホーム, 4-9
 - listener.ora ファイルのシングルインスタンス・データベース SID, 7-52
 - アドレスおよびシングルインスタンス・データベース SID 名の重複のチェック, 7-53
 - 「スタンドアロン・データベースの検証」コマンドによる tnsnames.ora の更新, 7-3
- 複数のリスナー, 7-6
- 複数のリスナーの構成, 7-6
- 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-4
 - Windows NT Service Pack 5 以上, B-4
- 不十分な権限, 7-55
- プライベート・インターコネクト, 1-8, 2-2
- 帯域幅, 3-7
- プライベート・ディスク
 - 汎用サービス, 14-11
- フラグ
 - Oracle Services for MSCS のエラーのトレース, C-5
- プリンタ
 - Oracle Reports Server, 10-17
- プロトコル・アドレス, 7-21

へ

変更

- Oracle Services for MSCS アカウントのドメイン名, 4-7
- 「可能所有者ノード」リスト, 2-23

ほ

ポート

- Oracle Forms Server, 9-10
- Oracle Reports Server, 10-13

ポート番号

- Oracle Forms Server 用の検証, 9-17

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, 11-16, B-2
- 登録, B-2
- 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, 7-49, 12-17
- トラブルシューティング, 7-49, 12-17

ま

マスター Oracle Reports Server

- 高い可用性の提供, 10-1

マスター / スレーブ実装

- Oracle Reports Server のための構成, 10-20

マルチスレッド・サーバー

- 「共有サーバー」を参照

マルチフォーム・アプリケーション, 9-19

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, 3-6

め

メタデータ

- クォーラム・リソース, 2-5

も

文字列

Oracle Services for MSCS のトレース, C-4

ゆ

ユーザー

Oracle Fail Safe Manager のドメイン・アカウント,
4-8

Oracle Services for MSCS 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, 10-17

ユーザー・セッション

フェイルオーバー後の再確立, 7-43, 7-44

ユーザー設定項目

設定, 7-55

「優先所有者ノード」リスト

グループの移動操作, 2-30

データベース停止のテスト, 4-4

フェイルオーバーへの影響, 2-27

フェイルバックへの影響, 2-30

り

リクエスト・ポート

Oracle Forms Load Balancer Server, 8-11

リスナー

DISPATCHERS パラメータへの情報指定, 7-28

LOCAL_LISTENER パラメータへの指定, 7-28

起動, 7-6

共有サーバーのための変更, 7-7

共有ソケットを使用可能にする, 7-21

グループに追加するシングルインスタンス・データ
ベース用に作成, 7-21

構成, 4-3

作成時の問題, 7-54

出力ファイル, 7-54

新規作成, 7-54

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・リソース, 2-6

複数のリスナーを伴うノードでの Oracle Net の構
成, 7-6

リスナー制御ユーティリティ (LSNRCTL) を使用
した作成, 7-54

リスナーで変更された動作, 7-4

ロールバック・ファイル内の定義, 7-53

ログ・ファイル, 7-48

リスナー制御ユーティリティ (LSNRCTL)

リスナーの作成, 7-54

リソース, 1-2

DLL, 2-8

Oracle Intelligent Agent, 7-35

アクセス, 2-5

依存性, 2-7

「一般」プロパティ・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, 2-11

「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」プロパティ・ページでの変更
による影響, 2-23

検出, 4-8

構成の誤りの修正, 7-46

構成の検証, 6-5, 6-7

再起動ポリシー, 2-26

障害, 2-14

定義, 2-5

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, 2-24

優先所有者ノードに戻ることに
関する, 2-29

例, 2-6

リソース・タイプ

IP アドレス, 2-8

MSCS, 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 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
- リソースへのアクセス, 2-5
- 「リソースをグループに追加」ウィザード
 - Oracle Applications Concurrent Manager, 12-7
 - Oracle Forms Load Balancer Server, 8-7
 - Oracle Forms Server, 9-6
 - Oracle HTTP Server, 11-6
 - Oracle Intelligent Agent, 7-35
 - Oracle MTS Service, 13-7
 - Oracle Reports Server, 10-9
 - Oracle シングルインスタンス・データベース, 7-11
 - 情報のロギング, C-5
 - 汎用サービス, 14-5
- 「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, 7-4
- 利点
 - Oracle Fail Safe, 1-3
 - Oracle Fail Safe を使用した汎用サービスの構成, 14-2

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-2
- レジストリ・キー
 - Oracle Applications Concurrent Manager 構成 ID, 12-5
 - Oracle Forms Server 仮想アドレス, 9-9
 - Oracle Forms Server ポート番号, 9-10
 - Oracle Forms 接続モード, 9-12
 - Oracle Forms プール・パラメータ, 9-12
 - Oracle Reports キャッシュ・ディレクトリ, 10-30
 - クラスタ全体に複製, 14-14
 - 汎用サービス, 14-14
- レジストリ変数
 - FORMS60_PATH, 9-13
 - Oracle Applications Concurrent Manager 構成 ID, 12-11
 - Oracle Reports キャッシュ・ディレクトリ, 10-30
 - REPORTS60_CGIMAP, 10-18
 - REPORTS60_PHYSICAL_MAP, 10-30
 - REPORTS60_VIRTUAL_MAP, 10-30

ろ

- ロード・バランス, 2-13, 8-2
 - Oracle Forms Server, 8-2
 - Oracle Reports Server, 10-6
 - アクティブ / アクティブ構成, 3-6
 - 静的, 2-17
 - 「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, 7-53, 7-54
- ロジカル・スタンバイ・データベース, 3-10

わ

- ワーカー・プロセス
 - ifweb60.exe, 9-12
 - プール, 9-12