

# Oracle8i

スタンバイ・データベース 概要および管理

リリース 8.1

2000 年 2 月

部品番号 : J00919-01

ORACLE®

---

Oracle8i スタンバイ・データベース 概要および管理, リリース 8.1

部品番号 : J00919-01

原本名 : Oracle8i Standby Database Concepts and Administration, Release 2 (8.1.6)

原本部品番号 : A76995-01

原本著者 : Lance Ashdown, Anna Logan

原本協力者 : Rick Anderson, Mark Johnson, Steve Lee, Barbara Benton

Copyright © 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されております。

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 .....	ix
------------	----

## 1 スタンバイ・データベースの概念

スタンバイ・データベース .....	1-2
構成オプション .....	1-2
利点および不利な点 .....	1-3
互換性および運用上の要件 .....	1-4
概念および用語 .....	1-5
スタンバイ・データベースのモード .....	1-7
手動リカバリ・モード .....	1-7
管理リカバリ・モード .....	1-8
読取り専用モード .....	1-9
スタンバイ・データベースへのフェイルオーバー .....	1-11
フェイルオーバーの結果 .....	1-12
フェイルオーバーを実行しないスタンバイ・データベースのテスト .....	1-14
フェイルオーバー後の元のプライマリ・データベースの再作成 .....	1-14
スタンバイ・データベースのライフ・サイクル .....	1-15
スタンバイ・データベースの作成 .....	1-15
ギャップ・シーケンス内のログを使用した手動リカバリ .....	1-16
管理リカバリおよび読取り専用のアクセス・サイクル .....	1-17
スタンバイ・データベースへのフェイルオーバー .....	1-18
スタンバイ・データベース環境の構成 .....	1-18
スタンバイ・データベースの数 .....	1-18
スタンバイ・サイトへのアーカイブ REDO ログの転送方法 .....	1-20
プライマリ・サイトおよびスタンバイ・サイトの場所およびディレクトリ構造 .....	1-21

スタンバイ・データベースのメンテナンス .....	1-25
アーカイブ REDO ログの状態のチェック .....	1-25
スタンバイ・データベースのバックアップ .....	1-26
プライマリ・データベースの物理的変更の反映 .....	1-26
スタンバイ・データベースの文 .....	1-27

## 2 スタンバイ・データベースの準備

スタンバイ・データベース作成前の環境評価 .....	2-2
スタンバイ・データベースの更新方法 .....	2-2
スタンバイ・データベースの別のホストへの配置 .....	2-3
プライマリ・データベースの中でのオンライン REDO ログの構成方法 .....	2-3
フェイルオーバー後のスタンバイ・データベースの接続方法 .....	2-4
スタンバイ・データベースのセットアップ: 基本作業 .....	2-4
管理リカバリのためのスタンバイ・データベースの準備: 基本作業 .....	2-5
手動リカバリのためのスタンバイ・データベースの準備: 基本作業 .....	2-6
スタンバイ・データベース・ファイルの作成 .....	2-7
スタンバイのデータ・ファイルの作成 .....	2-7
スタンバイの制御ファイルの作成 .....	2-8
スタンバイ・サイトへのファイル転送 .....	2-9
プライマリおよびスタンバイ・データベース用のネットワーク・ファイルの構成 .....	2-10
プライマリ・サイトの tnsnames.ora ファイルの構成 .....	2-12
スタンバイ・サイトの listener.ora ファイルの構成 .....	2-13
プライマリ・データベースの初期化パラメータ・ファイルの構成 .....	2-15
アーカイブ先の指定 .....	2-16
必須およびオプションのアーカイブ先の指定 .....	2-17
アーカイブ先の状態を使用可能にする方法 .....	2-17
失敗した宛先に再アーカイブするための Oracle の構成 .....	2-18
スタンバイ・データベースの初期化パラメータ・ファイルの構成 .....	2-19
スタンバイの初期化パラメータの構成: 一般的な考慮事項 .....	2-19
スタンバイの制御ファイル内のプライマリのファイル名の改名 .....	2-21
スタンバイ・データベースのアーカイブ REDO ログのファイル名の指定 .....	2-23
スタンバイの初期化パラメータ・ファイルの構成: 典型的な設定値 .....	2-25
パラメータおよびネットワーク・ファイルの構成: シナリオ .....	2-27
リカバリの準備としてのスタンバイ・インスタンスの起動 .....	2-30
スタンバイ・インスタンスの起動 .....	2-31

変換パラメータで取り込めないスタンバイのファイルの手動での改名 .....	2-31
初期化パラメータの設定値の変更の有効化 .....	2-32
<b>スタンバイ・データベース構成のトラブルシューティング .....</b>	<b>2-33</b>
スタンバイ・サイトはプライマリ・データベースによってアーカイブされたログを受け取っていない場合 .....	2-33
スタンバイ・データベースをマウントできない場合 .....	2-34

### 3 スタンバイ・データベースの管理

<b>スタンバイ・データベースのモードの選択 .....</b>	<b>3-2</b>
スタンバイ環境の典型的なモード .....	3-2
手動リカバリが必要な状況 .....	3-3
<b>スタンバイ・データベースの手動リカバリ・モード設定 .....</b>	<b>3-3</b>
<b>管理リカバリ開始前のギャップ・シーケンスの解決 .....</b>	<b>3-5</b>
ギャップ・シーケンスの原因 .....	3-5
ギャップ・シーケンスが存在するかどうかの確認 .....	3-8
ギャップ・シーケンス内のログのスタンバイ・サイトへの転送 .....	3-10
ギャップ・シーケンス内のログのスタンバイ・データベースへの適用 .....	3-12
<b>スタンバイ・データベースの管理リカバリ・モード設定 .....</b>	<b>3-13</b>
管理リカバリ・モードの開始 .....	3-13
管理リカバリの取消し .....	3-15
<b>読取り専用モードでスタンバイ・データベースをオープン .....</b>	<b>3-16</b>
読取り専用モードを実行するかどうかの検討 .....	3-16
読取り専用モードでのアーカイブ REDO ログの受取り .....	3-17
読取り専用モードのデータベースのオープン .....	3-17
一時表領域の作成 .....	3-18
<b>スタンバイ・データベースのアクティブ化 .....</b>	<b>3-19</b>
<b>Oracle Parallel Server 構成でのスタンバイ・データベースの使用 .....</b>	<b>3-22</b>

### 4 スタンバイ・データベースのメンテナンスの実行

<b>スタンバイ・データベースに影響を与えるイベントのモニター .....</b>	<b>4-2</b>
プライマリ・データベースおよびスタンバイ・データベースのモニター .....	4-2
スタンバイ・サイトで受け取ったアーカイブ・ログの判別 .....	4-5
スタンバイ・データベースに適用されたログの判別 .....	4-9
<b>スタンバイ・データベースに影響を与えるイベントへの応答 .....</b>	<b>4-10</b>
プライマリ・データベースへの表領域またはデータ・ファイルの追加 .....	4-10

プライマリ・データベースのデータ・ファイルの改名 .....	4-12
プライマリ・データベースの REDO ログの追加または削除 .....	4-12
プライマリ・データベースのアーカイブしていない REDO ログのリセットまたは消去 .....	4-12
プライマリ・データベースの制御ファイルの変更 .....	4-13
スタンバイ・データベースのデータ・ファイルをオフラインにする方法 .....	4-13
ダイレクト・パス処理の実行 .....	4-13
スタンバイ・データベースの制御ファイルのリフレッシュ .....	4-15
オンライン REDO ログの消去 .....	4-16
スタンバイ・データベースのバックアップ .....	4-16

## 5 スタンバイ・データベースのシナリオ

<b>シナリオ 1: 同一ホスト上のスタンバイ・データベースの作成</b> .....	5-2
ステップ 1: スタンバイ・データベースの計画 .....	5-2
ステップ 2: スタンバイ・データベースの作成 .....	5-3
ステップ 3: ネットワーク・ファイルの構成 .....	5-4
ステップ 4: プライマリ・データベースのパラメータ・ファイルの構成 .....	5-6
ステップ 5: スタンバイ・データベースのパラメータ・ファイルの構成 .....	5-8
ステップ 6: 管理リカバリの準備としてのスタンバイ・インスタンスの起動 .....	5-10
ステップ 7: ギャップ・シーケンス内のログの識別 .....	5-11
ステップ 8: ギャップ・シーケンス内のログのスタンバイのファイル・システムへの適用 .....	5-12
ステップ 9: ギャップ・シーケンス内のログのスタンバイ・データベースへの適用 .....	5-13
ステップ 10: スタンバイ・データベースの管理リカバリ・モード設定 .....	5-14
<b>シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成</b> .....	5-14
ステップ 1: プライマリ・データベースのデータ・ファイルのバックアップ .....	5-14
ステップ 2: スタンバイ・データベースの制御ファイルの作成 .....	5-15
ステップ 3: データ・ファイルおよび制御ファイルのスタンバイ・サイトへの転送 .....	5-16
ステップ 4: ネットワーク・ファイルの構成 .....	5-16
ステップ 5: スタンバイ・サイトでのリスナーの起動 .....	5-17
ステップ 6: スタンバイの初期化パラメータ・ファイルの構成 .....	5-17
ステップ 7: スタンバイの初期化パラメータ・ファイルのコピー .....	5-19
ステップ 8: スタンバイ・データベースの起動 .....	5-19
ステップ 9: プライマリの初期化パラメータ・ファイルの構成 .....	5-19
ステップ 10: ギャップ・シーケンス内のログの適用 .....	5-20
ステップ 11: スタンバイ・データベースの管理リカバリ・モード設定 .....	5-21

<b>シナリオ 3: プライマリ・データベースの物理的変更への対応</b> .....	5-21
プライマリ・データベースへのデータ・ファイルの追加 .....	5-22
プライマリ・データベースのデータ・ファイルの改名 .....	5-24
プライマリ・データベースのデータ・ファイルまたは表領域の削除 .....	5-25
オンライン REDO ログの追加または削除 .....	5-25
制御ファイルの変更 .....	5-26
スタンバイ・データベースの制御ファイルのリフレッシュ .....	5-26
スタンバイ・データベースの再構築が必要な物理的変更 .....	5-28
<b>シナリオ 4: NOLOGGING 句の指定後のリカバリ</b> .....	5-28
<b>シナリオ 5: 複数スタンバイ・データベース構成でのフェイルオーバー先の スタンバイ・データベースの決定</b> .....	5-31
<b>シナリオ 6: クライアント・アプリケーション・フェイルオーバーの構成</b> .....	5-36
ローカル TNS 構成 .....	5-36
Oracle Names Server 構成 .....	5-37
透過的アプリケーション・フェイルオーバー (TAF) 構成 .....	5-37
手動のネットワーク構成 .....	5-38
<b>シナリオ 7: ネットワーク障害後のリカバリ</b> .....	5-39
<b>シナリオ 8: スタンバイ・データベースの再作成</b> .....	5-43
<b>シナリオ 9: 進行中のリカバリがないスタンバイ・データベース</b> .....	5-46
進行中のリカバリがないスタンバイ・データベースの管理 .....	5-46
進行中のリカバリがないスタンバイ・データベースのアクティブ化 .....	5-48
<b>シナリオ 10: 時間差を設定したスタンバイ・データベース</b> .....	5-50
時間差を設定したスタンバイ・データベースの作成 .....	5-51
時間差を設定したスタンバイ・データベースの管理 .....	5-53
データベースの指定時刻へのロールバック .....	5-55
時間差をバイパスしたスタンバイ・データベースのアクティブ化 .....	5-55

## 索引



1-1	モードの切替え .....	1-7
1-2	スタンバイ・データベースの手動リカバリ・モード .....	1-8
1-3	スタンバイ・データベースの自動更新 .....	1-9
1-4	スタンバイ・データベースの読取り専用モード .....	1-10
1-5	スタンバイ・データベースへのフェイルオーバー .....	1-11
1-6	複数のスタンバイ・データベースを持つプライマリ・データベース .....	1-13
1-7	スタンバイ・データベースのライフ・サイクル .....	1-15
1-8	スタンバイ・データベースの作成 .....	1-16
1-9	異なるバックアップを使用したスタンバイ・データベースの作成 .....	1-19
1-10	非管理リカバリ環境 .....	1-21
1-11	考えられるスタンバイ構成 .....	1-23
2-1	ファイル名変換パラメータの設定 .....	2-23
3-1	ギャップ・シーケンス内のアーカイブ・ログの手動リカバリ .....	3-6
3-2	スタンバイ・データベースへのアーカイブ REDO ログの転送および適用 .....	3-13
3-3	複数インスタンス・プライマリ・データベースからのアーカイブ・ログの転送 .....	3-23



## 表

1-1	スタンバイ・データベースの文 .....	1-27
2-1	オンライン REDO ログの構成の影響 .....	2-3
2-2	作業リスト: 管理リカバリの準備 .....	2-5
2-3	作業リスト: 手動リカバリの準備 .....	2-6
2-4	スタンバイ・データベースの初期化パラメータの構成 .....	2-20
2-5	ファイル名の変換 .....	2-22
2-6	スタンバイ・データベースの初期化パラメータの設定 .....	2-25
4-1	コマンドの波及 .....	4-3
5-1	スタンバイ・データベースの初期化パラメータの構成 .....	5-9



---

# はじめに

スタンバイ・データベースは、Oracle8i のバックアップおよびリカバリのソリューションに向けてオラクル社が提供する最も強力な追加機能の 1 つです。このマニュアルでは、スタンバイ・データベースの作成、管理およびメンテナンスに必要な、概念と作業別の情報を説明します。

**関連項目：** バックアップおよびリカバリの一般的な情報は、『Oracle8i バックアップおよびリカバリ・ガイド』を参照してください。Recovery Manager ユーティリティに関連する概念、手続きおよび参照情報は、『Oracle8i Recovery Manager ユーザーズ・ガイドおよびリファレンス』を参照してください。

# Oracle8i の新機能

この項では、Oracle リリース 8.1.5 およびリリース 8.1.6 でのスタンバイ・データベースに関連する新機能を説明します。

## リリース 8.1.6

リリース 8.1.6 の新機能を使用し、LOG\_ARCHIVE\_TRACE 初期化パラメータを設定して、アーカイブ・アクティビティを記述するトレース・ファイルを生成できます (4-5 ページの「[スタンバイ・サイトで受け取ったアーカイブ・ログの判別](#)」を参照)。

## リリース 8.1.5

リリース 8.1.5 の新機能は次のとおりです。

- Oracle8i では、バックグラウンド・アーカイブ・プロセス (ARCH) または対応するフォアグラウンド・プロセスが、オンライン REDO ログ・ファイルを複数の宛先にアーカイブできるようにする方法を提供します (2-16 ページの「[アーカイブ先の指定](#)」を参照)。次の初期化パラメータを使用して、最大 5 つの宛先を指定できます。LOG\_ARCHIVE\_DEST\_ *n* (この *n* は 1 から 5 までの整数)。ローカル・ディスク・ベースのファイルを宛先にもでき、プライマリ・データベースに対してユーザーが指定したローカルまたはリモートのスタンバイ・データベースを宛先にもできます。
- スタンバイ・データベースを管理リカバリ・モードで実行すれば、本番データベースでアーカイブ REDO ログが生成されるたびに、スタンバイ・データベースにログが自動的に適用されるように設定できます (1-8 ページの「[管理リカバリ・モード](#)」を参照)。
- プライマリ・データベース・サイトからアーカイブ REDO ログの転送中も問合せおよびレポートにスタンバイ・データベースを使用するために、読取り専用モード・オプションを使用できます (1-9 ページの「[フェイルオーバーを実行しないスタンバイ・データベースのテスト](#)」を参照)。

# 構造

このマニュアルは、次の章から構成されています。

章	内容
<a href="#">第 1 章「スタンバイ・データベースの概念」</a>	Oracle8i スタンバイ・データベースの概要を説明します。
<a href="#">第 2 章「スタンバイ・データベースの準備」</a>	スタンバイ・データベースの作成方法を説明します。
<a href="#">第 3 章「スタンバイ・データベースの管理」</a>	スタンバイ・データベースを手動リカバリ・モード、管理リカバリ・モードおよび読取り専用モードで管理するためのガイドラインを説明します。

---

章	内容
第4章「スタンバイ・データベースのメンテナンスの実行」	スタンバイ・データベースの一般的なメンテナンス操作について、ステップ別に説明します。
第5章「スタンバイ・データベースのシナリオ」	スタンバイ・データベースの共通のシナリオについて説明します。

---

## このマニュアルでの変更点

このマニュアルはリリース 8.1.6 の新規マニュアルです。リリース 8.1.5 では、バックアップおよびリカバリの情報はすべて『Oracle8i バックアップおよびリカバリ・ガイド』に記載されています。リリース 8.1.6 では、バックアップおよびリカバリの情報は、次のマニュアルに分散して記載されています。

- 『Oracle8i バックアップおよびリカバリ・ガイド』
- 『Oracle8i Recovery Manager ユーザーズ・ガイドおよびリファレンス』
- 『Oracle8i スタンバイ・データベース 概要および管理』

## 対象読者

このマニュアルは、Oracle データベース・システムのバックアップ、リストアおよびリカバリ操作を管理する、データベース管理者（DBA）を対象としています。

## 読者に要求される知識

このマニュアルの読者は、リレーショナル・データベースの概念、バックアップおよびリカバリの基本的な管理について理解する必要があります。また、Oracle が作動するオペレーティング・システム環境を理解していることも前提となります。

## 規則

この項では、次の項目のこのマニュアルで使用されている規則について説明します。

- テキスト
- コード例

---

## テキスト

この項では、テキストで使用されている規則について説明します。

### 大文字

表領域名、初期化パラメータおよび SQL キーワードに注意を促すために大文字のテキストを使用しています。

たとえば、「プライベート・ロールバック・セグメントを作成する場合には、名前を、初期化パラメータ・ファイルの ROLLBACK\_SEGMENTS パラメータに含める必要があります。この情報は、SQL\*Plus で SHOW PARAMETER 文を発行して表示できます。」

### イタリック

テキスト内のイタリックの語句は、SQL 構文の変数です。

たとえば、「アーカイブ REDO ログはオフラインでコピーされたオンライン REDO ログです。この機能を使用可能にするには、データベースを ARCHIVELOG モードで実行する必要があります。Recovery Manager を使用する場合、**backup** コマンドの中で、**archivelog** に似た `'/oracle/archive/arc_*` 副次句を使用して、アーカイブ REDO ログを指定できます。」

### 太字

テキスト中の太字の語句は、オペレーティング・システム固有のコマンドです。

たとえば、「データベースをバックアップするには Recovery Manager の **backup** コマンドを使用してください。UNIX の場合は、**cp** コマンドを使用してファイルをコピーできます。」

### 固定幅フォント

ファイル名およびディレクトリは、固定幅フォントで表示します。

たとえば、「プライマリ・サイトの tnsnames.ora ファイル内の情報は、スタンバイ・サイトの listener.ora ファイル内の情報と対応している必要があります。」

## コード例

SQL および SQL\*Plus 文は、テキストの段落とは別に、固定幅フォントで表示されます。次に例を示します。

```
INSERT INTO emp (empno, ename) VALUES (1000, 'SMITH');
ALTER TABLESPACE users ADD DATAFILE 'users2.ora' SIZE 50K;
run {
    allocate channel ch1 type disk;
    backup database;
}
```

---

SQL および SQL\*Plus 文は、異なるプラットフォーム上の異なる環境で実行できます。このマニュアルでは可能な限り、特定のオペレーティング・システムまたはインタフェースの独自の情報ではなく、汎用の情報を提供します。ただし、オペレーティング・システム・レベルで構文の働きを具体例で説明する必要があることもあります。このような場合、このマニュアルでは UNIX コマンドライン・インタフェースの例を使用し、オペレーティング・システム・プロンプトを示すために % 記号を使用します。

## このマニュアルの使用法

このマニュアルの読者は、次の資料を読んでいることを想定しています。

- Oracle に関連する概要および用語を説明し、このマニュアル内のより詳細な情報の基礎となる、『Oracle8i 概要』マニュアルの最初の部分。『Oracle8i 概要』マニュアルの残りの部分では、Oracle のアーキテクチャおよび機能を詳細に説明しています。
- 『Oracle8i 管理者ガイド』の、制御ファイル、オンライン REDO ログおよびアーカイブ REDO ログの管理に関する章。

また、次のリファレンス・ガイドを参照する必要があります。

- 『Oracle8i SQL リファレンス』
- 『Oracle8i リファレンス・マニュアル』
- 『Oracle8i バックアップおよびリカバリ・ガイド』
- 『Oracle8i Recovery Manager ユーザーズ・ガイドおよびリファレンス』

---



---

# スタンバイ・データベースの概念

この章では、スタンバイ・データベースの特徴および機能について説明します。この章は次のトピックから構成されています。

- [スタンバイ・データベース](#)
- [スタンバイ・データベースのモード](#)
- [スタンバイ・データベースへのフェイルオーバー](#)
- [スタンバイ・データベースのライフ・サイクル](#)
- [スタンバイ・データベース環境の構成](#)
- [スタンバイ・データベースのメンテナンス](#)
- [スタンバイ・データベースの文](#)

## スタンバイ・データベース

スタンバイ・データベースは、プライマリ・データベースのバックアップから作成されるデータベースの複製です。プライマリ・データベースからスタンバイ・データベースにアーカイブ REDO ログを適用することで、2つのデータベースを同期化させることができます。

スタンバイ・データベースの主な目的は次のとおりです。

- 災害時の保護
- データ破損に対する保護
- 補足的レポート

プライマリ・データベースが破壊またはデータが破損した場合、スタンバイ・データベースへのフェイルオーバーを実行して、スタンバイ・データベースを新しいプライマリ・データベースに変更できます。また、読取り専用オプションを使用してスタンバイ・データベースをオープンすれば、スタンバイ・データベースを独立したレポート・データベースとして使用できます。

この項は、次のトピックから構成されています。

- [構成オプション](#)
- [利点および不利な点](#)
- [互換性および運用上の要件](#)
- [概念および用語](#)

## 構成オプション

次の作業方法によって、スタンバイ・データベースの設定方法も異なります。

- アーカイブ REDO ログのスタンバイ・サイトへの転送
- アーカイブ REDO ログのスタンバイ・データベースへの適用

たとえば、Oracle の管理スタンバイ環境を使用すると、スタンバイ・インスタンスが起動されている限り、プライマリ・データベースはスタンバイ・データベース・サイトに REDO ログを自動的にアーカイブできます。非管理スタンバイ環境をインプリメントした場合には、ログを手動で転送する必要があります。

スタンバイ・データベースが管理リカバリ・モードになっている場合には、スタンバイ・データベースはプライマリ・データベースから受け取ったログを自動的に適用します。スタンバイ・データベースを手動リカバリ・モードに変更し、スタンバイ・データベースに手動でログを適用することもできます。スタンバイ・データベースは、レポートに使用するために、いつでも読取り専用モードでオープンできます。

次の表は、ユーザーが選択する環境に基いて、使用可能な構成を示しています。

環境	転送方法	スタンバイ・データベースのモード	ネットワーク要件
管理	自動（または、必要に応じて手動）	管理リカバリ、手動リカバリまたは読取り専用	Net8
非管理	手動のみ	手動リカバリまたは読取り専用	なし

ほとんどのデータベース管理者（DBA）は管理リカバリ環境を選択しています。次のような場合には、非管理環境を選択することも考えられます。

- プライマリ・サイトとスタンバイ・サイト間で、管理リカバリのために必要な Net8 接続を保持しない場合。
- プライマリ・データベースでログをアーカイブしてから、スタンバイ・データベースにログを適用するまでの時間をあける場合。時間をあけることにより、破損したデータまたはエラーのあるデータが、プライマリ・データベースからスタンバイ・データベースへの転送を防止できます。

**関連項目：** スタンバイ・データベースの各モードの詳細は、1-7 ページの「[スタンバイ・データベースのモード](#)」を参照してください。スタンバイ・データベース環境の構成オプションの詳細は、1-18 ページの「[スタンバイ・データベース環境の構成](#)」を参照してください。時間差を付けるようにスタンバイ・データベースを設定する方法の詳細は、5-50 ページの「[シナリオ 10: 時間差を設定したスタンバイ・データベース](#)」を参照してください。

## 利点および不利な点

スタンバイ・データベースは、災害対策および補足的レポートの両方にとって、強力なツールになります。たとえば、次のようなことが可能になります。

- プライマリ・データベースから地理的に離れた場所にスタンバイ・データベースを保持できます。あるいは、地理的に分散した複数の場所に複数のスタンバイ・データベースを保持できます。
- プライマリ・データベースとスタンバイ・データベースを同じマシンの別々のディスク・ドライブに保持しておくと、プライマリ・データベースのドライブに障害が発生した場合、スタンバイ・データベースをアクティブにし、通常運用を再開できます。
- スタンバイ・データベースがプライマリ・データベースから自動的にスタンバイ・サイトに転送されたアーカイブ REDO ログを自動的に適用するように、管理スタンバイ構成をインプリメントできます。この方法を使用すれば、プライマリ・データベースへの変更を、スタンバイ・データベースに定期的に反映させることができます。

- プライマリ・データベースが完全に破壊された場合、時間およびデータの損失を最小限におさえて、スタンバイ・データベースを新しいプライマリ・データベースに変更できます。
- 破損したデータを含むアーカイブ・ログをスタンバイ・データベースに適用しないことによって、プライマリ・データベースでのエラーのあるバッチ・ジョブ、ユーザー・エラー（たとえば、間違った表の切捨て）またはアプリケーションの破損に対する保護機能を提供します。その後、破損していないスタンバイ・データベースをアクティブにし、これをプライマリ・データベースにすることができます。

スタンバイ・データベースはバックアップおよびリカバリ戦略上、多大な利点を提供しますが、コストもかかります。たとえば、スタンバイ・データベースでは次のコストが必要になります。

- スタンバイ・データベースを別のホスト上に保持して災害対策を万全にするには、コンピュータを追加する必要がある
- 管理スタンバイ環境を使用するには、Net8 接続の実装およびメンテナンスが必要になる
- 選択した実装方法に関わらず、システム・リソースおよび記憶領域を追加する必要がある
- プライマリ・データベースで実行される構造的操作（たとえば、表領域またはデータ・ファイルの追加）をミラー化するために、スタンバイ・データベースを管理する必要がある

## 互換性および運用上の要件

スタンバイ・データベースを保持する場合、次のような要件が発生します。

- プライマリ・データベースは ARCHIVELOG モードで実行する必要があります。
- 手動リカバリ・モードのスタンバイ・データベースは、Oracle リリース 7.3 またはこれ以降のリリースでのみ稼動します。
- 管理リカバリ・モードのスタンバイ・データベースは、Oracle リリース 8.1 またはこれ以降のリリースでのみ稼動します。
- 読取り専用モードのスタンバイ・データベースは、Oracle リリース 8.1.5 またはこれ以降のリリースでのみ稼動します。
- スタンバイ・データベースに適用する REDO ログは、アーカイブ REDO ログまたは現行ではないオンライン REDO ログである必要があります。現行の REDO ログ中のトランザクションは、手動でのアーカイブで使用できます。
- プライマリおよびスタンバイのホストでは、同バージョンおよび同リリースのオペレーティング・システムを使用する必要があります。ただし、スタンバイ・ホストでは、異なるディレクトリ構造を使用できます。

- フェイルオーバー操作が影響を受けないように、プライマリ・データベースおよびスタンバイ・データベースでは Oracle RDBMS の同バージョン、同リリースおよび同パッチを使用する必要があります。
- プライマリ・データベースとスタンバイ・データベースで同じ制御ファイルを共有することはできません。
- 同じホスト上にプライマリ・データベースおよびスタンバイ・データベースを保持するには、オペレーティング・システムによっては、同じデータベース名を持つ2つのインスタンスを同じマシン上に同時にマウントできないことがあります。これについては、どのプラットフォームを使用している場合でも解決方法があります。
- スタンバイ・データベースをアクティブにした後、スタンバイ・データベースを管理リカバリ・モードに戻すことはできません。アクティブにされたスタンバイ・データベースは、通常のプライマリ・データベースになります。

## 概念および用語

これ以降の章では、次の用語を使用します。

### アクティブ化

「フェイルオーバー」を参照してください。

### フェイルオーバー

スタンバイ・データベースを、通常のプライマリ・データベースとするための操作。この操作は、スタンバイ・データベースのアクティブ化とも呼ばれます。フェイルオーバー後には、スタンバイ・データベースを再度スタンバイ・データベースに戻すことはできません。

### ギャップ・シーケンス

スタンバイ・データベースを管理リカバリ・モードにする前に、手動で適用する必要のある一連のアーカイブ REDO ログ。

### 管理リカバリ・モード

次の SQL\*Plus 文を入力することにより開始される、スタンバイ・データベースのモード。

```
RECOVER MANAGED STANDBY DATABASE;
```

管理リカバリ・モードで実行されるスタンバイ・データベースでは、プライマリ・データベースから受け取った REDO ログは自動的に適用されます。

### 管理スタンバイ環境

プライマリ・データベースが、スタンバイ・サイトに REDO ログを自動的にアーカイブするための構成。管理リカバリ・モードのスタンバイ・データベースでは、プライマリ・データベースから受け取ったログは自動的にスタンバイ・データベースに適用されます。管理スタンバイ環境では、スタンバイ・データベースが管理リカバリ・モードでない場合でも、スタンバイ・サイトは、アーカイブ・ログを受け取り続けることに注意してください。

### 手動リカバリ・モード

次の SQL\*Plus 文を発行することにより開始されるスタンバイ・データベースのモード。

```
RECOVER STANDBY DATABASE;
```

このモードでは、スタンバイ・データベースのリカバリを手動で行うことができます。

### 非管理スタンバイ環境

プライマリ・データベースが、スタンバイ・サイトに自動的に REDO ログをアーカイブしない環境。この環境では、アーカイブ・ログをスタンバイ・サイトに手動で転送し、手動で適用する必要があります。

### プライマリ・データベース

スタンバイ・データベースを作成するために使用されるデータベース。すべてのスタンバイ・データベースは、1つのプライマリ・データベースと関連付けられます。1つのプライマリ・データベースは、複数のスタンバイ・データベースをサポートできます。

### プライマリ・サイト

プライマリ・データベースの場所。プライマリ・サイトとスタンバイ・サイトは別のホスト上にあっても、同じホスト上にあってもかまいません。

### 読取り専用モード

次の SQL 文を発行することにより開始されるスタンバイ・データベースのモード。

```
ALTER DATABASE OPEN READ ONLY;
```

このモードではスタンバイ・データベースの問合せはできますが、データベースの変更はできません。

### スタンバイ・データベース

プライマリ・データベースのバックアップを使用して作成されたデータベースの複製。スタンバイ・データベースには、独自の初期化パラメータ・ファイル、制御ファイルおよびデータ・ファイルがあります。

### スタンバイ・データベース環境

プライマリ・データベースおよびスタンバイ・データベースの物理構成。環境は、次の項目を含む多数の要因に依存します。

- プライマリ・データベースと関連付けられたスタンバイ・データベースの数
- データベースで使用されるマシンの数
- データベースで使用されるマシンのディレクトリ構造
- ネットワーク構成

### スタンバイ・サイト

スタンバイ・データベースの場所。スタンバイ・サイトは、プライマリ・データベースと同じホスト上でも、別のホスト上でもかまいません。

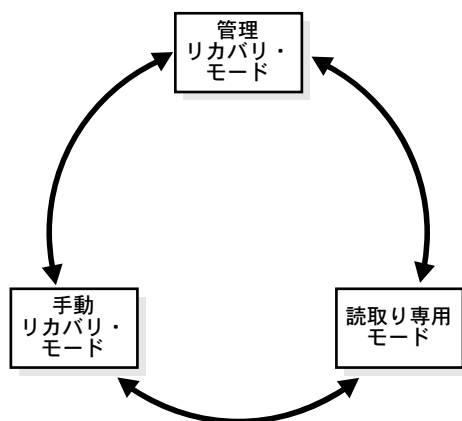
## スタンバイ・データベースのモード

スタンバイ・データベースでは、次のすべての操作を実行できますが、これらの操作を同時に実行することはできません。

- 手動リカバリ・モードのメンテナンス
- 管理リカバリ・モードのメンテナンス
- 問合せのための読取り専用モードでのオープン

スタンバイ・データベースは同時に複数のモードで実行することはできませんが、モード間の切替えは自由に行うことができます。たとえば、[図 1-1](#) で示しているように、管理リカバリ・モードで実行した後、読取り専用でオープンし、その後手動リカバリに切り替えてから、管理リカバリに戻ることができます。

図 1-1 モードの切替え

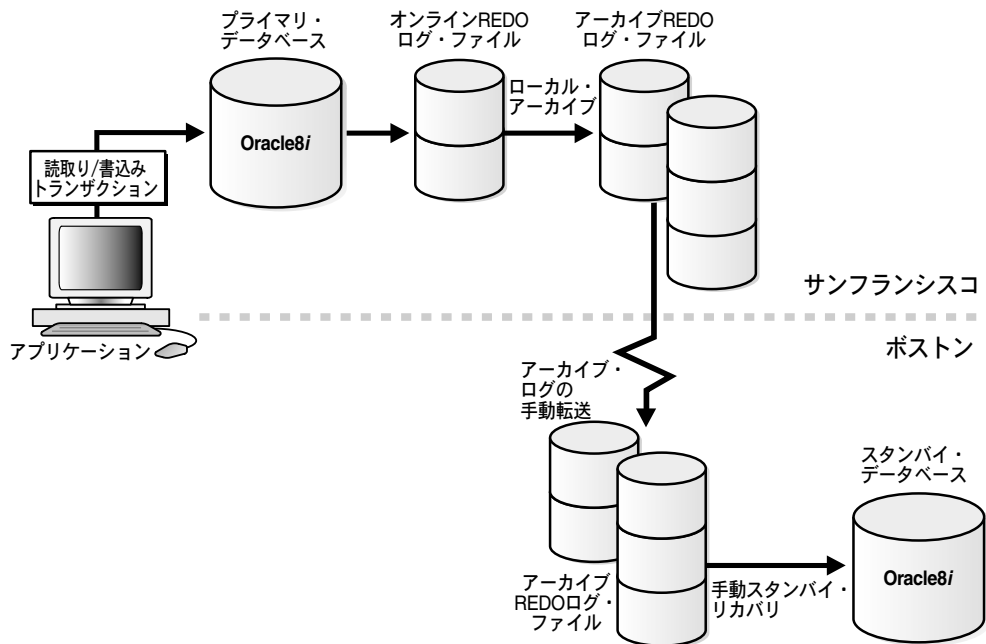


### 手動リカバリ・モード

データベースは手動リカバリ・モードにすることができますが、その場合には、プライマリ・データベースとの同期を保つために、スタンバイ・データベースへのアーカイブ REDO ログの転送および適用を手動で続ける必要があります。

手動でリカバリを実行するには、SQL\*Plus を使用してスタンバイ・データベースに接続し、RECOVER STANDBY DATABASE 文を発行します。[図 1-2](#) は、手動リカバリ・モードのデータベースの例を示しています。

図 1-2 スタンバイ・データベースの手動リカバリ・モード



プライマリ・データベースとスタンバイ・データベースを Net8 経由で接続しない環境では、手動リカバリ・モードが便利です。また、なんらかの原因で、管理リカバリ環境でプライマリ・データベースからスタンバイ・データベースにアーカイブ REDO ログを自動的に転送できない場合には、スタンバイ・データベースを更新するために手動リカバリを実行する必要があります。

**関連項目：** スタンバイ・データベースの手動でのリカバリ方法については、3-3 ページの「[スタンバイ・データベースの手動リカバリ・モード設定](#)」を参照してください。

## 管理リカバリ・モード

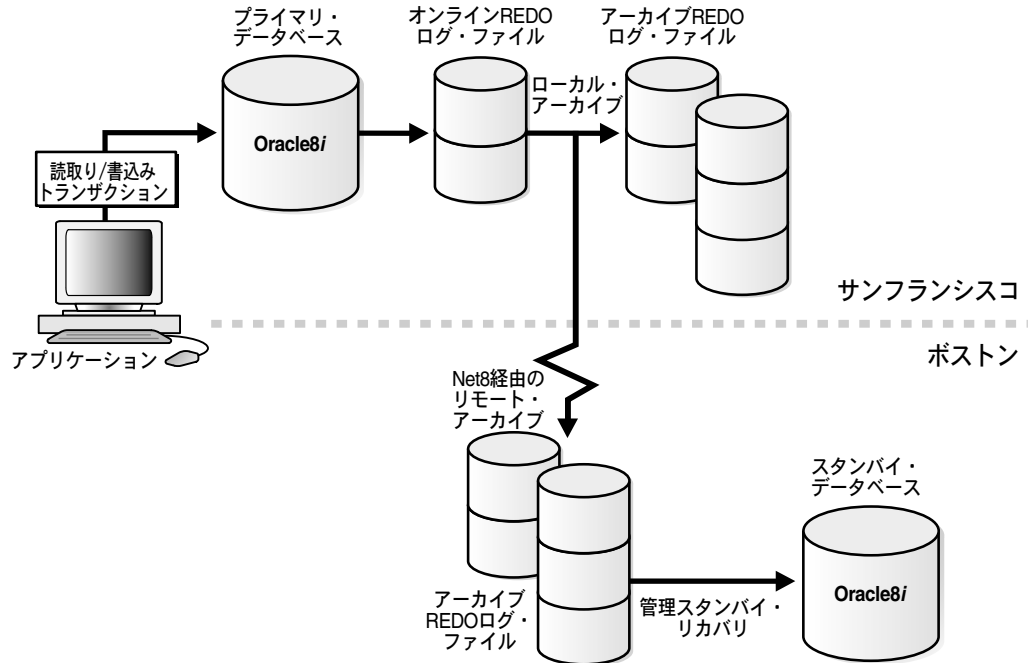
スタンバイ・データベースを管理リカバリ・モードにすると、スタンバイ・データベースはプライマリ・データベースからアーカイブ REDO ログを受け取りしだい、ログを自動的に適用します。管理リカバリを開始するには、SQL\*Plus を使用してスタンバイ・データベースに接続し、RECOVER STANDBY DATABASE 文を発行します。

データベースを管理リカバリ・モードで実行することによる最大の利点は、手動でアーカイブ REDO ログを転送または適用する必要がないことです。Oracle によりこの手順は自動化されています。たとえば、[図 1-3](#) では、サンフランシスコにあるプライマリ・データベース



からボストンにあるスタンバイ・データベースにアーカイブ REDO ログを転送していますが、スタンバイ・データベースではログは自動的に適用されます。

図 1-3 スタンバイ・データベースの自動更新

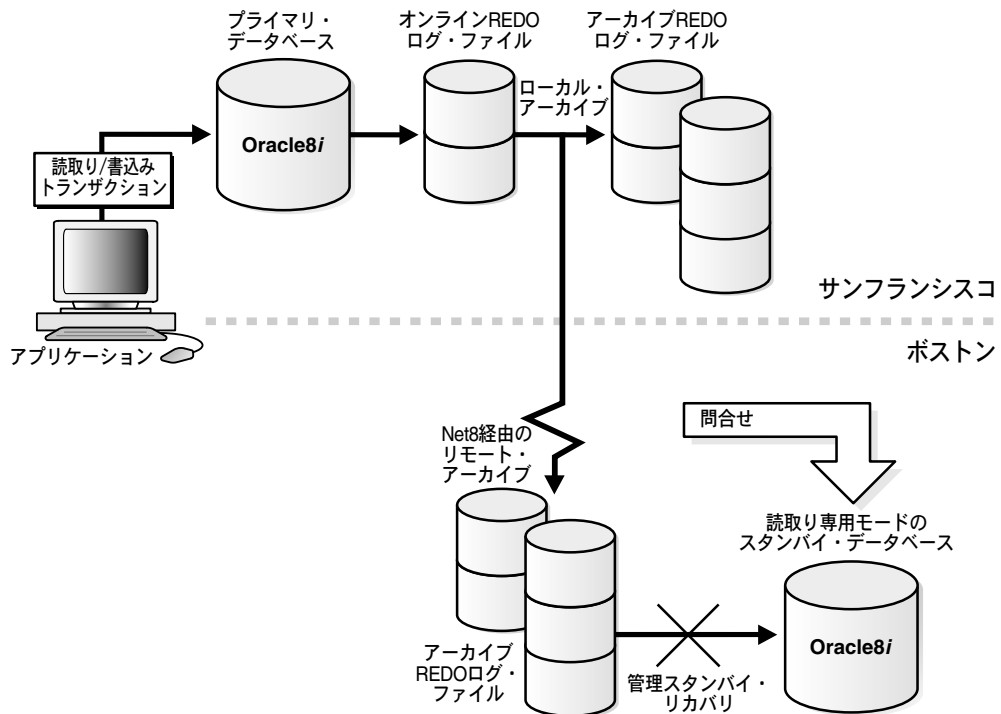


**関連項目：** スタンバイ・データベースを管理リカバリ・モードで実行する方法については、3-13 ページの「[スタンバイ・データベースの管理リカバリ・モード設定](#)」を参照してください。

## 読取り専用モード

手動リカバリまたは管理リカバリの終了後、スタンバイ・データベースを読取り専用モードでオープンすることもできます。その後は、データ・ファイルまたは REDO ログに影響を与えることなく、データベースの間合せ、一時表領域へのデータの格納を行うことができます（スタンバイ・データベース中に存在している場合に限り）。スタンバイ・データベースはシャットダウンせずに、いつでも手動リカバリ・モードまたは管理リカバリ・モードに戻ることができます。図 1-4 は、読取り専用モードのスタンバイ・データベースを示しています。

図 1-4 スタンバイ・データベースの読取り専用モード



管理スタンバイ環境では、スタンバイ・サイトはプライマリ・データベースがアーカイブした REDO ログを継続的に受け取り、制御ファイルはこのレコードによって継続的に更新されます。スタンバイ・データベースは読取り専用モード中ではリカバリを実行しませんが、この間も、スタンバイ・サイトへのアーカイブは続行されます。

プライマリ・データベースへの問合せの数を減らす場合、読取り専用のスタンバイ・データベースを使用すると便利です。たとえば、プライマリ・データベース内の特定の表領域で、変更は頻繁ではないが、アクセスが頻繁に行われる場合には、これらの問合せをスタンバイ・データベースに向けることにより、プライマリ・データベースが読取り要求でオーバーロードになることを防止できます。

**関連項目：** スタンバイ・データベースを読取り専用モードでオープンする方法については、3-16 ページの「[読取り専用モードでスタンバイ・データベースをオープン](#)」を参照してください。

## スタンバイ・データベースへのフェイルオーバー

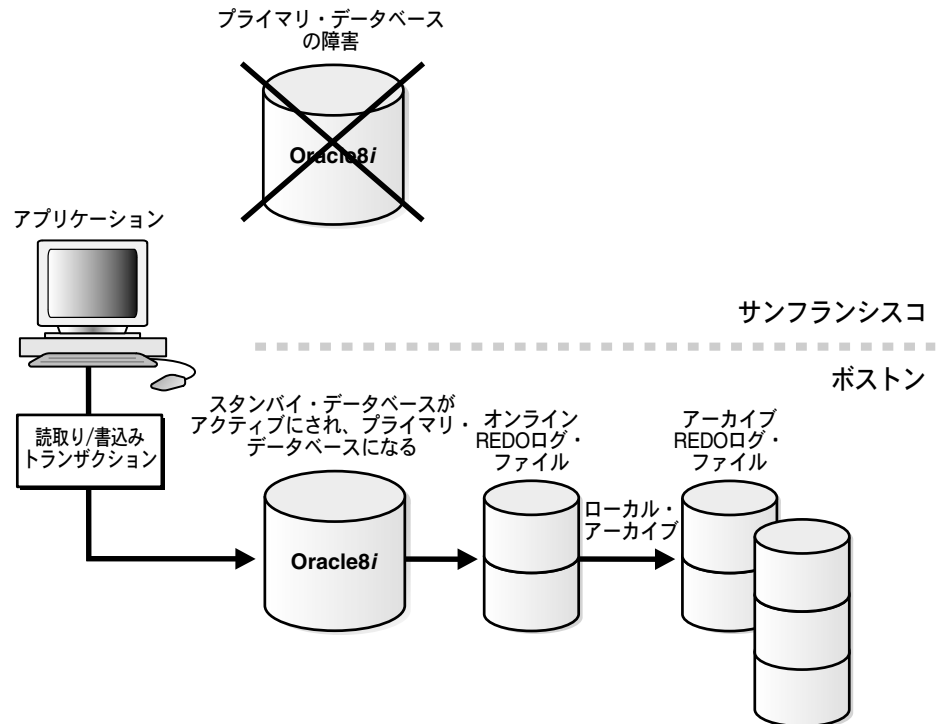
次の SQL 文を発行すると、スタンバイ・データベースへのフェイルオーバー（アクティブ化とも呼ばれる）が発生します。

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

この文は、スタンバイ・データベースがマウントされているときにしか発行できません。

図 1-5 は、サンフランシスコにあるプライマリ・データベースから、ボストンにある管理リカバリ・モードのスタンバイ・データベースへのフェイルオーバーを示しています。

図 1-5 スタンバイ・データベースへのフェイルオーバー



スタンバイ・データベースはアクティブにされると、スタンバイ・データベースではなくなり、完全な機能を備えたプライマリ・データベースになります。この時点で、データベースを読取り / 書き込みモードまたは読取り専用モードでオープンし、通常のように変更を加えたり、問合せを発行できるようになります。

---

---

**注意：** スタンバイ・データベースのアクティブ化は、永久的な操作です。アクティブ化を取り消し、データベースをスタンバイ・データベースとしての以前の役割に戻すことはできません。

---

---

**関連項目：** スタンバイ・データベースへのフェイルオーバーの実行方法については、3-19 ページの「[スタンバイ・データベースのアクティブ化](#)」を参照してください。

## フェイルオーバーの結果

フェイルオーバーにより、スタンバイ・データベースは永久的にプライマリ・データベースに変えられます。スタンバイのアクティブ化は一方方向の操作であるため、新しいプライマリ・データベースは、どのスタンバイ・モードにも戻せません。つまり、フェイルオーバーを実行した後で、この操作を取り消すことはできません。

---

---

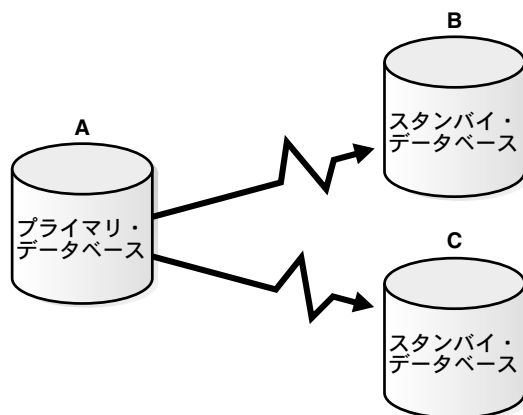
**注意：** スタンバイ・データベースをアクティブにすると、スタンバイ・データベースのオンライン・ログはリセットされます。アクティブ後には、スタンバイ・データベースのアーカイブ・ログとプライマリ・データベースのアーカイブ・ログの互換性はなくなります。

---

---

フェイルオーバーのもう 1 つの結果として、元のプライマリ・データベースをサポートしていた他のスタンバイ・データベースは、新しくアクティブにされたプライマリ・データベースのスタンバイ・データベースとしては無効になります。たとえば、[図 1-6](#) に示すように、プライマリ・データベース A がスタンバイ・データベース B および C をサポートしているとします。

図 1-6 複数のスタンバイ・データベースを持つプライマリ・データベース



A から B へフェイルオーバーを実行すると、C は新しくアクティブにされた B のスタンバイ・データベースとして機能しません。B の REDO ログはリセットされるため、B から C にアーカイブ REDO ログを適用できません。

場合によっては、複数のスタンバイ・データベースを保持することにより、フェイルオーバーの悪影響を減少させることができます。たとえば、プライマリ・データベース A をスタンバイ・データベース B および C がサポートする前述のシナリオを考えてみます。次のようなイベントが発生します。

1. A のマシンにメディア障害が発生します。
2. スタンバイ・データベース B をアクティブにします。ユーザーはプライマリ・データベースとしての B にアクセスするようになります。
3. A のマシンのメディア障害を迅速に修正します。
4. B をシャットダウンし、A を再起動します。
5. ユーザーは再度、プライマリ・データベースとしての A にアクセスするようになります。B は無効にされますが、C は A のスタンバイ・データベースとして機能しつづけます。

このシナリオの場合、B がプライマリ・データベースとして機能していた短期間の間に B で行われた変更は、アーカイブ REDO ログによって適用することはできません。LogMiner ユーティリティを使用してこれらの変更のレポートを生成し、変更を手動でデータベース A に再入力することができます。

## フェイルオーバーを実行しないスタンバイ・データベースのテスト

スタンバイ・データベースへのフェイルオーバーによって、スタンバイとしての機能が失われるため、フェイルオーバー操作は、絶対に必要な場合のみ実行してください。スタンバイ・データベースをテストする場合は、スタンバイ・データベースをアクティブにするのではなく、読取り専用モードでオープンしてください。スタンバイ・データベースを読取り専用モードでオープンすると、スタンバイ・データベースがプライマリ・データベースから受け取った REDO ログを使用してデータ・ファイルを正しく更新していることを確認するために、スタンバイ・データベースを問い合わせることができます。

## フェイルオーバー後の元のプライマリ・データベースの再作成

スタンバイ・データベースをアクティブにした後、フェイルオーバー操作の原因となった、元のプライマリ・サイトの問題を解決すると元のプライマリ・サイトでプライマリ・データベースを再作成することもできます。元のプライマリ・サイトがノード A にあり、アクティブにされたスタンバイ・サイトがノード B にあるものと仮定し、次のステップを実行してください。

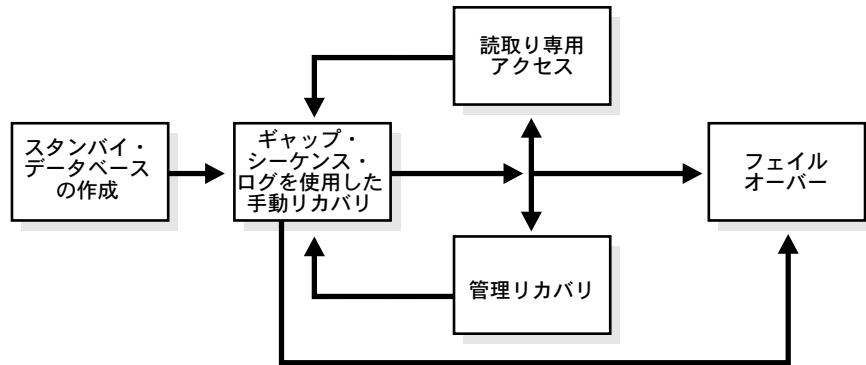
1. ノード B 上のアクティブにされたスタンバイ・データベースの一貫性バックアップを作成します。
2. ノード B で作成したバックアップをノード A にリストアします。
3. ノード B 上のアクティブにされたスタンバイ・データベースをシャットダウンします。
4. A のリストアしたデータベースをオープンします。これがプライマリ・データベースになります。
5. ノード A 上のデータベースのバックアップを作成します。
6. A のバックアップを使用し、ノード B 上にスタンバイ・データベースを再作成します。

**関連項目：** スタンバイ・データベースの再作成の方法については、5-43 ページの「[シナリオ 8: スタンバイ・データベースの再作成](#)」を参照してください。

## スタンバイ・データベースのライフ・サイクル

スタンバイ・データベースを実装する場合、ほとんどの場合は管理リカバリが使用されています。管理リカバリを想定したスタンバイ・データベースのライフ・サイクルを図 1-7 に示します。

図 1-7 スタンバイ・データベースのライフ・サイクル



この図では 4 つの段階を示しています。

1. スタンバイ・データベースの作成
2. ギャップ・シーケンス内のログを使用した手動リカバリ
3. 管理リカバリおよび読取り専用のアクセス・サイクル
4. スタンバイ・データベースへのフェイルオーバー

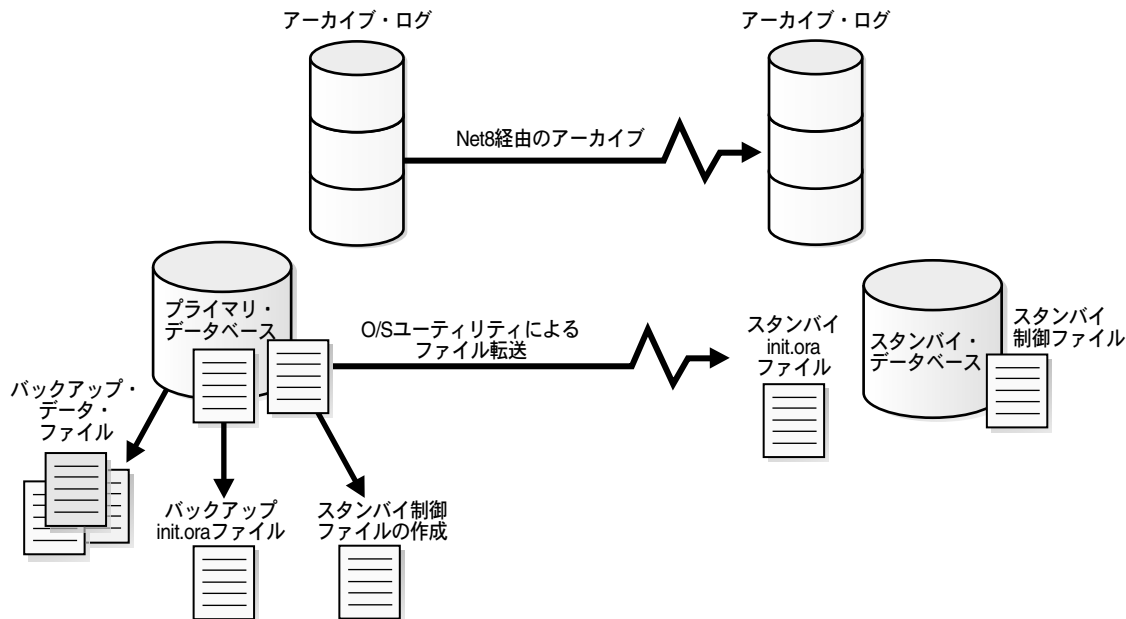
## スタンバイ・データベースの作成

この段階では、プライマリ・データベースの制御ファイルおよびデータ・ファイルのバックアップからスタンバイ・データベースを構築し、管理リカバリの準備をする必要があります。図 1-8 に示すように、この段階では次のような基本的作業を行います。

1. プライマリのデータ・ファイルのバックアップを作成（または以前のバックアップにアクセス）し、スタンバイの制御ファイルを作成します。
2. スタンバイのデータ・ファイルおよび制御ファイルをスタンバイ・サイトに転送します。
3. スタンバイ・サービス名に接続できるように Net8 を構成します。
4. プライマリおよびスタンバイの初期化パラメータ・ファイルを構成します。
5. プライマリ・サイトで自動アーカイブを開始します。

6. マウントせずにスタンバイ・インスタンスを起動します。

図 1-8 スタンバイ・データベースの作成



**関連項目：** この手順は、[第2章](#)で詳しく説明します。追加情報は、5-2 ページの「[シナリオ 1: 同一ホスト上のスタンバイ・データベースの作成](#)」および 5-14 ページの「[シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成](#)」を参照してください。

## ギャップ・シーケンス内のログを使用した手動リカバリ

プライマリ・データベースが生成した次のアーカイブ REDO ログをスタンバイ・データベースに適用できない場合には、必ずギャップ・シーケンスが作成されます。たとえば、プライマリ・データベースがスタンバイ・サイトにログ 100 をアーカイブしたとき、スタンバイの制御ファイルはプライマリ・データベースがアーカイブした最新のログが 89 であったときに作成されたものであるため、スタンバイの制御ファイルにはログ順序 89 以降のログに関する情報がありません。この場合のギャップ・シーケンスはログ 90 から 99 になります。



管理リカバリを開始するためには、最初にギャップ・シーケンス内のログを手動でスタンバイ・データベースに適用する必要があります。この手動リカバリを実行した後、RECOVER MANAGED STANDBY DATABASE 文を発行すると、後続のログは Oracle によって自動的にスタンバイ・データベースに適用されます。

## ギャップ・シーケンスの典型的な原因

ギャップ・シーケンスは一般的には次のような状況で発生します。

- スタンバイ・データベースの作成
- プライマリ・データベースのオープン中のスタンバイ・データベースのシャットダウン
- スタンバイ・サイトへのアーカイブを阻害するネットワーク障害

最初の2つの状況では、プライマリ・データベースはスタンバイ・サイトに REDO ログをアーカイブできますが、スタンバイ・データベースの制御ファイルは、マウントされていない間にアーカイブされたログを認識しません。プライマリ・データベースがプライマリ・サイトへのアーカイブを行ったが、スタンバイの制御ファイルがデータベースのリカバリに必要なログのレコードを含まない状況では、必ずギャップ・シーケンスが作成されます。

3番目の状況では、プライマリ・データベースはローカルのアーカイブを続行しますが、ネットワーク障害により、スタンバイ・サイトへのアーカイブはできません。アーカイブ・ログはプライマリ・サイトに蓄積されますが、スタンバイの制御ファイルはこれらのログを認識しません。このため、管理リカバリを開始する前に、蓄積されたログを手動で転送し、手動リカバリ操作でログを適用する必要があります。

## ギャップ・シーケンス・サイクル

プライマリ・データベースが、スタンバイの制御ファイルが認識しないログをアーカイブするたびにギャップ・シーケンスが発生するため、ギャップ・シーケンス・サイクルが発生することがあります。このサイクルは、ギャップ・シーケンス内のログを使用して手動リカバリを実行するために、管理リカバリ・モードまたは読取り専用モードのいずれかを終了する必要があります。順番にすべてのログの手動リカバリが完了した後、管理リカバリ・モードまたは読取り専用モードに戻ることができます。

**関連項目：** ギャップ・シーケンスの問題を解決するための詳しい手順は、3-5 ページの「[管理リカバリ開始前のギャップ・シーケンスの解決](#)」で説明しています。第5章では、ギャップ・シーケンスを解決するための手順を、様々なシナリオで説明しています。

## 管理リカバリおよび読取り専用のアクセス・サイクル

ほとんどの場合、データベースは主として管理リカバリ・モードで実行するか、読取り専用モードで実行することになります。管理リカバリ時には、スタンバイ・サイトはプライマリ・データベースからログを受け取り、ログはスタンバイのリカバリ処理によって自動的に適用されます。読取り専用モードでは、スタンバイ・サイトはプライマリ・データベースからログを受け取りますが、スタンバイのリカバリ処理によるログの適用はありません。

管理リカバリ・モードと読取り専用モードの切替えは簡単に行うことができます。読取り専用モードでは、受け取ったログによりスタンバイ制御ファイルの更新が続けられるため、管理リカバリ・モードに戻る前に手動リカバリを実行する必要はありません。

**関連項目：** 管理リカバリを実行するための詳細な手順は 3-13 ページの「[スタンバイ・データベースの管理リカバリ・モード設定](#)」で説明しています。読取り専用モードでスタンバイ・データベースをオープンするための詳細な手順は 3-16 ページの「[読取り専用モードでスタンバイ・データベースをオープン](#)」で説明しています。

## スタンバイ・データベースへのフェイルオーバー

データベースがマウントされている限り、スタンバイ・データベースはいつでもアクティブにできます。一般的には、フェイルオーバー操作は、スタンバイ・データベースが管理リカバリ・モードで実行されているときに実行されます。しかし、ギャップ・シーケンス内のログを使用して手動リカバリを実行しているときにプライマリ・データベースに障害が発生したときにも、フェイルオーバーを実行することができます。この場合には、プライマリ・データベースで行われた最新の変更がスタンバイ・データベースに含まれていなくても、スタンバイ・データベースをアクティブにできます。

フェイルオーバー後に、新しいプライマリ・データベースをバックアップし、新しいスタンバイ・データベースを作成することにより、スタンバイ・ライフ・サイクルを再び開始できます。

**関連項目：** フェイルオーバーを実行するための詳細な手順は 3-19 ページの「[スタンバイ・データベースのアクティブ化](#)」で説明しています。

## スタンバイ・データベース環境の構成

スタンバイ・データベース環境とは、1つのプライマリ・データベースと、1つまたは複数の関連するスタンバイ・データベースの物理構成を意味します。この項では、スタンバイ・データベース環境の構成に影響を与える主な要因を説明します。

- [スタンバイ・データベースの数](#)
- [スタンバイ・サイトへのアーカイブ REDO ログの転送方法](#)
- [プライマリ・サイトおよびスタンバイ・サイトの場所およびディレクトリ構造](#)

## スタンバイ・データベースの数

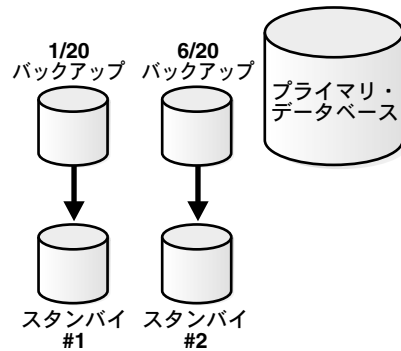
1つのスタンバイ・データベースは1つのプライマリ・データベースとしか同期させることはできませんが、1つのプライマリ・データベースがサポートできるスタンバイ・データベースの数には理論上は制限はありません。これらのスタンバイ・データベースは別個の独立した存在であり、複数のマシン上に置くことも、単一のマシン上に置くことも可能です。

## バックアップを使用したスタンバイの作成

各スタンバイ・データベースは、プライマリ・データベースのバックアップから作成する必要があります。一貫性バックアップまたは非一貫性バックアップ、オープンまたはクローズのバックアップを使用できます。

また、プライマリ・データベースの1つのバックアップを使用して、無数のスタンバイ・データベースを作成することもできますが、環境内の各種のスタンバイ・データベースを同じバックアップから作成する必要はありません。たとえば、1月20日に作成したプライマリ・データベースのバックアップから1つのスタンバイ・データベースを作成し、6月20日に作成したバックアップから別のスタンバイ・データベースを作成できます（[図 1-9](#)を参照）。バックアップの完全リカバリに必要なアーカイブ REDO ログがあれば、スタンバイ・データベースのベースとして使用できます。

図 1-9 異なるバックアップを使用したスタンバイ・データベースの作成



## スタンバイ・サイトへの自動アーカイブによる制限

管理スタンバイ環境では、プライマリ・データベースは最大4つのスタンバイ・サイトに自動的にアーカイブできます。このため、スタンバイ・データベース環境では、最大4つのスタンバイ・データベースを管理リカバリ・モードで同時に実行できるようになります。

自動アーカイブにより、管理リカバリ・モードではメンテナンスできるスタンバイ・データベースの数が制限を受けますが、手動リカバリ・モードではメンテナンスできるスタンバイ・データベースの数の制限はありません。たとえば、4箇所のスタンバイ・サイトに自動的にアーカイブするようにプライマリ・データベースを構成した後、オペレーティング・システム・コマンドを使用し、追加で5箇所のスタンバイ・サイトにアーカイブ REDO ログをコピーできます。このため、特定のプライマリ・データベースをサポートできるスタンバイ・データベースの数には理論上の制限はありません。

## スタンバイ・サイトへのアーカイブ REDO ログの転送方法

スタンバイ・データベース環境の重要な局面の1つとして、プライマリ・サイトからスタンバイ・サイトへのアーカイブ REDO ログの転送があります。ログを転送するには2つの方法があります。

- スタンバイ・サイトに自動的にアーカイブするようにプライマリ・データベースを構成する方法
- ログを手動で転送する方法

スタンバイ・サイトに自動的にアーカイブするのは、管理リカバリ環境で運用している場合です。

### 管理リカバリ環境

管理リカバリ環境をインプリメントする場合には、プライマリ・データベースとスタンバイ・データベースを Net8 で接続する必要があります。スタンバイ・サイトに自動アーカイブを行うようにプライマリの初期化パラメータ・ファイルを構成するには、サービス名を指定する必要があります。サービス名を指定するには、tnsnames.ora、listener.ora および names.ora などのネットワーク・ファイルを構成する必要があります。

プライマリ・データベースからスタンバイ・サービスにアーカイブする場合、Oracle はスタンバイ・サイト上のディレクトリに、Net8 を通じてアーカイブ REDO ログを自動的に転送します。プライマリ・データベースで各ログがアーカイブされるたびに、Oracle は新しいログを自動的にスタンバイ・サイトに転送します。

**自動アーカイブおよび管理リカバリの独立性** プライマリ・データベースは、スタンバイ・データベースがスタンバイ・リカバリ・モードでない場合でもスタンバイ・サイトへのアーカイブを継続できますが、スタンバイ・インスタンスが起動している場合のみ可能です。スタンバイ・データベースのリカバリのメカニズムは、スタンバイ・サイトへのアーカイブ REDO ログの自動転送のメカニズムとは別のものです。このため、スタンバイ・データベースの管理リカバリ・モードを終了し、一時的に読取り専用モードにすることが可能になります。スタンバイ・データベースが読取り専用モードの間、アーカイブ REDO ログはスタンバイ・サイトに蓄積されます。

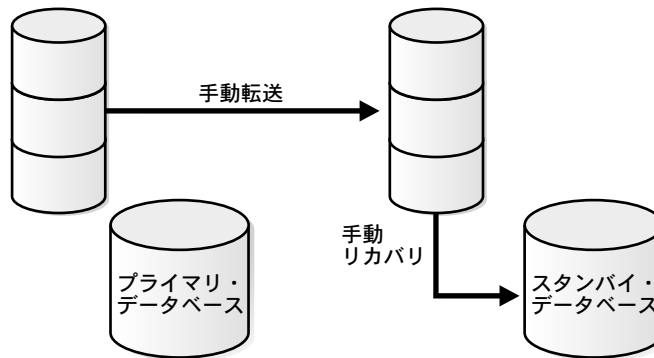
**アーカイブ REDO ログのオプションの手動転送** スタンバイ・サイトに自動的にアーカイブするようにプライマリ・データベースを構成した場合でも、必要に応じて手動でログを転送することができますが、スタンバイ・サイトをオプションの宛先として指定してある場合のみ可能です (2-17 ページの「**必須およびオプションのアーカイブ先の指定**」を参照)。

たとえば、Net8 構成の問題により、スタンバイ・サイトにアーカイブ・ログを転送できないとします。プライマリ・データベースはローカルにアーカイブを継続していますので、オペレーティング・システム・コマンドを使用して手動でログを転送後、スタンバイ・サイトで手動リカバリを実行し、スタンバイ・データベースを同期させることができます。

## 非管理リカバリ環境

非管理リカバリ環境では、スタンバイ・サイトに自動的にアーカイブするようにプライマリ・データベースを構成することはないので、アーカイブ REDO ログを手動で転送する必要があります。プライマリ・データベースはスタンバイ・サイトに自動的にアーカイブすることはないため、Net8 接続は必要ありません。たとえば、UNIX の **cp** または **ftp** などのオペレーティング・システムのユーティリティを使用し、スタンバイ・サイトにアーカイブ REDO ログを転送し、オペレーティング・システムの認証を使用してスタンバイ・データベースに接続した後、手動でデータベースのリカバリを行うことができます（図 1-10 を参照）。

図 1-10 非管理リカバリ環境



**注意：** 管理リカバリ環境と非管理リカバリ環境を同時に使用できないことはありません。たとえば、Net8 経由で 1 つのスタンバイ・サイトに自動的にアーカイブするようにプライマリ・データベースを構成しながら、ネットワーク化していない別のスタンバイ・サイトに、プライマリ・サイトから手動でアーカイブ・ログを転送できます。

## プライマリ・サイトおよびスタンバイ・サイトの場所およびディレクトリ構造

スタンバイ・データベース環境の重要な局面の 1 つに、使用するマシンの数および構成があります。次の事項は特に重要です。

- プライマリ・データベースとスタンバイ・データベースは同じホストに置くか、別々のホストに置くかどうか
- プライマリ・サイトとスタンバイ・サイトで同じディレクトリ構造を使用するか、別のディレクトリ構造を使用するかどうか

### マシンの数と場所

特定のプライマリ・データベースに関連付けるスタンバイ・データベースの数には理論上の制限はないのと同様に、スタンバイ・データベースを配置するマシンの数にも理論上の制限はありません。たとえば、スタンバイ・データベースは次のように配置できます。

- プライマリ・データベースと同じホスト上
- 同じデータ・センターの別のホスト上
- 同じ地域にある別のデータ・センターの別のホスト上
- 別の大陸など、地域的に離れた場所にある別のデータ・センターの別のホスト上

スタンバイ・データベース環境でのホストの場所は、災害リカバリ計画に大きな影響を与えます。たとえば、1つのデータ・センターにあるプライマリ・ホストが破壊された場合、スタンバイ・データベースが同じデータ・センターまたは別のデータ・センターにある別のホスト上に置かれていない限り、スタンバイ・データベースへのフェイルオーバーを実行することはできません。最悪の場合、データ・センターが完全に破壊されたときには、スタンバイ・データベースが離れた場所の別のマシンに置かれていない限り、スタンバイ・データベースへのフェイルオーバーを実行することはできません。

**関連項目：** 5-31 ページの「シナリオ 5: 複数スタンバイ・データベース構成でのフェイルオーバー先のスタンバイ・データベースの決定」では、プライマリ・データベースおよびスタンバイ・データベースの場所について説明しています。

### スタンバイ・サイトのディレクトリ構造

スタンバイ・サイトのディレクトリ構造は、スタンバイのデータ・ファイルおよび REDO ログのパス名を決定するものであるため重要です。プライマリ・データベースと同じホスト上にあるスタンバイ・サイトでは、別のディレクトリ構造を使用しないと、スタンバイ・データベースによりプライマリ・データベースのファイルが上書きされることになります。

一般的には、スタンバイのファイルを可能な限り同じパス名で使用してください。この方法により、ファイル名変換パラメータを設定したり、スタンバイのファイル名を手動で変更する必要がなくなります。しかし、別のディレクトリ構造を持つホストを使用したり、スタンバイ・データベースとプライマリ・データベースを同じホスト上に配置する必要がある場合には、最小限の追加の管理作業で可能です。

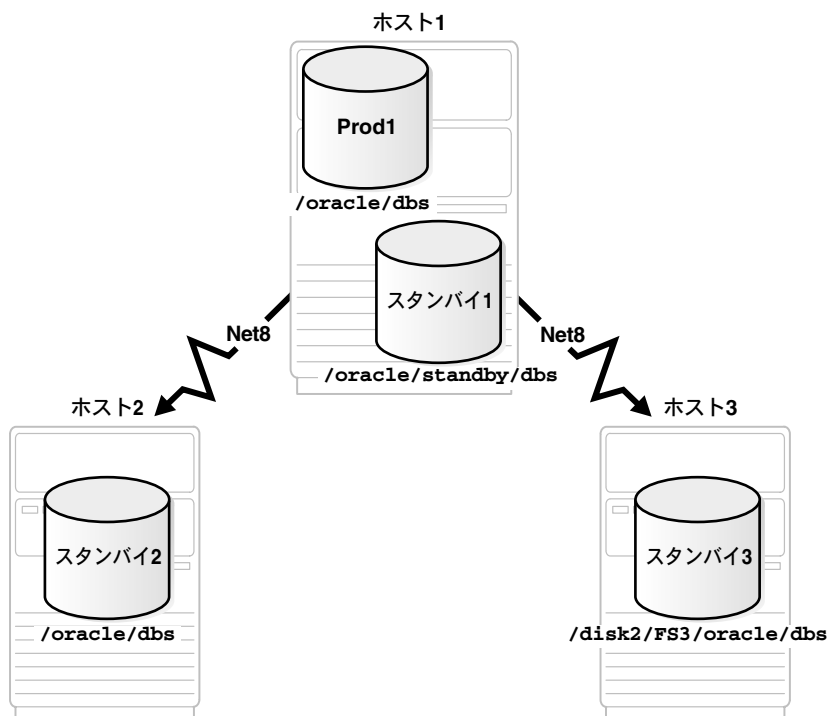
### 構成オプション

[図 1-11](#) に示すように、3つの基本的な構成オプションが提供されています。

- プライマリ・データベースと同じホスト上のスタンバイ・データベース
- プライマリ・ホストと同じディレクトリ構造を使用する、別のホスト上のスタンバイ・データベース

- プライマリ・ホストとは異なるディレクトリ構造を使用する、別のホスト上のスタンバイ・データベース

図 1-11 考えられるスタンバイ構成



次の表は、プライマリ・データベースおよびスタンバイ・データベースの考えられる構成と、それぞれの構成の結果を示しています。

スタンバイ・ホスト	ディレクトリ構造	結果
プライマリ・ホストと同じホスト	(必然的に)プライマリ・ホストと異なる	<ul style="list-style-type: none"><li>■ LOCK_NAME_SPACE パラメータを設定する必要がある。</li><li>■ スタンバイ・データベースの制御ファイル内のプライマリ・データベースのファイル名を改名する必要がある (2-21 ページの「<a href="#">スタンバイの制御ファイル内のプライマリのファイル名の改名</a>」を参照)。</li><li>■ オペレーティング・システムの中には、同じマシン上で、同じ名前前の 2 つのインスタンスを実行することを許可しないものがある。詳細は、プラットフォーム別のドキュメントを参照。</li><li>■ スタンバイ・データベースは災害に対する保護にならない。</li></ul>
プライマリ・ホストと別のホスト	プライマリ・ホストと同じ	<ul style="list-style-type: none"><li>■ スタンバイの制御ファイル内のプライマリ・データベースのファイル名を改名する必要はないが、新しいネーミング・スキームが必要な場合は改名できる (たとえば、複数のディスクにファイルを分散する)。</li><li>■ データベースに別の物理メディアを使用すれば、プライマリ・データの保護対策になる。</li></ul>
プライマリ・ホストと別のホスト	プライマリ・ホストと異なる	<ul style="list-style-type: none"><li>■ スタンバイ・データベースの制御ファイル内のプライマリ・データベースのファイル名を改名する必要がある (2-21 ページの「<a href="#">スタンバイの制御ファイル内のプライマリのファイル名の改名</a>」を参照)。</li><li>■ データベースに別の物理メディアを使用すれば、プライマリ・データの保護対策になる。</li></ul>

これらの構成は相互排他ですが (つまり特定のスタンバイ・データベースでは 1 つの構成しか使用できません)、特定のプライマリ・データベースに対して、複数のスタンバイ・データベースを同時に実行できます。このため、これらの構成を自由に組み合わせてインプリメントできます。たとえば、プライマリ・データベースと同じホスト上に 1 つのスタンバイ・データベースを配置し、同じデータ・センター内の別のホスト上に別のスタンバイ・データベースを配置し、地球の反対側にある別のホスト上に 3 番目のスタンバイ・データベースを配置できます。各スタンバイは、管理リカバリ・モード、手動リカバリ・モードまたは読取り専用モードで実行できます。

複数のスタンバイ・データベースをメンテナンスすれば、災害に対する保護になるのみでなく、スタンバイ・データベースの管理レベルの柔軟性が増すことになります。たとえば、ハードウェアのアップグレードのために 1 つのスタンバイ・データベースをシャットダウンしても、他のスタンバイ・データベースが稼動しているため、災害に対する保護が維持されます。また、1 つのスタンバイ・データベースへのアーカイブ REDO ログの適用を遅延させ、他のスタンバイ・データベースへの適用を遅延させなければ、プライマリ・データベー



スが破損した場合に、まだ破損していないスタンバイ・データベースをアクティブにすることができます。

スタンバイ・データベースは、誤って表を切り捨てるなどのユーザー・エラーに対する優れた保護策を提供します。スタンバイ・データベースへのアーカイブ REDO ログの適用を遅延させれば、スタンバイ・データベースを読み取り専用モードでオープンし、紛失したデータをエクスポートできます。その後、紛失したデータをプライマリ・データベースにインポートして戻すことができます。正しいデータをプライマリ・データベースに入れた後、スタンバイ・データベースのリカバリを続行できます。この手順では、ユーザー・エラーと、ユーザー・エラーの訂正の両方が進行するため、スタンバイ・データベースをアクティブにする必要はありません。

## スタンバイ・データベースのメンテナンス

スタンバイ環境の管理は大部分が自動化されていますが、完全に自動化されているわけではありません。様々な状況で、スタンバイ・データベースの状態チェックや、メンテナンスを実行する必要があります。最も一般的なメンテナンス操作は次のとおりです。

- [アーカイブ REDO ログの状態のチェック](#)
- [スタンバイ・データベースのバックアップ](#)
- [プライマリ・データベースの物理的変更の反映](#)

### アーカイブ REDO ログの状態のチェック

スタンバイ・データベースが正しく機能していることを確認するために、どのログがスタンバイ・サイトにアーカイブされ、どのログがスタンバイ・データベースに適用されているかを検査する必要があります。次の方法を使用できます。

- スタンバイ・データベースで V\$LOG\_HISTORY ビューを使用し、適用された一番新しいログを確認します。
- スタンバイの初期化パラメータ・ファイル内の LOG\_ARCHIVE\_TRACE パラメータを使用して作成したスタンバイ・トレース・ファイルを分析し、スタンバイ・サイトで受け取ったログについての情報を入手します。

**関連項目：** アーカイブ・ログの動作の分析の詳細は、2-33 ページの「[スタンバイ・データベース構成のトラブルシューティング](#)」を参照してください。

## スタンバイ・データベースのバックアップ

必要であれば、スタンバイ・データベースをバックアップできますが、データベースが手動リカバリ・モードまたは管理リカバリ・モードにある間はバックアップできません。スタンバイ・データベースの管理リカバリ・モードを終了し、バックアップを作成してから、管理リカバリを再開する必要があります。データベースがシャットダウンしているとき、または読取り専用モードのときにはバックアップを作成できます。

**関連項目：** バックアップ作成の詳細は、4-16 ページの「[スタンバイ・データベースのバックアップ](#)」を参照してください。

## プライマリ・データベースの物理的変更の反映

プライマリ・データベース構造の変更は、必ずスタンバイ・データベースに影響を与えます。次のような場合には、スタンバイ・データベースは適用された REDO により、自動的に更新されます。

- プライマリの表領域をオンラインまたはオフラインにした場合
- 表領域の状態を読取り / 書込みおよび読取り専用から変更した場合
- データ・ファイルを改名した場合
- 表領域を削除した場合

しかし、次のような状況では、スタンバイ・データベースのメンテナンスを実行する必要があります。

- プライマリ・データベースにデータ・ファイルを追加した場合
- プライマリ・データベースに表領域を作成した場合

次のような状況では、スタンバイ・データベース全体を再作成する必要があります。

- プライマリ・データベースのバックアップ制御ファイルをリストアした場合
- RESETLOGS オプションを使用してプライマリ・データベースをオープンした場合

**関連項目：** プライマリ・データベースの構造変更に関連したメンテナンスの詳細は、4-10 ページの「[スタンバイ・データベースに影響を与えるイベントへの応答](#)」を参照してください。プライマリ・データベースで物理的変更が行われた場合に必要な手順の説明は、5-21 ページの「[シナリオ 3: プライマリ・データベースの物理的変更への対応](#)」を参照してください。

# スタンバイ・データベースの文

スタンバイ・データベースに対する操作を指定するために、いくつかの異なる SQL および SQL\*Plus 文ではキーワード STANDBY が使用されます。スタンバイ独自の構文はありませんが、スタンバイ・データベースに対する操作の実行に便利なその他の SQL 文もあります。表 1-1 には、関連する文を記載します。

表 1-1 スタンバイ・データベースの文

文	説明
ALTER DATABASE ACTIVATE STANDBY DATABASE	スタンバイ・データベースをアクティブにすることにより、データベースのスタンバイとしての機能を完全に停止する。3-19 ページの「スタンバイ・データベースのアクティブ化」を参照。
ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename'	スタンバイの制御ファイルを作成する。プライマリ・データベースでこの文を発行する。2-8 ページの「スタンバイの制御ファイルの作成」を参照。
ALTER DATABASE MOUNT STANDBY DATABASE	スタンバイ・データベースをマウントする。
ALTER DATABASE OPEN READ ONLY	読取り専用モードでスタンバイ・データベースをオープンする。3-16 ページの「読取り専用モードでスタンバイ・データベースをオープン」を参照。
RECOVER [FROM '/dir'] STANDBY DATABASE	スタンバイ・データベースの手動リカバリを開始する。デフォルト以外のディレクトリで Oracle にアーカイブ REDO ログを検索させることができる。3-3 ページの「スタンバイ・データベースの手動リカバリ・モード設定」を参照。
RECOVER MANAGED STANDBY DATABASE [TIMEOUT integer]	スタンバイ・データベースの管理リカバリを開始する。TIMEOUT オプションを指定すると、Oracle はログの到着を integer 分の間待った後、リカバリを終了する。3-13 ページの「スタンバイ・データベースの管理リカバリ・モード設定」を参照。
RECOVER MANAGED STANDBY DATABASE CANCEL	Oracle は、管理リカバリが現行のログの書込みを終了するまで待ち、リカバリを終了する。3-15 ページの「管理リカバリの取消し」を参照。

表 1-1 ( 続き ) スタンバイ・データベースの文

文	説明
RECOVER MANAGED STANDBY DATABASE CANCEL IMMEDIATE	REDO ログからの別のブロックの読取り、または次の REDO ログ・ファイルのオープンのいずれか先に発生した事象の前に、管理リカバリを終了する。3-15 ページの「 <a href="#">管理リカバリの取消し</a> 」を参照。
STARTUP NOMOUNT pfile='initSID.ora'	制御ファイルをマウントせずにスタンバイ・インスタンスを起動する。スタンバイ・データベースをマウントする前にこの文を実行する必要がある。

---

## スタンバイ・データベースの準備

この章では、手動リカバリまたは管理リカバリを使用するためのスタンバイ・データベースの準備作業について説明します。この章は次のトピックから構成されています。

- スタンバイ・データベース作成前の環境評価
- スタンバイ・データベースのセットアップ: 基本作業
- スタンバイ・データベース・ファイルの作成
- プライマリおよびスタンバイ・データベース用のネットワーク・ファイルの構成
- プライマリ・データベースの初期化パラメータ・ファイルの構成
- スタンバイ・データベースの初期化パラメータ・ファイルの構成
- パラメータおよびネットワーク・ファイルの構成: シナリオ
- リカバリの準備としてのスタンバイ・インスタンスの起動
- スタンバイ・データベース構成のトラブルシューティング

## スタンバイ・データベース作成前の環境評価

スタンバイ・データベースをインプリメントするかどうかを決定した後、スタンバイ・データベースを使用する環境に関して、いくつかの重要な問題を考慮しておく必要があります。これらの質問に対する解答は、データの損失または破壊に対する保護策として使用するのか、または補足的レポートに使用するのかという、スタンバイ・データベースの目的によって異なります。

- [スタンバイ・データベースの更新方法](#)
- [スタンバイ・データベースの別のホストへの配置](#)
- [プライマリ・データベースの中でのオンライン REDO ログの構成方法](#)
- [フェイルオーバー後のスタンバイ・データベースの接続方法](#)

## スタンバイ・データベースの更新方法

最初の質問は、スタンバイ・データベースを管理スタンバイ環境で実行するかどうかということです (1-2 ページの「[構成オプション](#)」を参照)。この環境では、プライマリ・データベースはスタンバイ・インスタンスが起動されている限り、スタンバイ・サイトに REDO ログを自動的にアーカイブします。

管理スタンバイ環境をインプリメントするかどうかは、状況によって判断してください。判断の助けとして次の表を使用してください。

目的	スタンバイ環境
災害時の保護	管理
データ破損に対する保護	時間差を付けた管理または非管理
頻繁に変更されることのないデータ・ウェアハウスまたはデータベースの補足的レポート	管理または非管理
頻繁に変更されるデータ・ウェアハウスまたはデータベースの補足的レポート	管理

管理スタンバイ環境を選択した場合には、スタンバイ・データベースを管理リカバリ・モードで実行し、アーカイブ REDO ログを自動的に適用させることができます。データベースをレポート用に使用する場合でも、データ損失に対する保護策として使用する場合でも、このオプションではユーザーの介入は最小限ですみます。非管理環境を選択した場合には、ログを手動で転送し、スタンバイ・データベースの手動リカバリを実行する必要があります。

## スタンバイ・データベースの別のホストへの配置

次の理由により、スタンバイ・データベースは可能な限り、常に別のホストに配置することをお薦めします。

- 別のホストに配置しないと、スタンバイ・データベースは災害時の保護になりません。
- スタンバイ・データベースをプライマリ・データベースと同じホストに配置すると、システムのオーバーヘッドが発生します。
- 両方のホストで同じディレクトリ構造を使用できるため、スタンバイの制御ファイル内でデータ・ファイルおよびオンライン REDO ログを改名する必要がありません。プライマリ・サイトとスタンバイ・サイトで異なるディレクトリ構造を使用した場合は、ファイル名の変換が必要になります。

**関連項目：** スタンバイの制御ファイル内のプライマリ・ファイルの改名の詳細は、2-21 ページの「[スタンバイの制御ファイル内のプライマリファイル名の改名](#)」および 2-31 ページの「[変換パラメータで取り込めないスタンバイのファイルの手動での改名](#)」を参照してください。

## プライマリ・データベースの中でのオンライン REDO ログの構成方法

オンライン REDO ログのサイズおよびスイッチ頻度の両方が、プライマリ・サイトでのアーカイブ REDO ログの生成に影響を与えます。表 2-1 は、オンライン REDO ログの様々な構成が与える影響を示しています。

表 2-1 オンライン REDO ログの構成の影響

オンライン REDO ログの状態	影響
小	<p>ログ・スイッチがより頻繁になる。管理スタンバイ環境の場合、スイッチが頻繁になると、次のことが起こります。</p> <ul style="list-style-type: none"><li>■ 連続的なログ転送によりネットワーク通信量が増加する。</li><li>■ ギャップ・シーケンスのために管理リカバリを開始できないため、スタンバイ・データベースの手動リカバリという問題が発生する。ギャップ・シーケンスの説明は、3-5 ページの「<a href="#">管理リカバリ開始前のギャップ・シーケンスの解決</a>」を参照してください。</li><li>■ 各アーカイブ REDO ログに含まれるトランザクション数が減少するため、データ損失に対する保護が向上する。</li></ul>
大	<p>REDO ログのスイッチ頻度は減るが、各ファイルに格納されるトランザクション数が増える。この状況では、次のことが起こります。</p> <ul style="list-style-type: none"><li>■ 1 つのログの転送時間が長くなるため、ネットワーク・ボトルネックの発生する可能性が高くなる。</li><li>■ REDO ログをアーカイブする前にプライマリ・データベースで障害が発生した場合、失われる可能性のあるデータの量が増える。</li></ul>

表 2-1 ( 続き ) オンライン REDO ログの構成の影響

オンライン REDO ログの状態	影響
多数	プライマリ・データベースは、2 つの最小限の REDO ログをメンテナンスするときのように、REDO ログを頻繁に再利用する必要はありません。Oracle はログ・スイッチのたびにアーカイブすることを強制されていないため、災害発生時には、プライマリ・データベースがスタンバイ・サイトにログをアーカイブする前に、複数のログが消失する可能性があります。

フェイルオーバー後のスタンバイ・データベースの接続方法

プライマリ・データベースに障害が発生した場合、ユーザーはシームレスに、アクティブにされたスタンバイ・データベースへの接続を行う必要があります。たとえば、簡単な解決策として、スタンバイ・サービス名のエントリを含むように、クライアントの tnsnames.ora ファイルまたは Oracle Names Server を構成することができます。フェイルオーバー後には、ユーザーはプライマリのサービス名のかわりに、スタンバイのサービス名を使用してデータベースに接続できます。

その他のフェイルオーバーの解決策としては、次のものがあります。

- Net8 のフェイルオーバー機能
- サーバーの IP/DNS 構成
- アプリケーションのリルート

どの方法を使用するかを決定するときには、解決策の簡便さと、フェイルオーバーの円滑さを比較検討してください。

スタンバイ・データベースのセットアップ: 基本作業

スタンバイ・データベースの作成が決定し、データベースの配置場所を決定後、データベースの準備を開始できます。スタンバイ・データベースの作成手順は、管理スタンバイ環境を設定するのか、非管理スタンバイ環境を設定するのかによって異なります。

この項は、次のトピックから構成されています。

- [管理リカバリのためのスタンバイ・データベースの準備: 基本作業](#)
- [手動リカバリのためのスタンバイ・データベースの準備: 基本作業](#)

**関連項目：** スタンバイ・データベース自体の説明およびその仕組みの概要は、[第 1 章](#)を参照してください。管理スタンバイ環境などの重要な用語の説明は、1-5 ページの「[概念および用語](#)」を参照してください。



## 管理リカバリのためのスタンバイ・データベースの準備: 基本作業

管理リカバリ用にスタンバイ・データベースをセットアップするためには、一連の異なる作業を実行する必要があります。準備が完了し、管理リカバリを開始すると、スタンバイ・データベースはプライマリ・データベースから REDO ログを受け取るたびに、自動的にかつ継続的にログを適用します。

表 2-2 には、スタンバイ・データベースのセットアップおよび管理リカバリを開始できるようにするためのデータベースの同期化についての基本作業を簡単に説明しています。

**表 2-2 作業リスト: 管理リカバリの準備**

	作業	手順
1	プライマリ・データベースのデータ・ファイルの新しいバックアップを作成するか、古いバックアップにアクセスします。	2-7 ページの「スタンバイのデータ・ファイルの作成」
2	プライマリ・データベースに接続し、スタンバイの制御ファイルを作成します。	2-8 ページの「スタンバイの制御ファイルの作成」
3	バックアップのデータ・ファイルと制御ファイルを、プライマリ・サイトからスタンバイ・サイトにコピーします。	2-9 ページの「スタンバイ・サイトへのファイル転送」
4	スタンバイ・データベースのサービス名を作成します。	2-12 ページの「プライマリ・サイトの <code>tnsnames.ora</code> ファイルの構成」
5	プライマリ・サイトからアーカイブ REDO ログを受け取ることができるように、スタンバイ・サイトにリスナーを構成します。	2-13 ページの「スタンバイ・サイトの <code>listener.ora</code> ファイルの構成」
6	プライマリ・データベースの初期化パラメータを設定します。	2-15 ページの「プライマリ・データベースの初期化パラメータ・ファイルの構成」
7	スタンバイの初期化パラメータ・ファイルを作成し、スタンバイ・データベースの初期化パラメータを設定します。構成によっては、ファイル名変換パラメータを設定する必要があります。	2-19 ページの「スタンバイ・データベースの初期化パラメータ・ファイルの構成」
8	スタンバイ・インスタンスを起動し、スタンバイ・データベースをマウントします。	2-31 ページの「スタンバイ・インスタンスの起動」
9	ステップ 7 で、 <code>DB_FILE_NAME_CONVERT</code> および <code>LOG_FILE_NAME_CONVERT</code> を使用して自動的に改名されないすべてのファイルについて、スタンバイの制御ファイル内で、プライマリのデータ・ファイルおよび REDO ログの名前を手動で変更します。ステップ 7 ですべてのファイルを改名していた場合は、次の作業に進んでください。	2-31 ページの「変換パラメータで取り込めないスタンバイのファイルの手動での改名」

表 2-2 ( 続き ) 作業リスト: 管理リカバリの準備

	作業	手順
10	スタンバイ・サイトへのアーカイブを開始できるように、プライマリ・データベースの初期化パラメータの変更を手動で有効にします。	2-32 ページの「初期化パラメータの設定値の変更の有効化」

手動リカバリのためのスタンバイ・データベースの準備: 基本作業

表 2-3 は、手動リカバリのためのスタンバイ・データベースをセットアップするための基本作業を示しています。この手順は、Net8 を通じてスタンバイ・データベースに接続することを想定しています。スタンバイ・データベースへの接続に Net8 を使用しない場合は、作業 4 および 5 を省略してください。

表 2-3 作業リスト: 手動リカバリの準備

	作業	手順
1	プライマリ・データ・ファイルの新しいバックアップを作成するか、古いバックアップにアクセスします。	2-7 ページの「スタンバイのデータ・ファイルの作成」
2	プライマリ・データベースに接続し、スタンバイの制御ファイルを作成します。	2-8 ページの「スタンバイの制御ファイルの作成」
3	バックアップのデータ・ファイルとスタンバイの制御ファイルを、プライマリ・サイトからスタンバイ・サイトにコピーします。	2-9 ページの「スタンバイ・サイトへのファイル転送」
4	スタンバイ・データベースへの Net8 接続を作成する場合は、サービス名を作成します。	2-12 ページの「プライマリ・サイトの tnsnames.ora ファイルの構成」
5	スタンバイ・データベースへの Net8 接続を作成する場合は、スタンバイ・インスタンスへの接続要求を受け取ることができるように、スタンバイ・サイトにリスナーを構成します。	2-13 ページの「スタンバイ・サイトの listener.ora ファイルの構成」
6	スタンバイ・サイトにスタンバイの初期化パラメータ・ファイルを作成し、スタンバイ・データベースの初期化パラメータを設定します。オプションとして、スタンバイの制御ファイル内でプライマリ・ファイルを自動的に改名するために、DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を設定します。	2-19 ページの「スタンバイ・データベースの初期化パラメータ・ファイルの構成」
7	スタンバイ・インスタンスを起動し、スタンバイ・データベースをマウントします。	2-31 ページの「スタンバイ・インスタンスの起動」

表 2-3 ( 続き ) 作業リスト : 手動リカバリの準備

	作業	手順
8	スタンバイ・データベースに接続中に、ステップ 6 で DB_FILE_NAME_CONVERT および LOG_FILE_NAME_CONVERT を使用して自動的に改名されないすべてのファイルについて、スタンバイの制御ファイル内で、プライマリのデータ・ファイルおよび REDO ログの名前を手動で変更します。ステップ 6 ですべてのファイルを改名した場合は、次の作業に進んでください。	2-31 ページの「 <a href="#">変換パラメータで取り込めないスタンバイのファイルの手動での改名</a> 」

## スタンバイ・データベース・ファイルの作成

スタンバイ・データベースは、プライマリ・データベースと同じホスト上でも、別のホスト上でも作成できます。同じホストにスタンバイ・データベースを作成するときには、スタンバイ・データベース・ファイルを作成するときにプライマリ・データベースのファイルを上書きしないように、作成手順に十分注意してください。

スタンバイ・データベース・ファイルの作成は 3 つの段階で行われます。

1. [スタンバイのデータ・ファイルの作成](#)
2. [スタンバイの制御ファイルの作成](#)
3. [スタンバイ・サイトへのファイル転送](#)

## スタンバイのデータ・ファイルの作成

最初に、プライマリ・データベースのデータ・ファイルのバックアップを作成します。スタンバイのデータ・ファイルは、このバックアップから作成します。

データベースの完全リカバリに必要な REDO ログがアーカイブされている限り、プライマリ・データベースのどのバックアップを使用してもかまいません。古いバックアップまたは新しいバックアップ、一貫性バックアップまたは非一貫性バックアップを使用できます。ホット・バックアップには、バックアップの実行中もデータベースをオープンしたままにできるという利点があります。しかし、多数のアーカイブ REDO ログを適用する必要がないため、新しい、クローズ状態の一貫性バックアップを使用するのが便利です。

### スタンバイ・データベースのベースとする一貫性 / 全体バックアップの作成

1. プライマリ・データベースで SQL\*Plus セッションを開始し、V\$DATAFILE ビューを問い合わせてプライマリのデータ・ファイルのリストを入手します。たとえば、次のように入力します。

```
SQL> SELECT name FROM v$datafile;
NAME
-----
/oracle/dbs/tbs_01.f
/oracle/dbs/tbs_02.f
/oracle/dbs/tbs_03.f
/oracle/dbs2/tbs_11.f
/oracle/dbs2/tbs_12.f
/oracle/dbs3/tbs_21.f
/oracle/dbs3/tbs_22.f
7 rows selected.
```

2. プライマリ・データベースをクリーンにシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

3. オペレーティング・システムのユーティリティを使用し、プライマリ・データベースからデータ・ファイルの一貫性バックアップを作成します。たとえば、/backup 一時ディレクトリにすべてのデータ・ファイルをコピーするには、次のように入力します。

```
% cp /oracle/dbs/*.f /backup
% cp /oracle/dbs2/*.f /backup
% cp /oracle/dbs3/*.f /backup
```

4. プライマリ・データベースをリオープンします。たとえば、次のように入力します。

```
SQL> STARTUP pfile=initPROD1.ora;
```

**関連項目：** オペレーティング・システムのバックアップ方法は、『Oracle8i バックアップおよびリカバリ・ガイド』を参照してください。

## スタンバイの制御ファイルの作成

スタンバイのデータ・ファイルとして使用するためのバックアップを作成後、スタンバイ・データベースの制御ファイルを作成します。制御ファイルは、バックアップ・データ・ファイルの最終タイムスタンプよりも後に作成されている必要があります。

---

---

**注意：** プライマリ・データベースとスタンバイ・データベースの両方で1つの制御ファイルを使用することはできません。スタンバイ・インスタンスはプライマリ・インスタンスには依存しないため、専用のデータベース・ファイルを所有する必要があります。

---

---

スタンバイ・データベースの制御ファイルの作成

1. プライマリ・データベースのモードが ARCHIVELOG モードであり、アーカイブが使用可能になっていることを確認します。ARCHIVE LOG LIST 文を発行するか、V\$DATABASE ビューを問い合わせます。

```
SQL> CONNECT sys/change_on_install@prod1
SQL> ARCHIVE LOG LIST
Database log mode           Archive Mode
Automatic archival         Enabled
Archive destination        /vobs/oracle/work/arc_dest/arc
Oldest online log sequence  821
Next log sequence to archive 822
Current log sequence        822
```

2. プライマリ・データベースに接続し、スタンバイ・データベースの制御ファイルを作成します。たとえば、スタンバイの制御ファイルをプライマリ・サイト上に oracle/dbs/stbycf.f として作成する場合は、次のように入力します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/oracle/dbs/stbycf.f';
```

作成するスタンバイの制御ファイルのファイル名は、プライマリ・データベースの現在の制御ファイルのファイル名とは異なるものにする必要がありますので注意してください。

**関連項目：** ALTER DATABASE の構文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

スタンバイ・サイトへのファイル転送

スタンバイのデータ・ファイルおよび制御ファイルを正常に作成した後、オペレーティング・システムのユーティリティを使用し、ファイルをスタンバイ・サイトに転送します。たとえば、スタンバイ・サイトとプライマリ・サイトが同じホスト上にある場合には、UNIX の cp コマンドを使用してファイルを転送します。スタンバイとプライマリが異なるホスト上にある場合には、ftp を使用します。

スタンバイ・データベースの場所	作業
プライマリ・データベースと同じディレクトリを持つ別のホスト	スタンバイのファイルではプライマリのファイルと同じパス名を使用できます。この場合、スタンバイの制御ファイル内でプライマリのデータ・ファイルを改名する必要はありません。

スタンバイ・データベースの場所	作業
プライマリ・データベースと同じ ホストまたは異なるディレクトリ 構造を持つ別のホスト	スタンバイ・サイトにコピーした後で、スタンバイの制御 ファイル内のプライマリのデータ・ファイルを改名する必 要があります。次の方法を使用できます。 <ul style="list-style-type: none"><li>■ ファイル名変換の初期化パラメータを設定します (2-21 ページの「<a href="#">スタンバイの制御ファイル内のプ ライマリのファイル名の改名</a>」を参照)。</li><li>■ ALTER DATABASE 文を使用してファイルを手動で改 名します (2-31 ページの「<a href="#">変換パラメータで取り込め ないスタンバイのファイルの手動での改名</a>」を参照)。</li><li>■ 変換パラメータと手動の改名を組み合わせて使用しま す。</li></ul>

データ・ファイルおよび制御ファイルのスタンバイ・サイトへの転送

オペレーティング・システムのコマンドまたはユーティリティを使用し、作成した制御ファイルおよびデータ・ファイルのバックアップをスタンバイ・サイトに転送します。バイナリ・ファイルの転送に適した方法を使用してください。

1. 転送時間が短いため、最初に制御ファイルを転送してください。たとえば、次のように入力します。

```
% cp /backup/db.cf /standby/oracle/dbs/db.cf
```

2. バックアップ・データ・ファイルを転送します。たとえば、次のように入力します。

```
% cp /backup/*.df /standby/oracle/dbs
```

3. 使用可能なすべてのアーカイブ REDO ログをスタンバイ・サイトに転送します。たとえば、次のように入力します。

```
% cp /arc_dest/*.arc /standby/arc_dest
```

プライマリおよびスタンバイ・データベース用のネットワーク・ファイルの構成

ネットワーク構成はシステムごとに大きく異なります。プライマリ・サイトからスタンバイ・サイトにアーカイブを行う管理スタンバイ環境をインプリメントする予定がないときには、Net8 は必要ありません。オペレーティング・システムの認証を使用してスタンバイ・データベースに接続できます。しかし、管理スタンバイ環境をインプリメントするためには、スタンバイ・インスタンスのサービス名を作成する必要があります。このため、Net8 を使用する必要があります。

管理スタンバイ環境では、スタンバイ・サイトがプライマリ・サイトからアーカイブ・ログを受け取れるように、スタンバイ・サイトに listener.ora ファイルを構成する必要があります。ネーム・サーバーを使用しない場合は、プライマリ・サイトに tnsnames.ora ファイルを構成する必要があります。また、SQL\*Net.ora および names.ora などの追加のネットワーク・ファイルを構成する必要があることもあります。Net8 構成の詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

プライマリ・サイトの tnsnames.ora ファイル内の情報は、スタンバイ・サイトの listener.ora ファイル内の情報と対応している必要があります。次の表は、一致させる必要のあるパラメータと、その設定値の例を示しています。tnsnames.ora ファイルおよび listener.ora ファイルで一致させる必要のある値は太字で示しています。

値	tnsnames.ora の設定例	listener.ora の設定例
PORT	sbdb = (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp) <b>(PORT=1512)</b> (HOST=hst1)) (CONNECT_DATA = (SID=stby1)))	LISTENER = (ADDRESS_LIST= (ADDRESS=(PROTOCOL=tcp) <b>(PORT=1512)</b> (HOST=hst1)))
HOST	sbdb = (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp) <b>(PORT=1512)</b> ( <b>HOST=hst1</b> )) (CONNECT_DATA = (SID=stby1)))	LISTENER = (ADDRESS_LIST= (ADDRESS=(PROTOCOL=tcp) <b>(PORT=1512)</b> ( <b>HOST=hst1</b> )))
SID (tnsnames) = SID_NAME (listener)	sbdb = (DESCRIPTION= (ADDRESS=(PROTOCOL=tcp) (PORT=1512)(HOST=hst)) (CONNECT_DATA = ( <b>SID=stby1</b> )))	SID_LIST_LISTENER = (SID_LIST= (SID_DESC= ( <b>SID_</b> <b>NAME=stby1</b> )(ORACLE_ HOME=/oracle)))
PROTOCOL	sbdb = (DESCRIPTION= (ADDRESS= <b>(PROTOCOL=tcp)</b> (PORT=1512)(HOST=hst1)) (CONNECT_DATA = (SID=stby1)))	LISTENER = (ADDRESS_LIST= (ADDRESS= <b>(PROTOCOL=tcp)</b> (PORT=1512)(HOST=hst1)))
KEY	sbdb = (DESCRIPTION= (ADDRESS=(PROTOCOL=ipc) <b>(KEY=foo)</b> )(CONNECT_ DATA=(SID=stby1)))	LISTENER = (ADDRESS_LIST= (ADDRESS = (PROTOCOL=ipc) <b>(KEY=foo)</b> )))

典型的な構成でネットワーク・ファイルを変更するための一般的な手順を紹介します。この項は、次のトピックから構成されています。

- プライマリ・サイトの tnsnames.ora ファイルの構成
- スタンバイ・サイトの listener.ora ファイルの構成

## プライマリ・サイトの tnsnames.ora ファイルの構成

スタンバイ・サイトに REDO ログをアーカイブする場合に、Oracle Names Server を使用していない場合は、プライマリ・サイトで tnsnames.ora ファイルを構成する必要があります。このファイルは、スタンバイ・インスタンスのサービス名を含みます。

### TCP 接続用の tnsnames.ora ファイルの変更

1. テキスト・エディタを起動し、tnsnames.ora ファイルをオープンします。このファイルは、通常 \$ORACLE\_HOME/network/admin に入っています。
2. tnsnames.ora ファイルに次のエントリを追加し、*standby\_service\_name*、*port\_number*、*host\_name* および *standby\_sid* の適切な値を代入します。SID が listener.ora ファイルの SID\_NAME と一致し、PORT 値が 2 つのファイルで同じになっていることを確認します。

```
standby_service_name = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (PORT=port_number) (HOST=host_name))
  (CONNECT_DATA= (SID=standby_sid))
)
```

たとえば、スタンバイ・データベース STANDBY1 がホスト REMOTE2 上にある場合には、次のように入力できます。

```
standby1 = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (PORT=1512) (HOST=remote2))
  (CONNECT_DATA= (SID=standby1))
)
```

### IPC 接続用の tnsnames.ora ファイルの変更

1. テキスト・エディタを起動し、tnsnames.ora ファイルをオープンします。このファイルは通常は \$ORACLE\_HOME/network/admin に入っています。
2. 次のエントリを tnsnames.ora ファイルに追加し、*standby\_service\_name*、*key\_handle* および *standby\_sid* の適切な値を代入します。SID が listener.ora ファイルの SID\_NAME と一致し、KEY 値が 2 つのファイルで同じになっていることを確認します。

```
standby_service_name = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=ipc) (KEY=key_handle))
  (CONNECT_DATA= (SID=standby_sid))
)
```

たとえば、スタンバイ・データベース STANDBY1 の場合、次のように入力します。

```
standby1 = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=ipc) (KEY=stby))
  (CONNECT_DATA= (SID=standby1))
)
```



**関連項目：** tnsnames.ora ファイルまたはネットワーク構成パラメータの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

## スタンバイ・サイトの listener.ora ファイルの構成

REDO ログをスタンバイ・サイトにアーカイブするためには、スタンバイ・サイトで listener.ora ファイルを構成する必要があります。リスナーはクライアント・アプリケーションからネットワーク接続要求を受け取り、要求をスタンバイ・サーバーにリルートします。

### TCP 接続用の listener.ora ファイルの変更

1. テキスト・エディタを起動し、listener.ora ファイルをオープンします。このファイルは通常は \$ORACLE\_HOME/network/admin に入っています。
2. *port\_number* および *host\_name* の適切な値を代入し、ファイルにリスナーを追加します。新しいリスナーを作成し、自由に名前を付けることもできますし、既存のリスナーに新しいアドレスを追加することもできます。

```
STANDBY_LISTENER = (ADDRESS_LIST=
  (ADDRESS= (PROTOCOL=tcp) (PORT=port_number) (HOST=host_name))
)
```

たとえば、スタンバイ・ホストの名前が REMOTE2 の場合には、次のように入力します。

```
STANDBY_LISTENER = (ADDRESS_LIST=
  (ADDRESS= (PROTOCOL=tcp) (PORT=1512) (HOST=remote2))
)
```

3. リスナーのために SID リストに次のエントリを追加し、*standby\_sid\_name* および *oracle\_home* の適切な値を代入します。

```
SID_LIST_STANDBY_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=standby_sid_name) (ORACLE_HOME=/oracle_home))
)
```

たとえば、スタンバイ・データベースの名前が STANDBY1 の場合には、次のように入力します。

```
SID_LIST_STANDBY_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=standby1) (ORACLE_HOME=/oracle))
)
```

4. リスナーがすでに起動している場合には、stop と入力してから start と入力します。リスナーが起動していない場合には、オペレーティング・システム・コマンドを使用してリスナーを起動します。たとえば、UNIX システムでは次のように入力します。

```
% lsnrctl

LSNRCTL for Solaris: Version 8.1.5.0.0 - Production on 23-MAR-99 12:04:10

(c) Copyright 1998 Oracle Corporation. All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
```

5. status コマンドを発行し、サービスが使用可能であることを確認します。

```
LSNRCTL> status
```

スタンバイ・サービスの情報が正しく構成されている場合には、有効なサービス名のリスト中にスタンバイ・サービスが表示されます。

```
standby1                has 1 service handler(s)
```

### IPC 接続用の listener.ora ファイルの変更

1. テキスト・エディタを起動し、listener.ora ファイルをオープンします。このファイルは通常は \$ORACLE\_HOME/network/admin に入っています。
2. *key\_handle* の適切な値を代入し、ファイルにリスナーを追加します。新しいリスナーを作成し、自由に名前を付けることも、既存のリスナーに新しいアドレスを追加することもできます。

```
STANDBY_LISTENER = (ADDRESS_LIST=
  (ADDRESS = (PROTOCOL=ipc) (KEY=key_handle))
)
```

たとえば、次のように入力します。

```
STANDBY_LISTENER = (ADDRESS_LIST=
  (ADDRESS = (PROTOCOL=ipc) (KEY=stby))
)
```

3. リスナーのために SID リストに次のエントリを追加し、*standby\_sid\_name* および *oracle\_home* の適切な値を代入します。

```
SID_LIST_STANDBY_LISTENER = (SID_LIST=
  (SID_DESC = (SID_NAME=standby_sid_name) (ORACLE_HOME=oracle_home))
)
```

たとえば、スタンバイ・データベースの名前が STANDBY1 の場合には、次のように入力します。

```
SID_LIST_STANDBY_LISTENER = (SID_LIST=
  (SID_DESC = (SID_NAME=standby1) (ORACLE_HOME=/oracle))
)
```

)

4. リスナーがすでに起動している場合には、`stop` と入力してから `start` と入力します。リスナーが起動していない場合には、オペレーティング・システム・コマンドを使用してリスナーを起動します。たとえば、UNIX システムでは次のように入力します。

```
% lsnrctl
```

```
LSNRCTL for Solaris: Version 8.1.5.0.0 - Production on 23-MAR-99 12:04:10
```

```
(c) Copyright 1998 Oracle Corporation. All rights reserved.
```

```
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> start
```

5. `status` コマンドを発行し、サービスが使用可能であることを確認します。

```
LSNRCTL> status
```

スタンバイ・サービスの情報が正しく構成されている場合には、有効なサービス名のリスト中にスタンバイ・サービスが表示されます。

```
standby1                has 1 service handler(s)
```

## プライマリ・データベースの初期化パラメータ・ファイルの構成

管理スタンバイ環境をインプリメントする予定がない場合には、プライマリ・システムの初期化パラメータ・ファイルを変更する必要はありません。アーカイブ REDO ログをスタンバイ・データベースに手動で転送および適用する必要があるため、アーカイブ先を変更する必要はありません。

管理スタンバイ環境をインプリメントする予定がある場合には、プライマリ・システムの初期化パラメータ・ファイルに、新しいアーカイブ先を追加する必要があります。この項では、管理スタンバイ環境をインプリメントすることを想定しています。

プライマリ・データベースからローカルまたはリモートのスタンバイ・データベースの場所にアーカイブを行うためには、次の条件が満たされる必要があります。

- プライマリの初期化パラメータ・ファイル内の `LOG_ARCHIVE_DEST_n` パラメータ（この  $n$  は 1 から 5 までの整数）が、`SERVICE` 属性を使用して正しく定義されていること。`SERVICE` 属性は、ローカルまたはリモートの、すべてのスタンバイ・データベースの場所に対して使用する必要があります。
- プライマリ・サイトの `tnsnames.ora` ファイルと、スタンバイ・サイトの `listener.ora` ファイル内のエントリが、正しく対応していること。

- スタンバイ・サイトでリスナーが起動していること。
  - スタンバイ・サイトでスタンバイ・インスタンスが起動していること。
- この項は、次のトピックから構成されています。

- [アーカイブ先の指定](#)
- [必須およびオプションのアーカイブ先の指定](#)
- [アーカイブ先の状態を使用可能にする方法](#)
- [失敗した宛先に再アーカイブするための Oracle の構成](#)

**関連項目：** アーカイブ REDO ログの管理の詳細は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。Oracle ネットワーク・オプションの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。アーカイバ・プロセスの概要は、『Oracle8i 概要』を参照してください。

## アーカイブ先の指定

LOG\_ARCHIVE\_DEST\_ *n* 初期化パラメータ（この *n* は 1 から 5 までの整数）を設定し、プライマリ・データベースのアーカイブ・ログの宛先を最大で 5 つ指定してください。次の例で示すように、接尾辞の数値パラメータは、それぞれが個別の宛先を指定します。

```
# first local archiving destination
LOG_ARCHIVE_DEST_1 = 'LOCATION=/oracle/arc/'

# second local archiving destination
LOG_ARCHIVE_DEST_2 = 'LOCATION=/oracle/arc2/'

# third standby archiving destination
LOG_ARCHIVE_DEST_3 = 'SERVICE=stby'
```

管理リカバリ・モードでスタンバイ・データベースに REDO ログをアーカイブする場合には、SERVICE キーワードと一緒に LOG\_ARCHIVE\_DEST\_ *n* を使用する必要があります。少なくとも 1 つの宛先は、スタンバイ・サイトではない、ローカル・ディレクトリにする必要がありますので注意してください。

LOG\_ARCHIVE\_DEST\_ *n* を使用する場合は、次のキーワードを使用してアーカイブ場所を指定します。

キーワード	意味	例
LOCATION	ローカル・ディレクトリ	LOG_ARCHIVE_DEST_1= 'LOCATION=/arc_dest/'
SERVICE	Net8 サービス名によるアーカイブ	LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1'

LOCATION キーワードを使用するときには、オペレーティング・システムの有効なパス名を指定してください。SERVICE が指定されると、Oracle は tnsnames.ora ファイルを通じてネット・サービス名を接続記述子に変換します。この記述子には、スタンバイ・データベースへの接続に必要な情報が含まれています。

**関連項目：** LOG\_ARCHIVE\_DEST\_*n* およびアーカイブ・プロセスの詳細は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。

## 必須およびオプションのアーカイブ先の指定

LOG\_ARCHIVE\_DEST\_*n* パラメータを使用すると、宛先の属性を OPTIONAL（デフォルト）にするか、MANDATORY にするかを指定できます。たとえば、次のようにパラメータを設定できます。

```
LOG_ARCHIVE_DEST_3 = 'SERVICE=standby1 MANDATORY'
```

オラクル社は、ローカル・ディレクトリ宛先を MANDATORY として指定することをお勧めします。

LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST=*m* パラメータ（この *m* は 1 から 5 までの整数）は、LGWR がオンライン REDO ログを上書きするまでにアーカイブが成功する必要がある宛先の数を指定します。MANDATORY のすべての宛先およびスタンバイではない OPTIONAL の宛先は、LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST=*m* カウントの数に加えられます。たとえば、次のようにパラメータを設定できます。

```
LOG_ARCHIVE_MIN_SUCCEED_DEST = 2    # Oracle must archive to at least two
                                     # locations before overwriting the online
                                     # redo logs.
```

**関連項目：** OPTIONAL および MANDATORY キーワードの詳細は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。

## アーカイブ先の状態を使用可能にする方法

LOG\_ARCHIVE\_DEST\_STATE\_*n*（この *n* は 1 から 5 までの整数）初期化パラメータは、対応する LOG\_ARCHIVE\_DEST\_*n* パラメータ（この *n* は同じ整数）で指定された宛先の状態を指定します。たとえば、LOG\_ARCHIVE\_DEST\_STATE\_3 パラメータは、LOG\_ARCHIVE\_DEST\_3 というアーカイブ先の状態を指定します。

アーカイブ先のパラメータには、ENABLE および DEFER という 2 つの値を使用できます。ENABLE は Oracle が宛先にアーカイブできることを指定し、DEFER はアーカイブできないことを指定します。

たとえば、次のようにパラメータを設定できます。

```
LOG_ARCHIVE_DEST_STATE_2 = ENABLE  
LOG_ARCHIVE_DEST_2 = 'SERVICE=stby1'
```

**関連項目：** アーカイブ先の状態の詳細は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。LOG\_ARCHIVE\_DEST\_STATE\_*n* パラメータの説明は、『Oracle8i リファレンス・マニュアル』を参照してください。

## 失敗した宛先に再アーカイブするための Oracle の構成

エラー後に、失敗した宛先に再アーカイブさせるかどうか、さらにそのタイミングを指定するためには、LOG\_ARCHIVE\_DEST\_*n* パラメータを使用します。REOPEN=*s*（この *s* は整数）オプションは、失敗した宛先にアーカイバ・プロセスによる再アクセスを試みるまでに待つ秒数を指定します。REOPEN は OPEN エラーのみでなく、すべてのエラーに適用されます。

REOPEN を指定した場合、キーワードのデフォルト値は 300 秒です。REOPEN を指定しないと、値は 0 になりますが、これはこのオプションをオフにすることと同じです。REOPEN オプションをオフにすると、アーカイバ・プロセスはエラー後に宛先への再アクセスを行いません。

REOPEN を使用して、再接続およびアーカイブ・ログの転送の試行回数の制限を指定することはできません。REOPEN は成功するか失敗するかのいずれかであり、この結果 REOPEN 情報はリセットされます。

たとえば、次のようにパラメータを設定すると、リオープン時間を 60 秒に指定できます。

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=standby2 OPTIONAL REOPEN=60'
```

**関連項目：** REOPEN の使用方法の詳細は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。

## スタンバイ・データベースの初期化パラメータ・ファイルの構成

プライマリ・データベースの初期化パラメータ・ファイルを構成すると、スタンバイ・データベースで使用するためにこのファイルを複製できます。スタンバイの初期化パラメータ・ファイルを作成するための手順は次のとおりです。

1. オペレーティング・システムのユーティリティを使用し、プライマリ・データベースの初期化パラメータ・ファイルをコピーします。
2. スタンバイ・データベースで使用するために、初期化パラメータ・ファイルを編集します。
3. 適切なオペレーティング・システムのユーティリティを使用し、スタンバイ・サイトに初期化パラメータ・ファイルを転送します。

この項は、次のトピックから構成されています。

- [スタンバイの初期化パラメータの構成 : 一般的な考慮事項](#)
- [スタンバイの制御ファイル内のプライマリのファイル名の改名](#)
- [スタンバイ・データベースのアーカイブ REDO ログのファイル名の指定](#)
- [スタンバイの初期化パラメータ・ファイルの構成 : 典型的な設定値](#)

### スタンバイの初期化パラメータの構成 : 一般的な考慮事項

一般的には、ほとんどの初期化パラメータはプライマリ・データベースとスタンバイ・データベースで同じにする必要がありますが、`CONTROL_FILES` および `DB_FILE_NAME_CONVERT` などの初期化パラメータは変更する必要があります。次の表に記載されているものとは異なる初期化パラメータを使用した場合、スタンバイ・データベースのパフォーマンスが低下したり、場合によっては、スタンバイ・データベースの稼動が停止することがあります。パラメータ値の変更は、スタンバイ・データベースの機能上必要な場合、またはファイル名の変換のみに限ってください。

[表 2-4](#) に記載された初期化パラメータは、スタンバイ・データベースの構成に重要な役割を果たします。

表 2-4 スタンバイ・データベースの初期化パラメータの構成

パラメータ	ガイドライン
COMPATIBLE	プライマリ・データベースとスタンバイ・データベースで必ず同じ設定にしてください。設定が異なる場合、プライマリ・データベースからスタンバイ・データベースにログを適用できなくなることがあります。
DB_NAME	必ずプライマリ・データベースの DB_NAME の値と同じ設定にしてください。
CONTROL_FILES	必ずプライマリ・データベースの CONTROL_FILES の値と異なる値を設定してください。スタンバイ・データベースの制御ファイルの名前が、スタンバイ・サイトに存在している必要があります。
DB_FILE_NAME_CONVERT	たとえば、両方のデータベースが同じホスト上にある場合、または別のスタンバイ・ホストがプライマリ・ホストとは異なるパス名を使用する場合など、スタンバイのデータ・ファイルのファイル名とプライマリのデータ・ファイルのファイル名を区別するために設定します。スタンバイの制御ファイルはプライマリの制御ファイルのコピーであるため、スタンバイ・データベースがプライマリ・データベースと同じホスト上にある場合、または別のホスト上で異なるパス名を使用する場合には、スタンバイのファイル名を変換する必要があります。2-21 ページの「 <a href="#">スタンバイの制御ファイル内のプライマリのファイル名の改名</a> 」も参照してください。
LOCK_NAME_SPACE	スタンバイ・データベースとプライマリ・データベースでホストを共有する場合は、必ずこの値を設定してください。分散ロック・マネージャがロック名の作成に使用するネーム・スペースを指定します。
LOG_ARCHIVE_DEST	手動リカバリ・モードのスタンバイ・データベースで、アーカイブ・ログの場所を指定します。スタンバイ・データベースで手動リカバリを実行する場合、Oracle は LOG_ARCHIVE_DEST またはユーザー定義のファイル名を使用してログを検索します。  このパラメータは、管理リカバリでも使用されます。スタンバイ・サイトでログが欠落しているために管理リカバリが停止した場合は、RECOVER STANDBY DATABASE を発行して手動リカバリを開始する必要がありますが、これにより、Oracle はデフォルトで、LOG_ARCHIVE_DEST からログを検索するようになります。一般的には、管理リカバリでは STANDBY_ARCHIVE_DEST および LOG_ARCHIVE_DEST を同じ値に設定します。



表 2-4 ( 続き ) スタンバイ・データベースの初期化パラメータの構成

パラメータ	ガイドライン
LOG_ARCHIVE_ TRACE	オプションとして、このパラメータを整数の値に設定すると、スタンバイ・サイトへの REDO ログのアーカイブの進捗を知ることができます。Oracle は、プライマリ・データベースから受け取ったアーカイブ・ログの監査証跡をトレース・ファイルに書き込みます。このパラメータは、プライマリ・データベースのアーカイブ・プロセス (ARCn およびフォアグラウンド・プロセス) と、スタンバイ・データベースの RFS プロセスから生成される出力を制御します。このパラメータで使用可能な整数値およびその意味の詳細は、4-5 ページの「 <a href="#">スタンバイ・サイトで受け取ったアーカイブ・ログの判別</a> 」を参照してください。
LOG_FILE_NAME_ CONVERT	スタンバイの REDO ログのファイル名と、プライマリ・データベースの REDO ログのファイル名を区別できるようにするために設定します。このパラメータ値は、プライマリ・データベース上の新しいログ・ファイルのファイル名を、スタンバイ・データベース上のログ・ファイルに変換します。プライマリ・データベースにログ・ファイルを追加した場合、スタンバイ・データベースにも対応するファイルを追加する必要があります。スタンバイ・データベースが更新された場合、このパラメータは、プライマリ・データベース上のログ・ファイル名を、スタンバイ・データベース上のログ・ファイル名に変換するために使用されます。スタンバイ・データベース上に書き込み可能な状態でファイルが存在していないと、エラーが発生し、リカバリ処理は停止します。
STANDBY_ARCHIVE_ DEST	<p>プライマリ・データベースから受け取ったアーカイブ・ログの場所を判別するために、管理リカバリ・モードのスタンバイ・データベースでのみ使用されます。管理リカバリ・モードでは、この値と、LOG_ARCHIVE_FORMAT を使用し、スタンバイ・データベースの完全修飾のログ・ファイル名を生成し、このファイル名をスタンバイの制御ファイルに格納します。管理リカバリでは、このデータを使用してドライブのリカバリを行います。</p> <p>管理リカバリでは、STANDBY_ARCHIVE_DEST および LOG_ARCHIVE_DEST を同じ値に設定してください。ギャップ・シーケンスのために手動リカバリが必要になった場合には、欠落しているログを他のログと同じ場所にコピーし、手動のリカバリを行ってください。その後、スタンバイ・データベースを管理リカバリ・モードに戻すことができます。</p>

**関連項目：** これらの初期化パラメータの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

## スタンバイの制御ファイル内のプライマリのファイル名の改名

スタンバイ・サイトでプライマリ・サイトと同じディレクトリ構造を使用する場合には、スタンバイの制御ファイル内でプライマリのファイル名を改名する必要はありません。しかし、プライマリ・データベースとスタンバイ・データベースが同じノードを使用する場合、またはプライマリ・サイトとスタンバイ・サイトで異なるディレクトリ構造を使用する場合には、スタンバイのデータ・ファイルにアーカイブ・ログを適用できるように、スタンバイの制御ファイル内でプライマリのファイルを改名する必要があります。

スタンバイ・データベースが、スタンバイ・データベースの制御ファイル内のデータに基づいて、データ・ファイルおよびアーカイブ REDO ログのファイル名を自動的に変換するように、初期化パラメータを設定できます。これらのパラメータを使用してプライマリのすべてのファイル名を自動的に改名できない場合には、手動で改名する必要があります（2-31ページの「[変換パラメータで取り込めないスタンバイのファイルの手動での改名](#)」を参照）。

**注意：** スタンバイ・データベースとプライマリ・データベースがノードを共有する場合に LOCK\_NAME\_SPACE パラメータを異なる設定にしないと、ORA-1102 エラーが発生しますので注意してください。

表 2-5 内の初期化パラメータはファイル名の自動変換を実行します。

表 2-5 ファイル名の変換

パラメータ	機能
DB_FILE_NAME_CONVERT	プライマリ・データベースのデータ・ファイルのファイル名を、スタンバイのデータ・ファイルのファイル名に変換します。たとえば、tbs_* から standbytbs_* に変換します。
LOG_FILE_NAME_CONVERT	プライマリ・データベースの REDO ログのファイル名をスタンバイ・データベースの REDO ログのファイル名に変換します。たとえば、log_* から standbylog_* に変換します。

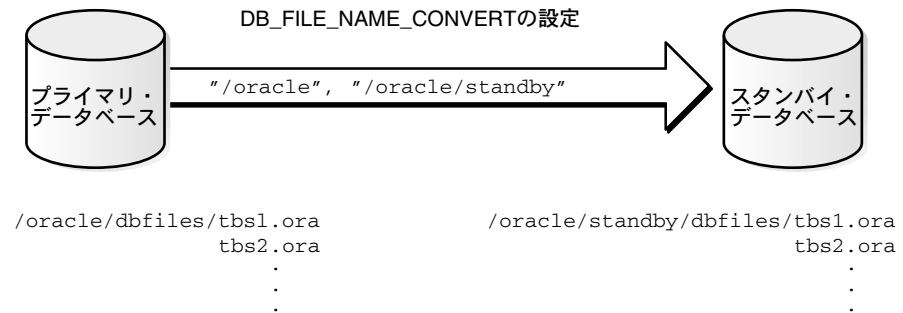
DB\_FILE\_NAME\_CONVERT は、プライマリ・データベースのデータ・ファイルのファイル名をスタンバイ・データベースのファイル名に変換するために使用します。LOG\_FILE\_NAME\_CONVERT は、プライマリ・データベースの新しい REDO ログのファイル名をスタンバイ・データベースのファイル名に変換するために使用します。プライマリ・データベースにデータ・ファイルまたはログを追加した場合、スタンバイ・データベースにも対応するファイルを追加する必要があります。

スタンバイ・データベースが更新された場合には、DB\_FILE\_NAME\_CONVERT を使用し、プライマリ・データベースのデータ・ファイルの名前を、スタンバイ・データベースのデータ・ファイルの名前に変換します。スタンバイ・データベース上に書き込み可能な状態でファイルが存在していないと、エラーが発生し、リカバリ処理は停止します。

DB\_FILE\_NAME\_CONVERT および LOG\_FILE\_NAME\_CONVERT パラメータには 2 つの文字列が必要です。最初の文字列はプライマリ・データベースのファイル名で検索する文字列です。この文字列が一致した場合には、2 番目の文字列で置換し、スタンバイ・データベースのファイル名を構成します。

図 2-1 は、ファイル名変換パラメータの仕組みを示しています。

図 2-1 ファイル名変換パラメータの設定



次の文を実行した場合には、変換パラメータは対象となるファイルに適用されません。

- ALTER TABLESPACE ADD DATAFILE
- ALTER TABLESPACE RENAME DATAFILE
- ALTER DATABASE ADD LOGFILE
- ALTER DATABASE RENAME FILE
- ALTER DATABASE CREATE DATAFILE ...AS

**関連項目：** スタンバイ・データベースにデータ・ファイルを追加する方法は、4-10 ページの「[プライマリ・データベースへの表領域またはデータ・ファイルの追加](#)」を参照してください。

## スタンバイ・データベースのアーカイブ REDO ログのファイル名の指定

スタンバイ・サイトのアーカイブ REDO ログのネーミング規則は、スタンバイ・データベースが、管理スタンバイ環境の一部であるかどうかによって異なります。この項は、次のトピックから構成されています。

- [管理リカバリのログ・ファイル名の指定](#)
- [手動リカバリのログ・ファイル名の指定](#)

### 管理リカバリのログ・ファイル名の指定

管理スタンバイ環境では、スタンバイ・データベースはスタンバイの初期化パラメータ・ファイル内の STANDBY\_ARCHIVE\_DEST パラメータを使用し、アーカイブ REDO ログを格納するディレクトリを判断します。Oracle はこの値と LOG\_ARCHIVE\_FORMAT を使用し、スタンバイ・サイトでアーカイブ・ログのファイル名を生成します。

パラメータ	意味	例
STANDBY_ARCHIVE_DEST	ログを入れるディレクトリ	STANDBY_ARCHIVE_DEST= /arc_dest/
LOG_ARCHIVE_FORMAT	ログのファイル名の書式	LOG_ARCHIVE_FORMAT = "log_%t_%s.arc" <b>注意:</b> %s は、順序番号を表します。OPS 構成に必要な %t は、スレッドを表します。

Oracle は、完全修飾のファイル名をスタンバイの制御ファイルに格納します。管理リカバリでは、この情報を使用してリカバリ操作が実行されます。この情報には V\$ARCHIVED\_LOG ビューからアクセスします。

```
SQL> SELECT name FROM v$archived_log;
NAME
```

```
-----
/arc_dest/log_1_771.arc
/arc_dest/log_1_772.arc
/arc_dest/log_1_773.arc
/arc_dest/log_1_774.arc
/arc_dest/log_1_775.arc
```

手動リカバリのログ・ファイル名の指定

RECOVER MANAGED STANDBY DATABASE は例外として、RECOVER STANDBY DATABASE 文は、次のいずれかに基いてアーカイブ・ファイルの場所を提供します。

- LOG\_ARCHIVE\_DEST の値
- ユーザー指定のファイル名

管理リカバリ・モードでスタンバイ・データベースを実行する場合でも、たとえばスタンバイ・サイトでアーカイブ REDO ログが欠落している場合など、手動リカバリが必要になることがあります。このような状況は、ネットワーク・エラーにより、プライマリ・データベースから 1 つまたは複数のアーカイブ REDO ログが転送されなかったときに発生する可能性があります。

このような状況で手動で RECOVER STANDBY DATABASE 文を発行するためには、LOG\_ARCHIVE\_DEST パラメータを使用して、必要なアーカイブ REDO ログを検索する必要があります。通常は管理リカバリ・モードを使用するスタンバイ・データベースの場合には、STANBY\_ARCHIVE\_DEST および LOG\_ARCHIVE\_DEST を同じ値に設定することをお勧めします。この方法を使用すると、スタンバイ・データベースは管理リカバリ・モードでも手動リカバリ・モードでも同じアーカイブ REDO ログのセットにアクセスできます。

## スタンバイの初期化パラメータ・ファイルの構成：典型的な設定値

スタンバイ・データベースの初期化パラメータ・ファイルの構成方法は、いくつかの、相互に関連する要因に依存します。

- プライマリ・データベースとスタンバイ・データベースは同じホスト上にあるかどうか
- スタンバイ・サイトのディレクトリ構造はプライマリ・サイトのディレクトリ構造と同じかどうか
- スタンバイ・データベースを管理リカバリ・モードで実行するか、手動リカバリ・モードで実行するかどうか
- 複数のスタンバイ・データベースを保持するかどうか

表 2-6 は、スタンバイ・データベースの構成別に、設定する必要のあるパラメータを示しています。

表 2-6 スタンバイ・データベースの初期化パラメータの設定

スタンバイ・ホスト	ディレクトリ構造	リカバリ・モード	スタンバイの初期化パラメータ
プライマリ・ホストと同じホスト	プライマリ・ホストと異なる	手動	COMPATIBLE CONTROL_FILES DB_FILE_NAME_CONVERT LOCK_NAME_SPACE LOG_ARCHIVE_DEST LOG_ARCHIVE_TRACE LOG_FILE_NAME_CONVERT
プライマリ・ホストと同じホスト	プライマリ・ホストと異なる	管理	手動リカバリと同じですが、STANDBY_ARCHIVE_DEST のみは LOG_ARCHIVE_DEST と同じ値に設定します。
プライマリ・ホストと別のホスト	プライマリ・ホストと同じ	手動	COMPATIBLE CONTROL_FILES LOG_ARCHIVE_DEST LOG_ARCHIVE_TRACE
プライマリ・ホストと別のホスト	プライマリ・ホストと同じ	管理	手動リカバリと同じですが、STANDBY_ARCHIVE_DEST のみは LOG_ARCHIVE_DEST と同じ値に設定します。

表 2-6 ( 続き ) スタンバイ・データベースの初期化パラメータの設定

スタンバイ・ホスト	ディレクトリ構造	リカバリ・モード	スタンバイの初期化パラメータ
プライマリ・ホストと別のホスト	プライマリ・ホストと異なる	手動	COMPATIBLE CONTROL_FILES DB_FILE_NAME_CONVERT LOG_ARCHIVE_DEST LOG_ARCHIVE_TRACE LOG_FILE_NAME_CONVERT
プライマリ・ホストと別のホスト	プライマリ・ホストと異なる	管理	手動リカバリと同じですが、STANDBY_ARCHIVE_DEST のみは LOG_ARCHIVE_DEST と同じ値に設定します。

**例：プライマリ・データベースと同じホスト上の管理スタンバイ・データベース** プライマリ・データベース PROD1 と同じホスト上にある、管理スタンバイ・データベースの初期化パラメータの設定例は次のとおりです。

```
COMPATIBLE = 8.1.6
CONTROL_FILES = /oracle/work/standby/dbs/cf1.f
DB_FILE_NAME_CONVERT = /oracle/dbs/tbs, /oracle/work/standby/dbs/standby
DB_NAME = prod1
LOCK_NAME_SPACE = stby # required for a standby on the same host as the primary
LOG_ARCHIVE_DEST = /oracle/work/standby/dbs/arc/
LOG_ARCHIVE_FORMAT = arc_%t_%s.ar
LOG_ARCHIVE_TRACE = 8
LOG_FILE_NAME_CONVERT = /oracle/dbs/t1, /oracle/work/standby/dbs/standby_t1
STANDBY_ARCHIVE_DEST = /oracle/work/standby/dbs/arc/
```

**例：プライマリ・データベースと異なるホスト上の管理スタンバイ・データベース** プライマリ・データベース PROD1 とは別のホスト上にあるが、同じディレクトリ構造を使用する、管理スタンバイ・データベースの初期化パラメータの設定例は次のとおりです。

```
COMPATIBLE = 8.1.6
CONTROL_FILES = /oracle/work/standby/dbs/cf1.f
DB_NAME = prod1
LOG_ARCHIVE_DEST = /oracle/work/standby/arc_dest/
LOG_ARCHIVE_FORMAT = arc_%t_%s.ar
LOG_ARCHIVE_TRACE = 6
STANDBY_ARCHIVE_DEST = /oracle/work/standby/arc_dest/
```

**例：プライマリ・データベースと異なるホスト上の手動スタンバイ・データベース** プライマリ・データベース PROD1 とは別のホスト上にあり、異なるディレクトリ構造を使用する、手動リカバリ・モードのスタンバイ・データベースの初期化パラメータの設定例は次のとおりです。

```
COMPATIBLE = 8.1.6
CONTROL_FILES = /oracle/work/standby/cf1.f
DB_FILE_NAME_CONVERT = /oracle, /oracle/work/standby
DB_NAME = prod1
LOG_ARCHIVE_DEST = /oracle/work/arc_dest/
LOG_ARCHIVE_FORMAT = arc_%t_%s.ar
LOG_ARCHIVE_TRACE = 34
LOG_FILE_NAME_CONVERT = /oracle/dbs/t1, /oracle/work/standby/dbs/standby_t1
```

## パラメータおよびネットワーク・ファイルの構成：シナリオ

このシナリオは次のような仮定に基きます。

- プライマリ・データベース PROD1 はホスト LOCAL 上にあります。
- STANDBY1、STANDBY2 および STANDBY3 という3つのスタンバイ・データベースが存在します。
- STANDBY1 はローカル・ホスト LOCAL 上にあり、STANDBY2 はリモート・ホスト REMOTE2 上にあり、STANDBY3 はリモート・ホスト REMOTE3 上にあります。
- ホスト REMOTE2 は LOCAL と同じディレクトリ構造ですが、REMOTE3 は LOCAL とは異なるディレクトリ構造を使用しています。

**PROD1 初期化パラメータ・ファイルの設定** プライマリ・データベース PROD1 の初期化パラメータ・ファイル内の LOG\_ARCHIVE\_DEST\_1、LOG\_ARCHIVE\_DEST\_2、LOG\_ARCHIVE\_DEST\_3 および LOG\_ARCHIVE\_DEST\_4 の設定例は次のとおりです。

```
# This example specifies a local archiving destination and enables the
# destination.
```

```
LOG_ARCHIVE_DEST_1 = 'LOCATION=/arc_dest/'
LOG_ARCHIVE_DEST_STATE_1 = ENABLE
```

```
# This example specifies net service name "standby1", makes archiving mandatory,
# and enables the destination.
```

```
# A REOPEN value of 5 indicates that if the LOG_ARCHIVE_DEST_2 location
# encounters an error during archival of a redo log file, the destination
# remains inactive until the archival of a redo file is about to begin and 5
# seconds has elapsed. At that time, Oracle again attempts the archival to LOG_
# ARCHIVE_DEST_2.
```

```
# If Oracle encounters an error when archiving to a mandatory destination, the
# destination is inactive for the duration of the archival of the current redo
# log file. The destination may be reactivated (based on the REOPEN attribute)
# at the start of the archival of another redo log.
```

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 MANDATORY REOPEN=5'
LOG_ARCHIVE_DEST_STATE_2 = ENABLE
```

```
# Specifies net service name "standby2", makes archiving optional, and makes
# Oracle retry archiving after 5 seconds should an error occur. The destination
# is enabled.
```

```
LOG_ARCHIVE_DEST_3 = 'SERVICE=standby2 OPTIONAL REOPEN=5'
LOG_ARCHIVE_DEST_STATE_3 = ENABLE
```

```
# Specifies net service name "standby3", makes archiving optional, and makes
# Oracle retry archiving after 60 seconds should an error occur. The
# destination is deferred, however, which means that Oracle has currently
# disabled archiving to this destination.
```

```
LOG_ARCHIVE_DEST_4 = 'SERVICE=standby3 OPTIONAL REOPEN=60'
LOG_ARCHIVE_DEST_STATE_4 = DEFER
```

**TNS の設定** 前述の例で、スタンバイ・データベース STANDBY1、STANDBY2 および STANDBY3 に接続するための、プライマリ・ホストの tnsnames.ora ファイルの設定値は次のとおりです。

```
# The standby1 database is on the same node as the primary database. Connection
# is made through the IPC protocol.
```

```
standby1 = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=ipc)
    (KEY=local_standby))
  (CONNECT_DATA=
    (SID=stby1)
    (SERVER=DEDICATED)))
```

```
# The standby2 database is on a different node from the primary database.
# Connection is made through the TCP protocol.
```

```
standby2 = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=remote2)
    (PORT=1512))
  (CONNECT_DATA=
    (SID=stby2)
    (GLOBAL_NAME=standby2))
```



```

(SERVER=DEDICATED)))

# The standby3 database is on a different node from the primary database.
# Connection is made through the TCP protocol.
standby3 = (DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=remote3)
    (PORT=1523))
  (CONNECT_DATA=
    (SID=stby3)
    (GLOBAL_NAME=standby3)
    (SERVER=DEDICATED)))

```

**リスナーの設定** スタンバイ・データベース STANDBY1、STANDBY2 および STANDBY3 の listener.ora ファイルの設定値は次のとおりです。

```

# The listener settings for standby1 on host local
LISTENER = (ADDRESS_LIST=
  (ADDRESS=
    (PROTOCOL=ipc)
    (KEY=local_standby)))

SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=stby1) (ORACLE_HOME=/oracle))

# The listener settings for standby2 on the remote host remote2
LISTENER = (ADDRESS_LIST=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=remote2)
    (PORT=1512)))

SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=stby2) (ORACLE_HOME=/oracle))

# The listener settings for standby3 on the remote host remote3
LISTENER = (ADDRESS_LIST=
  (ADDRESS=
    (PROTOCOL=tcp)
    (HOST=remote3)
    (PORT=1523)))

SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=stby3) (ORACLE_HOME=/fs3/oracle))

```

**STANDBY1、STANDBY2 および STANDBY3 初期化パラメータ・ファイルの設定値** 前述の例の、スタンバイ・データベース STANDBY1、STANDBY2 および STANDBY3 の初期化パラメータ・ファイルの設定値は次のとおりです。これらの設定値により、スタンバイ・データベースでのアーカイブ REDO ログのファイル名が決定します。

```
# The init.ora values for the standby1 database, which is on the same host as
# the primary database
STANDBY_ARCHIVE_DEST = /oracle/standby/arc/
LOG_ARCHIVE_DEST = /oracle/standby/arc/ # the location is the same as
                                         # STANDBY_ARCHIVE_DEST

LOG_ARCHIVE_FORMAT = log%t_%s.arc
LOG_ARCHIVE_TRACE = 16
LOCK_NAME_SPACE = foo_bar
DB_FILE_NAME_CONVERT = /oracle/dbs, /oracle/standby/dbs
LOG_FILE_NAME_CONVERT = /oracle/dbs, /oracle/standby/dbs

# The init.ora values for the standby2 database, which is on
# host remote2. Host remote2 uses the same directory structure as host local.
STANDBY_ARCHIVE_DEST = /oracle/standby/arc/
LOG_ARCHIVE_DEST = /oracle/standby/arc/ # the location is the same as
                                         # STANDBY_ARCHIVE_DEST

LOG_ARCHIVE_FORMAT = log%t_%s.arc
LOG_ARCHIVE_TRACE = 16

# The init.ora values for the standby3 database, which is on host remote3. Host
# remote3 uses a different directory structure from host local.
STANDBY_ARCHIVE_DEST = /fs3/arc_dest/
LOG_ARCHIVE_DEST = /fs3/arc_dest/ # the location is the same as
                                  # STANDBY_ARCHIVE_DEST

LOG_ARCHIVE_FORMAT = log%t_%s.arc
DBS_FILE_NAME_CONVERT = /oracle, /fs3/oracle
LOG_FILE_NAME_CONVERT = /oracle, /fs3/oracle
```

## リカバリの準備としてのスタンバイ・インスタンスの起動

スタンバイ・データベースの準備の最終段階は、手動リカバリまたは管理リカバリを開始できるように、スタンバイ・データベース・インスタンスを起動する段階です。この段階では次の3つの作業を行います。

1. [スタンバイ・インスタンスの起動](#)
2. [変換パラメータで取り込めないスタンバイのファイルの手動での改名](#)
3. [初期化パラメータの設定値の変更の有効化](#)

## スタンバイ・インスタンスの起動

必要なすべてのパラメータおよびネットワーク・ファイルを構成した後、スタンバイ・インスタンスを起動できます。インスタンスが起動されていない場合、スタンバイ・データベースは、プライマリ・データベースによってスタンバイ・サイトに自動的に送信されたアーカイブ REDO ログを受け取ることができません。

### スタンバイ・インスタンスの起動方法

1. SQL\*Plus を使用してスタンバイ・データベース・インスタンスに接続します。たとえば、次のように入力します。

```
SQL> CONNECT sys/change_on_install@standby1
```

2. データベースをマウントせずに、スタンバイ・データベースで Oracle インスタンスを起動します。たとえば、次のように入力します。

```
SQL> STARTUP NOMOUNT pfile=initSTANDBY.ora;
```

3. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

## 変換パラメータで取り込めないスタンバイのファイルの手動での改名

スタンバイの制御ファイル内で、変換パラメータによってプライマリのすべてのデータ・ファイルおよび REDO ログ・ファイルを改名できないことがあります。たとえば、次の表のように改名する予定の次のようなデータ・ファイルがデータベースに含まれているとします。

プライマリのファイル名	スタンバイのファイル名
/oracle/dbs/df1.f	/standby/df1.f
/oracle/dbs/df2.f	/standby/df2.f
/data/df3.f	/standby/df3.f

最初の 2 つのデータ・ファイルのファイル名を変換するために、DB\_FILE\_NAME\_CONVERT を次のように設定できます。

```
DB_FILE_NAME_CONVERT = '/oracle/dbs', '/standby'
```

ただし、このパラメータでは、/data/df3.f は改名できません。このデータ・ファイルは、次のような SQL 文を発行し、スタンバイ・データベースの制御ファイルによって、このファイルを手動で改名する必要があります。

```
SQL> ALTER DATABASE RENAME FILE '/data/df3.f' to '/standby/df3.f';
```

## データ・ファイルの手動の改名

1. SQL\*Plus を使用してスタンバイ・インスタンスを起動してから（まだ起動していない場合）、データベースをマウントします。

```
SQL> STARTUP NOMOUNT pfile=initSTANDBY1.ora;  
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

2. 改名が必要な各データ・ファイルに対して ALTER DATABASE 文を発行します。この *old\_name* は制御ファイル内に記録されているデータ・ファイルの古い名前であり、*new\_name* はスタンバイの制御ファイル内に記録するデータ・ファイルの新しい名前です。

```
SQL> ALTER DATABASE RENAME FILE 'old_name' TO 'new_name';
```

DB\_FILE\_NAME\_CONVERT パラメータで取り込めないすべてのデータ・ファイルを手動で改名することにより、スタンバイの制御ファイルはリカバリ処理時のログ・ストリームを正しく解析できるようになります。

## 初期化パラメータの設定値の変更の有効化

スタンバイ・サイトにアーカイブするようにプライマリの初期化パラメータ・ファイルを構成した場合には、スタンバイ・インスタンスおよびスタンバイ・サイトのリスナーを起動した後で、これらの新しいパラメータの設定値を有効にしておく必要があります。

データベースのオープン中でもプライマリ・データベースの初期化パラメータ・ファイルの変更を行うことはできますが、これらの変更は、インスタンスが再起動されるまでは有効になりません。データベースをオープンしている場合に、データベースを再起動しない場合は、ALTER SYSTEM 文を使用し、パラメータの変更を動的に有効にします。

たとえば、データベースのオープン中に、初期化パラメータ・ファイルで次のような変更を行ったとします。

```
LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest/ MANDATORY REOPEN=2";  
LOG_ARCHIVE_DEST_2="SERVICE=stby1 MANDATORY REOPEN=2";  
LOG_ARCHIVE_DEST_STATE_1=ENABLE;  
LOG_ARCHIVE_DEST_STATE_2=ENABLE;  
LOG_ARCHIVE_MIN_SUCCEED_DEST=2;
```

その後、SQL\*Plus を使用してプライマリ・データベースに接続し、ALTER SYSTEM 文を次のように発行し、これらの設定値を有効にすることができます。

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_1="LOCATION=/arc_dest/ MANDATORY REOPEN=2";  
ALTER SYSTEM SET LOG_ARCHIVE_DEST_2="SERVICE=stby1 MANDATORY REOPEN=2";  
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_1=ENABLE;  
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;  
ALTER SYSTEM SET LOG_ARCHIVE_MIN_SUCCEED_DEST=2;
```

これらの変更が有効になると、プライマリ・データベースはスタンバイ・データベースへの REDO ログのアーカイブを開始できます。

## スタンバイ・データベース構成のトラブルシューティング

スタンバイ・データベースの準備中に発生する問題には、次の状態が考えられます。

- スタンバイ・サイトはプライマリ・データベースによってアーカイブされたログを受け取っていない場合
- スタンバイ・データベースをマウントできない場合

### スタンバイ・サイトはプライマリ・データベースによってアーカイブされたログを受け取っていない場合

スタンバイ・サイトがログを受け取っていない場合は、最初に V\$ARCHIVE\_DEST ビューを問い合わせ、プライマリ・データベースのアーカイブ状況に関する情報を入手する必要があります。特に、エラー・メッセージを調べてください。たとえば、次のように入力します。

```
SQL> SELECT dest_id "ID",
2> status "DB_status",
3> destination "Archive_dest",
4> error "Error"
5> FROM v$archive_dest;
```

ID	DB_status	Archive_dest	Error
1	VALID	/vobs/oracle/work/arc_dest/arc	
2	ERROR	standby1	ORA-16012: Archivelog standby database identifier mismatch
3	INACTIVE		
4	INACTIVE		
5	INACTIVE		

5 rows selected.

問合せの出力では情報が不十分な場合は、考えられる問題点について次のリストを確認してください。次の条件に当てはまるものがある場合は、プライマリ・データベースからスタンバイ・サイトへのアーカイブは失敗します。

- スタンバイ・インスタンスのサービス名が、プライマリ・サイトの tnsnames.ora ファイル内で正しく構成されていません。
- プライマリの初期化パラメータ・ファイルの LOG\_ARCHIVE\_DEST\_*n* パラメータにリストされているサービス名が正しくありません。
- スタンバイのアーカイブ先の状態を指定する LOG\_ARCHIVE\_DEST\_STATE\_*n* パラメータの値が DEFER になっています。
- スタンバイ・サイトで listener.ora が正しく構成されていません。
- リスナーが起動していません。

- スタンバイ・インスタンスが起動していません。
- プライマリの初期化パラメータ・ファイルにスタンバイのアーカイブ先が追加されていますが、変更が有効になっていません。
- スタンバイ・データベースのベースとして無効なバックアップが使用されています（たとえば、誤ったデータベースのバックアップを使用している、またはスタンバイの制御ファイルが正しい方法で作成されていない）。

## スタンバイ・データベースをマウントできない場合

次の条件に当てはまるものがある場合は、スタンバイ・データベースをマウントできません。

- スタンバイ・インスタンスが NOMOUNT モードで起動されていません。最初にインスタンスを起動し、その後データベースをマウントする必要があります。
- スタンバイの制御ファイルが ALTER DATABASE CREATE STANDBY CONTROLFILE ... 文を使用して作成されていません。次の種類の制御ファイル・バックアップは使用できません。
  - RMAN 作成のバックアップ
  - オペレーティング・システム作成のバックアップ
  - STANDBY オプションを指定しない ALTER DATABASE 文を使用して作成されたバックアップ

---

## スタンバイ・データベースの管理

この章では、スタンバイ・データベースの管理方法について説明します。この章は次のトピックから構成されています。

- スタンバイ・データベースのモードの選択
- スタンバイ・データベースの手動リカバリ・モード設定
- 管理リカバリ開始前のギャップ・シーケンスの解決
- スタンバイ・データベースの管理リカバリ・モード設定
- 読取り専用モードでスタンバイ・データベースをオープン
- スタンバイ・データベースのアクティブ化
- Oracle Parallel Server 構成でのスタンバイ・データベースの使用

## スタンバイ・データベースのモードの選択

1-7 ページの「[スタンバイ・データベースのモード](#)」で説明したように、スタンバイ・データベースは次のモードで実行できますが、複数のモードを同時に実行することはできません。

- 管理リカバリ・モード
- 手動リカバリ・モード
- 読取り専用モード

これらのモードで実行する前に、スタンバイ・インスタンスを起動してから、データベースをマウントする必要があります。この章のこの項では、様々なモードを開始するための手順と、スタンバイ・データベースへのフェイルオーバーを実行するための手順を説明します。

## スタンバイ環境の典型的なモード

一般的に、スタンバイ・データベースを作成する理由には、次のようなものがあります。

- プライマリ・データベースの完全破壊に対する保護
- プライマリ・データベースのアプリケーションの破損に対する保護
- プライマリ・データベースに含まれるデータの補足的レポート

データの喪失または破損に対して最大限の保護策を講じるには、管理スタンバイ環境の管理リカバリ・モードでスタンバイ・データベースを使用してください。このセットアップでは、プライマリ・データベースはスタンバイ・サイトにログをアーカイブし、スタンバイ・データベースは自動的にこれらのログを適用します。

---

**注意：** 管理リカバリの前に、スタンバイ・データベースに手動でアーカイブ・ログを適用する必要がある場合があります。この理由については、3-5 ページの「[管理リカバリ開始前のギャップ・シーケンスの解決](#)」を参照してください。

---

スタンバイ・データベースをレポート用に使用する場合は、管理スタンバイ環境で読取り専用モードでデータベースをオープンしてください。スタンバイ・データベースがこのモードの場合、Oracle はアーカイブ REDO ログを適用できませんが、ユーザーからデータベースへの問合せは実行できます。プライマリ・データベースは、スタンバイ・インスタンスが起動されている限り、スタンバイ・サイトへのアーカイブを続けます。

管理リカバリ・モードと読取り専用モードは簡単に切替え可能です。管理スタンバイ環境のほとんどのシステムでは、次の作業のために様々な状況でこのような切替えを行う必要があります。



- 主にレポート用に使用されるスタンバイ・データベースの更新
- 主に災害時保護に使用されるデータベースにデータが適切に適用されていることの確認

**関連項目：** 管理リカバリの開始方法の詳細は、3-13 ページの「[スタンバイ・データベースの管理リカバリ・モード設定](#)」を参照してください。スタンバイ・データベースを読取り専用モードでオープンする方法については、3-16 ページの「[読取り専用モードでスタンバイ・データベースをオープン](#)」を参照してください。

## 手動リカバリが必要な状況

次のような状況では手動リカバリ・モードが必要になります。

管理スタンバイ環境	手動リカバリの理由
はい	ギャップ・シーケンスを解決する必要があります。つまり、スタンバイ・データベースを、管理リカバリを開始できる状態にする必要があります（3-5 ページの「 <a href="#">管理リカバリ開始前のギャップ・シーケンスの解決</a> 」を参照）。
いいえ	管理リカバリはオプションではないため（1-21 ページの「 <a href="#">非管理リカバリ環境</a> 」を参照）、スタンバイ・サイトにログを手動で転送し、スタンバイ・データベースにログを手動で適用する必要があります。

このため、管理スタンバイ環境をインプリメントした場合でも、スタンバイ・データベースで手動リカバリを実行する必要があることがあります。

**関連項目：** スタンバイ・データベースの手動リカバリの実行方法については、3-3 ページの「[スタンバイ・データベースの手動リカバリ・モード設定](#)」を参照してください。

## スタンバイ・データベースの手動リカバリ・モード設定

スタンバイ・データベースを起動し、マウントした後、データベースを手動リカバリ・モードに設定できます。スタンバイ・データベースを最新の状態にしておくためには、手動でプライマリ・データベースからスタンバイ・データベースにアーカイブ REDO ログを適用する必要があります。

アーカイブ・ログは次の方法のいずれかでスタンバイ・サイトに到着します。

- プライマリ・データベースによるログの自動アーカイブ（管理スタンバイ環境をインプリメントしてある場合のみ）。
- オペレーティング・システムのユーティリティまたはその他の方法による、ログの手動転送。

スタンバイ・データベースは、スタンバイの初期化パラメータ・ファイル内の次のパラメータによって、指定された場所にアーカイブ・ログ・ファイル・グループが入っているものと想定します。

- LOG\_ARCHIVE\_DEST\_*n* (この *n* は 1 から 5 までの整数) によって指定された最初の有効なディスク位置
- LOG\_ARCHIVE\_DEST によって指定された場所

初期化パラメータ・ファイル内で指定された場所にアーカイブ・ログがない場合には、RECOVER 文の FROM オプションを使用して、代替の場所を指定できます。

### スタンバイ・データベースの手動リカバリ・モード設定

1. SQL\*Plus を使用してスタンバイ・インスタンスに接続してから、スタンバイ・データベースで Oracle インスタンスを起動します。たとえば、次のように入力します。

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
```

2. スタンバイ・データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. Oracle がスタンバイ・サイトにログを自動的にアーカイブしていない場合は、バイナリ・データの転送のための適切なオペレーティング・システムのユーティリティを使用し、スタンバイ・ホスト上の任意の場所にログを手動で転送してください。たとえば、次のように入力します。

```
% cp /oracle/arc_dest/*.arc /standby/arc_dest
```

4. RECOVER 文を発行し、スタンバイ・データベースを手動リカバリ・モードにします。

---

---

**注意：** FROM 'location' オプションは、アーカイブ・ログ・グループが、スタンバイの初期化パラメータ・ファイル内の LOG\_ARCHIVE\_DEST\_*n* (この *n* は 1 から 5 まで整数) または LOG\_ARCHIVE\_DEST パラメータで指定された場所がない場合のみ指定してください。

---

---

たとえば、次の文のいずれかを実行します。

```
RECOVER STANDBY DATABASE # uses location for logs specified in
                           # initialization parameter file
RECOVER FROM '/logs' STANDBY DATABASE # specifies nondefault location
```

スタンバイ・データベースを最新の状態に保つためには、Oracle からアーカイブ REDO ログが生成されるたびに、ログの転送および適用を継続する必要があります。

## 管理リカバリ開始前のギャップ・シーケンスの解決

ギャップ・シーケンスは、スタンバイ・データベースで管理リカバリ・モードを開始するために必要となるアーカイブ REDO ログの範囲です。プライマリ・データベースで生成された次のアーカイブ・ログを、管理リカバリ・モードのスタンバイ・データベースに適用できる場合は、スタンバイ・データベースで管理リカバリを開始できます。この項は、次のトピックから構成されています。

- [ギャップ・シーケンスの原因](#)
- [ギャップ・シーケンスが存在するかどうかの確認](#)
- [ギャップ・シーケンス内のログのスタンバイ・サイトへの転送](#)
- [ギャップ・シーケンス内のログのスタンバイ・データベースへの適用](#)

## ギャップ・シーケンスの原因

ギャップ・シーケンスは、プライマリ・データベースがログをアーカイブし、そのログがスタンバイ・サイトに転送されなかったときに発生する可能性があります。スタンバイ・データベースでは REDO ログを順次に適用する必要があるため、ログが欠落していると、管理リカバリで後続のログを適用できなくなります。

ギャップ・シーケンスは次の状況で発生する可能性があります。

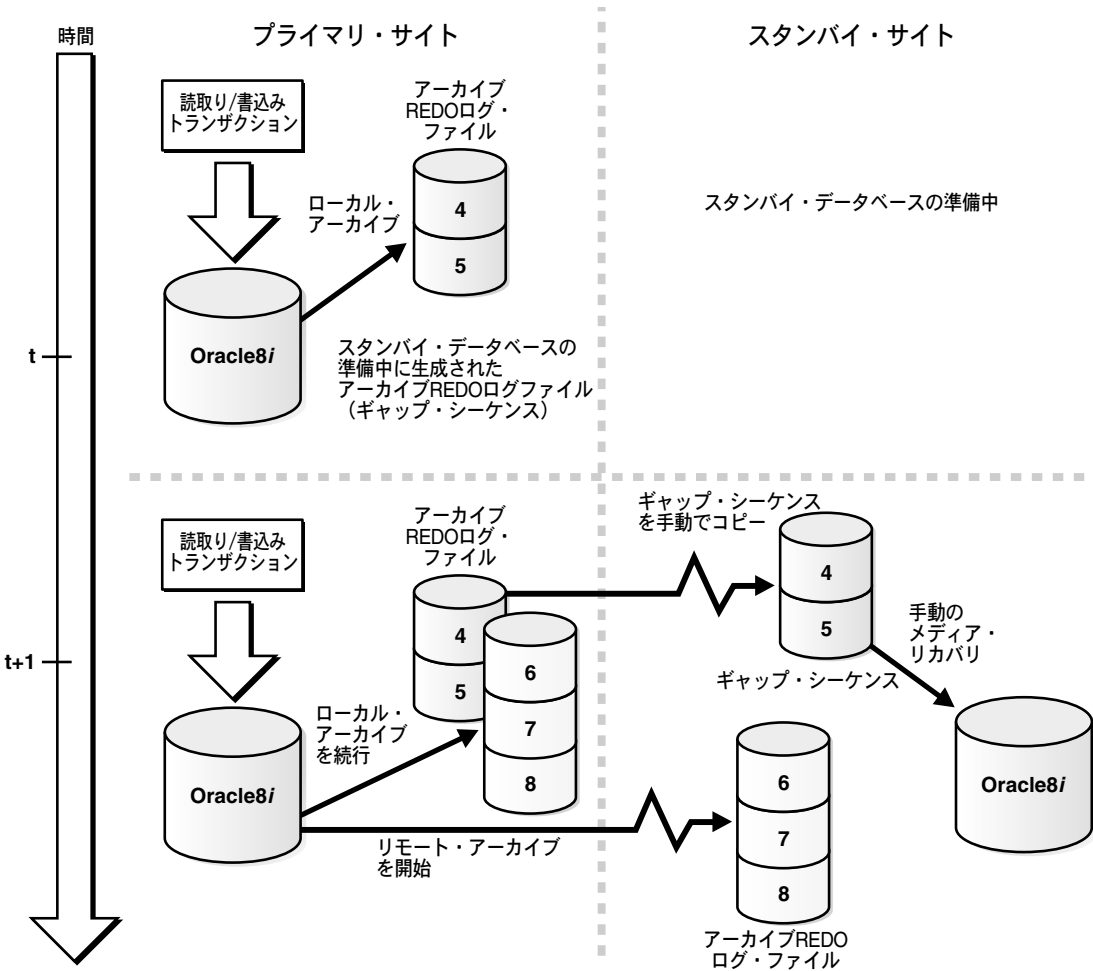
- [スタンバイ・データベースの作成](#)
- [プライマリ・データベースのオープン中のスタンバイ・データベースのシャットダウン](#)
- [スタンバイ・サイトへのログの転送を阻害するネットワーク障害](#)

## スタンバイ・データベースの作成

1つの例として、ギャップ・シーケンスは古いバックアップからスタンバイ・データベースを作成したときに発生します。たとえば、スタンバイ・データベースがログ 100 までの変更を含むバックアップから作成されている場合、プライマリ・データベースにログ 150 までの変更が現在含まれているとすると、スタンバイ・データベースで管理リカバリを開始する前に、ログ 101 から 150 を手動で適用する必要があります。

もう1つの典型的な例として、オープン中のデータベースのホット・バックアップからスタンバイ・データベースを生成したときにも、ギャップ・シーケンスが発生します。たとえば、[図 3-1](#) のような状況を想定してください。

図 3-1 ギャップ・シーケンス内のアーカイブ・ログの手動リカバリ



次のような作業を行います。

1. データベース PRIMARY のホット・バックアップを作成します。
2. ネットワーク・ファイルの構成作業中の  $t$  の時点で、PRIMARY はログ 4 および 5 をアーカイブします。
3.  $t+1$  の時点でスタンバイ・インスタンスを起動します。

4. PRIMARY はログ 6、7 および 8 をプライマリ・サイトおよびスタンバイ・サイトの両方にアーカイブします。

アーカイブ・ログ 4 および 5 はギャップ・シーケンスの一部となるため、管理リカバリによってアーカイブ・ログ 6、7 および 8 をスタンバイ・データベースに適用する前に、スタンバイ・データベースに手動で適用する必要があります。

## プライマリ・データベースのオープン中のスタンバイ・データベースのシャットダウン

メンテナンス上の問題を解決するために、スタンバイ・データベースをシャットダウンする必要がある場合があります。たとえば、次の場合はスタンバイ・データベースをシャットダウンする必要があります。

- プライマリ・データベースで、MAXDATAFILE などの制御ファイル・パラメータを変更した場合
- ALTER DATABASE CLEAR UNARCHIVED LOGFILES 文を発行した場合

---

**注意：** プライマリ・データベースで RESETLOGS 操作を実行すると、スタンバイ・データベースは無効になります。プライマリ・データベースのログをリセットした場合は、スタンバイ・データベースを再構築する必要があります。

---

ギャップ・シーケンスが作成されないように、次の規則を守ってください。

- プライマリ・データベースを起動する前にスタンバイ・データベースおよびリスナーを起動する
- スタンバイ・データベースをシャットダウンする前にプライマリ・データベースをシャットダウンする

前述の 2 つの規則のいずれかに違反した場合、プライマリ・データベースがオープンし、アーカイブしている間、スタンバイ・データベースがダウンします。このため、ギャップ・シーケンスが作成される可能性があります。その後スタンバイ・データベースを再起動したときには、管理リカバリを開始する前に、スタンバイ・データベースを手動でプライマリ・データベースと同期させる必要があります。

---

**注意：** プライマリの初期化パラメータ・ファイルの LOG\_ARCHIVE\_DEST\_# パラメータの 1 つで、スタンバイ・サイトが MANDATORY として指定されている場合には (2-17 ページの「[必須およびオプションのアーカイブ先の指定](#)」を参照)、スタンバイ・データベースをシャットダウンする前に、動的に OPTIONAL に変更してください。そうしておかないと、プライマリ・データベースはオンライン REDO ログをアーカイブできないため、最終的には停止します。

---

## スタンバイ・サイトへのログの転送を阻害するネットワーク障害

管理スタンバイ環境の場合にネットワークがダウンすると、プライマリ・データベースはディスクへのアーカイブは続行できますが、スタンバイ・サイトへのアーカイブはできなくなります。この状況では、アーカイブ・ログはプライマリ・サイトに通常どおりに蓄積されますが、スタンバイ・インスタンスはこれらのログを認識しません。

この問題を防止するために、スタンバイの宛先の状態を必須として指定できます。アーカイブ先が必須の場合、プライマリ・データベースはスタンバイ・サイトにアーカイブできるようになるまでは、ログのアーカイブを行いません。たとえば、STANDBY1 を必須のアーカイブ先に構成するために、プライマリの初期化パラメータ・ファイルで次のように設定します。

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 MANDATORY'
```

このように構成した場合、ネットワークの問題を解決しない限り、プライマリ・データベースはアーカイブされていないオンライン REDO ログにスイッチできないため、最終的には停止することになります。プライマリ・データベースの中にオンライン REDO ログが 2 つしかない場合には、この問題は深刻です。

**関連項目：** スタンバイのアーカイブの OPTIONAL および MANDATORY キーワードの重要性の説明は、『Oracle8i 管理者ガイド』のアーカイブ REDO ログに関する章を参照してください。追加情報は 5-39 ページの「シナリオ 7: ネットワーク障害後のリカバリ」を参照してください。

## ギャップ・シーケンスが存在するかどうかの確認

ギャップ・シーケンスが存在しているかどうかを確認するには、次の手順で SQL スクリプトを実行します。ギャップ・シーケンスが存在する場合には、問合せの出力に、ギャップ・シーケンス内のすべてのログのスレッド番号および順序番号が指定されています。特定のスレッドにギャップ・シーケンスがない場合には、問合せから行が戻されないか、LowSeq# および HighSeq# 列の中に同じ番号が戻されます。

### ギャップ・シーケンス内のログの識別

1. SQL\*Plus を起動し、スタンバイ・データベースをマウントします。

```
SQL> CONNECT sys/sys_pwd@standby1 AS SYSDBA
SQL> STARTUP NOMOUNT pfile=/oracle/admin/pfile/init.ora
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

2. 次の SELECT 文をスクリプトにコピーし、スタンバイ・データベースでこのスクリプトを実行します。

```
SELECT high.thread#, "LowGap#", "HighGap#"
FROM
(
  SELECT thread#, MIN(sequence#)-1 "HighGap#"

```

```

FROM
(
  SELECT a.thread#, a.sequence#
  FROM
  (
    SELECT *
    FROM v$log_history
  ) a,
  (
    SELECT thread#, MAX(next_change#) gap1
    FROM v$log_history
    GROUP BY thread#
  ) b
  WHERE a.thread# = b.thread#
  AND a.next_change# > gap1
)
GROUP BY thread#
) high,

(
  SELECT thread#, MIN(sequence#) "LowGap#"
  FROM
  (
    SELECT thread#, sequence#
    FROM v$log_history, v$datafile
    WHERE checkpoint_change# <= next_change#
    AND checkpoint_change# >= first_change#
  )
  GROUP BY thread#
) low
WHERE low.thread# = high.thread#;

```

3. 問合せの出力でギャップ・シーケンスを判別します。たとえば、次のような出力が考えられます。

```

SQL> @gap
THREAD#      LowSeq#      HighSeq#
-----
1           460           463
2           202           204
3           100           100

```

すべてのスレッドにギャップ・シーケンスがあるわけではありません。この例が示すように、スレッド3のLowSeq#とHighSeq#は同じであるため、このスレッドにはギャップ・シーケンスはありません。

スタンバイ・データベースで管理リカバリの準備をするために、各スレッドのギャップ・シーケンス内のログをスタンバイ・データベースに適用する必要があります。

**関連項目：** 手動リカバリの実行方法の詳細は、3-3 ページの「[スタンバイ・データベースの手動リカバリ・モード設定](#)」を参照してください。

## ギャップ・シーケンス内のログのスタンバイ・サイトへの転送

ギャップ・シーケンス内のログ順序番号が判明した後、プライマリ・サイトで V\$ARCHIVED\_LOG ビューを問い合わせ、そのファイル名を入手できます（4-5 ページの「[スタンバイ・サイトで受け取ったアーカイブ・ログの判別](#)」を参照）。スタンバイ・サイトのアーカイブ・ログのファイル名は、スタンバイの初期化パラメータ・ファイル内の STANDBY\_ARCHIVE\_DEST および LOG\_ARCHIVE\_FORMAT パラメータによって生成されます。

スタンバイ・データベースがプライマリ・データベースと同じホスト上にある場合、またはスタンバイ・データベースがプライマリ・データベースとは異なるディレクトリ構造を持つリモート・ホスト上にある場合には、スタンバイ・サイトのログのファイル名を、プライマリ・データベースでアーカイブされたログのファイル名と同じにすることはできません。スタンバイ・サイトにアーカイブ・ログを転送する前に、スタンバイ・サイトでのログの正しいファイル名を判別してください。

### ギャップ・シーケンス内のログのスタンバイ・サイトへの転送

1. 先に入手しておいたギャップ・シーケンスのリストを確認します。たとえば、次のギャップ・シーケンスが発生したものとします。

THREAD#	LowSeq#	HighSeq#
-----	-----	-----
1	460	463
2	202	204
3	100	100

スレッド 3 にはギャップ・シーケンスはないため、スレッド 1 および 2 のログを転送するのみで済みます。

2. プライマリ・データベースでアーカイブされた、ギャップ・シーケンス内のログのファイル名を判別します。たとえば、プライマリ・データベースに接続した後、SQL\*Plus を使用し、SQL 問合せを発行し、各スレッド中のログの名前を入手します。

```
SQL> CONNECT sys/sys_pwd@primary
SQL> SELECT name
2> FROM v$archived_log
3> WHERE sequence# in (460, 202);

NAME
-----
/primary/thread1_dest/arcr_1_460.arc
```



```
/primary/thread2_dest/arcr_2_202.arc
2 rows selected.
```

3. スタンバイの初期化パラメータ・ファイル内の STANDBY\_ARCHIVE\_DEST および LOG\_ARCHIVE\_FORMAT の設定値を確認します。たとえば、次のように指定されているとします。

```
STANDBY_ARCHIVE_DEST = /standby/arc_dest/
LOG_ARCHIVE_FORMAT = log_%t_%s.arc
```

これらのパラメータの設定値は、スタンバイ・サイトでのアーカイブ REDO ログのファイル名を決定するものです。

4. STANDBY\_ARCHIVE\_DEST および LOG\_ARCHIVE\_FORMAT の値に従って改名し、プライマリ・サイトからスタンバイ・サイトにギャップ・シーケンスのログを転送します。たとえば、次のように入力します。

```
% cp /primary/thread1_dest/arcr_1_460.arc /standby/arc_dest/log_1_460.arc
% cp /primary/thread1_dest/arcr_1_461.arc /standby/arc_dest/log_1_461.arc
% cp /primary/thread1_dest/arcr_1_462.arc /standby/arc_dest/log_1_462.arc
% cp /primary/thread1_dest/arcr_1_463.arc /standby/arc_dest/log_1_463.arc

% cp /primary/thread1_dest/arcr_2_202.arc /standby/arc_dest/log_2_202.arc
% cp /primary/thread1_dest/arcr_2_203.arc /standby/arc_dest/log_2_203.arc
% cp /primary/thread1_dest/arcr_2_204.arc /standby/arc_dest/log_2_204.arc
```

5. LOG\_ARCHIVE\_DEST と STANDBY\_ARCHIVE\_DEST のパラメータ値が同じでない場合には、STANDBY\_ARCHIVE\_DEST ディレクトリから LOG\_ARCHIVE\_DEST ディレクトリにギャップ・シーケンスのログをコピーします。これらのパラメータ値が同じ場合には、このステップを実行する必要はありません。

たとえば、スタンバイの初期化パラメータが次のように設定されているとします。

```
STANDBY_ARCHIVE_DEST = /standby/arc_dest/
LOG_ARCHIVE_DEST = /log_dest/
```

パラメータ値が異なるため、LOG\_ARCHIVE\_DEST の場所にアーカイブ・ログをコピーします。

```
% cp /standby/arc_dest/* /log_dest/
```

手動リカバリが開始されると、Oracle は LOG\_ARCHIVE\_DEST の値を調べて、ログの場所を確認します。

必要なすべてのログが STANDBY\_ARCHIVE\_DEST ディレクトリに入っているため、スタンバイ・データベースにギャップ・シーケンスのログを適用するという、次の段階に進むことができます。

## ギャップ・シーケンス内のログのスタンバイ・データベースへの適用

ギャップ・シーケンス内のログをスタンバイ・サイトに転送した後、RECOVER AUTOMATIC 文を使用してログを適用できます。ログの適用後、スタンバイ・データベースの管理リカバリ・モードを開始できます。

### ギャップ・シーケンス内のアーカイブ REDO ログの適用

1. SQL\*Plus を起動し、スタンバイ・データベースをマウントします（まだマウントされていない場合）。たとえば、次のように入力します。

```
SQL> CONNECT sys/sys_pwd@standby1 AS SYSDBA
SQL> STARTUP NOMOUNT pfile=/oracle/admin/pfile/initSTBY.ora
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

2. AUTOMATIC オプションを使用してデータベースのリカバリを行います。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE
```

使用可能なログのリカバリ後に、Oracle は存在しないログの名前のプロンプトを表示します。これは、リカバリ処理が、プライマリ・データベースによってスタンバイ・サイトにアーカイブされたログを認識できないためです。たとえば、次のように表示されることがあります。

```
ORA-00308: アーカイブ・ログ '/oracle/standby/standby_logs/arcr_1_540.arc' をオープンできません。
ORA-27037: ファイル・ステータスを取得できません。
SVR4 Error: 2: No such file or directory
Additional information: 3
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

3. Oracle が使用可能なログを適用した後、次の文を実行し、リカバリの取消しを行います（または [Ctrl] を押しながら [C] を押す）。

```
SQL> CANCEL
Media recovery cancelled.
```

リカバリの取消し後の次のエラー・メッセージは許容範囲であり、問題を示すものではありません。

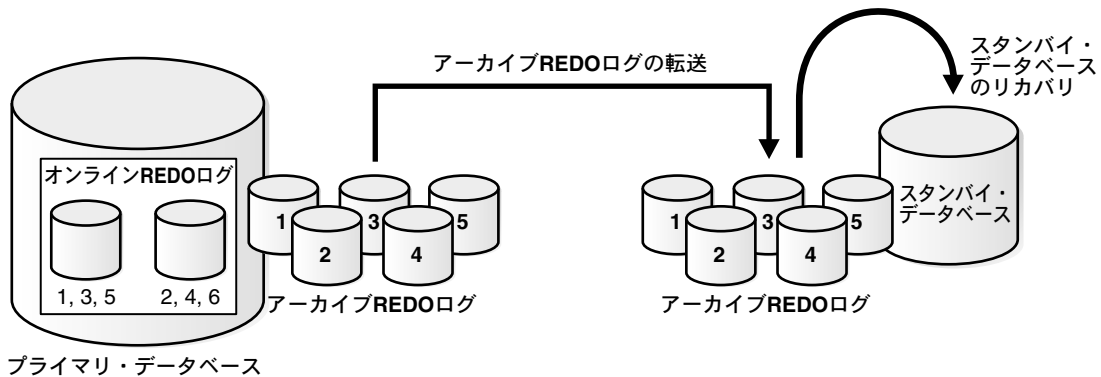
```
ORA-01547: 警告: RECOVER は成功しましたが OPEN RESETLOGS が次のエラーを受け取りました。
ORA-01194: ファイル 1 は一貫した状態にするためにさらにリカバリが必要です。
ORA-01110: データ・ファイル 1: 'some_filename'
ORA-01112: メディアのリカバリ処理が開始されていません。
```

これで、管理リカバリを開始できることになります。

## スタンバイ・データベースの管理リカバリ・モード設定

管理スタンバイ環境をインプリメントした場合には、ローカルまたはリモートのホストへのアーカイブを自動化できます。図 3-2 に示すように、Oracle はプライマリ・データベースからのアーカイブ・ログの到着を待ち、スタンバイ・データベースに自動的にログを適用することにより、スタンバイ・データベースとプライマリ・データベースの同期を保ちます。この機能により、アーカイブ・ログのファイル名を指定して、対話式にリカバリ処理を行う必要はなくなります。

図 3-2 スタンバイ・データベースへのアーカイブ REDO ログの転送および適用



## 管理リカバリ・モードの開始

管理リカバリを使用可能にするには、次の SQL\*Plus 文を使用します。

```
RECOVER MANAGED STANDBY DATABASE
```

RECOVER 文の TIMEOUT オプションを使用すると、オプションでタイムアウト間隔を指定できます。この場合、管理リカバリ操作では、指定した分数待機した後、要求されたアーカイブ・ログ・エントリを、Oracle がスタンバイの制御ファイルのディレクトリに書き込みます。

スタンバイの制御ファイル内で必要な次のログ・エントリを検出できないために Oracle がタイムアウトした場合には、システムは適切なメッセージを発行し、管理リカバリ・モードを終了します。デフォルトでは、管理リカバリ操作が次のアーカイブ・ログを待つ時間は無限であり、この操作は CANCEL 文（または [Ctrl]+[C] キーの組合せ）、シャットダウンまたはクラッシュによってのみ終了します。

### スタンバイ・データベースの管理リカバリ・モード設定

1. スタンバイ・データベースに接続した後、必要であればパラメータ・ファイルを指定し、スタンバイ・データベースをマウントせずに起動します。

```
STARTUP NOMOUNT pfile=initSTANDBY.ora
```

2. データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. スタンバイ・データベースが同期していること、つまり、ギャップ・シーケンスがないことを確認します。ギャップ・シーケンスが存在するかどうかを確認するには、3-8 ページの「[ギャップ・シーケンスが存在するかどうかの確認](#)」を参照してください。ギャップ・シーケンスがある場合には、手動でデータベースを同期化します。ギャップ・シーケンスがない場合は、次のステップに進んでください。
4. スタンバイ・データベースを管理リカバリ・モードにします。

```
RECOVER MANAGED STANDBY DATABASE
```

---

---

**注意：** RECOVER MANAGED STANDBY DATABASE 文では、FROM 'location' オプションは使用できません。

---

---

オプションのタイムアウト・オプションを使用する場合は、コマンド構文に TIMEOUT *integer* を追加してください。

```
RECOVER MANAGED STANDBY DATABASE TIMEOUT 60
```

これにより、Oracle は管理リカバリを開始します。プライマリ・データベースからスタンバイ・サイトに REDO ログがアーカイブされるたびに、スタンバイ・データベースはログを自動的に適用します。

---

---

**注意：** RECOVER 文を実行した後、RECOVER 文の次にプロンプトが表示されますが、これは予想された動作です。

---

---

### 管理リカバリで転送されたアーカイブ REDO ログが適用されていることを迅速に確認

2-33 ページの「[スタンバイ・データベース構成のトラブルシューティング](#)」で説明したように、アーカイブ REDO ログの状態を判別するにはいくつかの方法があります。次の手順では、プライマリ・サイトで強制的にアーカイブすることにより、スタンバイ・データベースでビューを問い合せて、アーカイブ・ログが適用されたかどうかを確認できます。

1. プライマリ・データベースに接続し、データベースがオープンされていることを確認します。

```
% sqlplus sys/sys_pwd@primary
SQL> SELECT status FROM v$instance;
```

```
STATUS
-----
OPEN
```

2. 現行のオンライン REDO ログをアーカイブします。

```
ALTER SYSTEM ARCHIVE LOG CURRENT;
System altered.
```

3. プライマリ・データベースを問い合せて、一番新しくアーカイブされた REDO ログを確認します。

```
SQL> SELECT max(sequence#) FROM v$log_history;
```

```
MAX(SEQUENCE#)
-----
541
```

4. スタンバイ・インスタンスで新しいセッションを作成し、V\$LOG\_HISTORY ビューを問い合せます。

```
% sqlplus sys/sys_pwd@standby1
SQL> SELECT max(sequence#) FROM v$log_history;
```

```
MAX(SEQUENCE#)
-----
541
```

順序番号は、プライマリ・サイトでの番号と同じである必要があります。番号が同じでない場合は、Oracle がログの受取りおよび適用を終了するまでしばらく待ってから、再び問い合せてください。それでも Oracle がログを適用しない場合は、2-33 ページの「スタンバイ・データベース構成のトラブルシューティング」でトラブルシューティング情報の入手方法を確認してください。

## 管理リカバリの取消し

管理リカバリ操作は、いずれかの SQL\*Plus 文を発行することにより、いつでも取消しできます。

```
RECOVER MANAGED STANDBY DATABASE CANCEL
RECOVER MANAGED STANDBY DATABASE CANCEL IMMEDIATE
```

最初の文は、管理リカバリ操作が現行の REDO ログを終了するまで待ち、その後リカバリを終了します。しかし、IMMEDIATE オプションを使用した場合には、Oracle は REDO ログからの別のブロックの読取り、または次の REDO ログ・ファイルのオープンのいずれかの事象の前に、管理リカバリ操作を停止します。次の状況を考えてみましょう。

リカバリの取消し	影響
リカバリが次の REDO ログをオープンする前	CANCEL IMMEDIATE は CANCEL と同じです。
スタンバイ・データベースによる REDO ログの処理中	CANCEL IMMEDIATE によりデータベースは非一貫性状態になります。Oracle は非一貫性データベースのオープンは許可しませんが、それでも手動リカバリまたは管理リカバリは開始できます。

## 読取り専用モードでスタンバイ・データベースをオープン

読取り専用モードでは、ユーザーはオンラインのデータ変更を考慮する必要なしに、スタンバイ・データベースのオープンおよび問合せを行うことができます。この機能があるため、スタンバイ・データベースをレポート用に使用することにより、プライマリ・データベースのシステム・オーバーヘッドを削減できます。また、管理スタンバイ・データベースが適切に更新されていることを確認するために、定期的に読取り専用モードでスタンバイ・データベースをオープンすることもできます。

この項は、次のトピックから構成されています。

- [読取り専用モードを実行するかどうかの検討](#)
- [読取り専用モードでのアーカイブ REDO ログの受取り](#)
- [読取り専用モードのデータベースのオープン](#)
- [一時表領域の作成](#)

### 読取り専用モードを実行するかどうかの検討

スタンバイ・データベースを読取り専用モードで実行するかどうかを決定するときには、次のことを考慮してください。

- 主に災害時の保護のためにスタンバイ・データベースをメンテナンスする場合には、情報のソースとして、スタンバイ・データベースに依存しすぎないようにしてください。災害が発生した場合には、スタンバイ・データベースを迅速にアクティブにし、すべてのユーザー・アクティビティを即時に停止する必要があります。
- スタンバイ・データベースを問合せに使用すると、管理リカバリは使用できなくなります。ある時点において、スタンバイ・データベースに対してリカバリ操作を実行し、プライマリ・データベースと再同期させる必要があります。このアクションにより、災害

時のリカバリ・データベースとしてのスタンバイ・データベースの役割は制約を受けます。

災害時の保護およびレポートの両方に使用するためにスタンバイ・データベースが必要な場合には、複数のスタンバイ・データベースをメンテナンスし、いくつかのデータベースは読取り専用モードで実行し、いくつかのデータベースは管理リカバリ・モードで実行することができます。読取り専用データベースでは再同期化が必要になりますが、リカバリ・モードのデータベースにより、災害時の保護が提供されます。

## 読取り専用モードでのアーカイブ REDO ログの受取り

スタンバイ・データベースが読取り専用モードになっている間、サイトではプライマリ・サイトからのアーカイブ REDO ログの受取りを継続できます。ただし Oracle は、管理リカバリの場合のように、これらのログを自動的に適用することはありません。このため、読取り専用のスタンバイ・データベースは、アーカイブ・レベルではプライマリ・データベースと同期していません。すべてのアーカイブ REDO ログが適用されていない限り、フェイルオーバー時にスタンバイ・データベースをアクティブにしないでください。

## 読取り専用モードのデータベースのオープン

スタンバイ・データベースでは次のような状態が考えられます。

- シャットダウン
- 手動リカバリ・モード
- 管理リカバリ・モード
- 読取り専用モード

次の手順を使用して、最初の 3 つの状態のいずれかからでも読取り専用モードに移行または戻すことができます。

### データベースのシャットダウン中にスタンバイ・データベースを読取り専用モードでオープン

1. SQL\*Plus を使用し、スタンバイ・データベースをマウントせずに、Oracle インスタンスを起動します。

```
SQL> STARTUP NOMOUNT pfile=initSTANDBY.ora
```

2. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. 読取り専用モードでデータベースをオープンします。

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

### 手動リカバリ・モードで、スタンバイ・データベースを読取り専用モードでオープン

1. 次のように入力し、リカバリの取消しを行います（アーカイブ REDO ログのフローを終了させ、プロンプトを表示します）。

```
SQL> RECOVER CANCEL
```

2. 読取り専用モードでデータベースをオープンします。

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

### 管理リカバリ・モードで、スタンバイ・データベースを読取り専用モードでオープン

1. SQL\*Plus セッションを起動し、次の文を実行します。

```
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL
```

2. 読取り専用モードでデータベースをオープンします。

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

### スタンバイ・データベースの読取り専用モードから管理リカバリ・モードへの変更

1. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了します。
2. 次の文を発行します。

```
SQL> RECOVER MANAGED STANDBY DATABASE # you can also set the TIMEOUT option
```

### スタンバイ・データベースの読取り専用モードから手動リカバリ・モードへの変更

1. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了します。
2. 次の文を発行します。

```
SQL> RECOVER STANDBY DATABASE # you can also set the TIMEOUT option
```

## 一時表領域の作成

読取り専用のスタンバイ・データベースで問合せを実行するには、Oracle データベース・サーバーがオンディスク・ソート操作を実行する必要があります。しかし、変更によってスタンバイの制御ファイルがプライマリの制御ファイルと異なるものになるため、Oracle がデータ・ディクショナリに書き込むためのソート操作の場所を表領域の中で割り当てることができません。

一時表領域を使用すると、問合せを行うために、読取り専用モードでテンポラリ・ファイル・エントリを追加できます。これにより、ディクショナリ・ファイルに影響を与えたり、REDO エントリを生成することなく、Oracle8i 読取り専用データベースでオンディスク・ソート操作を実行できます。



一時表領域の作成については、次のような要件があります。

- 一時的で、ローカルに管理され、テンポラリ・ファイルのみを含む必要があります。
- プライマリ・データベースで、ローカルに管理される一時表領域の使用に対するユーザー・レベルの割当ておよび許可が必要です。スタンバイ・データベースでこれらの設定値の変更はできません。

また、次のガイドラインにも従ってください。

- 必要な場合に管理リカバリ・モードまたは手動リカバリ・モードに切り替えられるように、スタンバイ・データベースのデータの妥当性チェックの時間は最小限にします。
- レポートのランタイムを最小化します。
- プライマリ・データベースでのみ最適化をインプリメントします。

**関連項目：** テンポラリ・ファイルおよび一時表領域の使用方法的詳細は、『Oracle8i 管理者ガイド』を参照してください。

### 読取り専用スタンバイ・データベースで使用する一時表領域の作成

1. 3-17 ページの「[読取り専用モードのデータベースのオープン](#)」に記載された適切な手順を使用し、スタンバイ・データベースを読取り専用モードでオープンします。
2. 一時表領域を作成します。たとえば、次のように入力します。

```
SQL> CREATE TEMPORARY TABLESPACE tbs_1 TEMPFILE 'file_1.f'  
      EXTENT MANAGEMENT LOCAL UNIFORM SIZE 16M;
```

**関連項目：** CREATE TEMPORARY TABLESPACE 文の詳細は、『Oracle8i SQL リファレンス』を参照してください。

## スタンバイ・データベースのアクティブ化

緊急時以外は、スタンバイ・データベースのアクティブ化またはフェイルオーバーを実行しないでください。スタンバイ・データベースはアクティブにすると、通常の本番データベースになり、スタンバイとしての機能を失います。アクティブ化を取り消してデータベースをスタンバイ・モードに戻すことはできません。

---

**注意：** スタンバイ・データベースが正しく更新されているかどうかを確認するためにスタンバイ・データベースをアクティブにしないでください。かわりに読取り専用モードでオープンしてください。

---

緊急時の状況によっては、プライマリ・データベースのファイルにアクセスできないこともあります。ファイルにアクセスできる場合には、プライマリ・データベースの現行のオンライン REDO ログを手動でアーカイブし、使用可能なすべてのアーカイブ REDO ログをスタンバイ・データベースに転送および適用する必要があります。

### プライマリ・データベースの REDO ログの救済

1. 可能であれば、プライマリ・データベースの現行のオンライン REDO ログをアーカイブします。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

2. 管理スタンバイ環境を使用していない場合には、まだスタンバイ・データベースに適用されていない、使用可能なすべてのアーカイブ・ログを、スタンバイ・サイトに手動で転送します。パイナリ・データの転送に適したオペレーティング・システムのユーティリティを使用してください。たとえば、次のように入力します。

```
% cp /oracle/arc_dest/*.arc /standby/arc_dest
```

3. スタンバイ・データベースが現在、管理リカバリ・モードを使用している場合には、リカバリの取消しを行います。

```
SQL> CANCEL
```

スタンバイ・データベースが現在、管理リカバリ・モードを使用していない場合には、マウント済みの状態にします。たとえば、スタンバイ・インスタンスが起動していない場合には、次のように入力します。

```
SQL> STARTUP NOMOUNT pfile=initSTANDBY.ora  
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

4. スタンバイ・データベースをアクティブにする前に、スタンバイ・データベースの手動リカバリを実行します。ここでは、オプションとして FROM 'location' オプションを指定できます (3-3 ページの「[スタンバイ・データベースの手動リカバリ・モード設定](#)」を参照)。たとえば、次の文のいずれかを入力します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE # uses location for logs listed in  
initialization parameter file  
SQL> RECOVER AUTOMATIC FROM '/logs' STANDBY DATABASE # specifies nondefault  
location
```

この手順を実行すると、スタンバイ・データベースはプライマリ・データベースの障害の直前の時点までロールフォワードされます。現行の REDO ログ以外のすべての REDO ログをスタンバイ・データベースに適用できます。現行以外のオンライン REDO ログを紛失し、これらのログがアーカイブされていない場合には、プライマリ・データベースのアーカイブされていない REDO ログからトランザクションのリカバリを行わずに、スタンバイ・データベースをアクティブにします。

スタンバイ・データベースをアクティブにすると、自動的にオンライン REDO ログはリセットされます。この場合、スタンバイ・データベースの REDO ログとプライマリ・データベースの REDO ログの互換性が失われます。元のプライマリ・データベースからアクティブにされたスタンバイ・データベース、またはアクティブにされたスタンバイ・データベースから元のプライマリ・データベースにアーカイブ REDO ログを適用することはできません。また、スタンバイ・データベースはアクティブにされたときにマウントされていないため、アクティブ化の直後には表およびビューには有用な情報は含まれていません。

## スタンバイ・データベースのアクティブ化

1. 次の問合せを実行し、スタンバイ・データベースが EXCLUSIVE モードで実行されていることを確認します。

```
SQL> SELECT name,value FROM v$parameter WHERE name='parallel_server';
NAME                                VALUE
-----
parallel_server                      FALSE
1 row selected.
```

値が TRUE の場合には、データベースは排他的にマウントされていません。値が FALSE の場合には、データベースは排他的にマウントされています。

2. スタンバイ・データベースをアクティブにします。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

このコマンドは、オンライン REDO ログをリセットするので注意してください。

3. スタンバイ・インスタンスをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE
```

4. 新しい本番データベースのバックアップを、可能な限り早く作成してください。この時点で、以前はスタンバイ・データベースだったものが本番データベースになります。この作業は必須ではありませんが、バックアップがないと、アクティブにした後に変更のリカバリを行うことができないため、安全策として実行するようにお勧めします。

5. 新しい本番インスタンスを、読取り / 書込みまたは読取り専用モードで起動します。

```
SQL> STARTUP MOUNT
SQL> ALTER DATABASE READ ONLY; # opens the database in read-only mode
SQL> ALTER DATABASE READ WRITE; # opens the database in read/write mode
```

---

**注意：** スタンバイ・データベースをアクティブにすると、元の本番データベースの、アーカイブされていないログのすべてのトランザクションは失われます。スタンバイの REDO ログはアクティブ化の時にリセットされるため、アクティブ化の前にアーカイブされたログは適用できません。

---

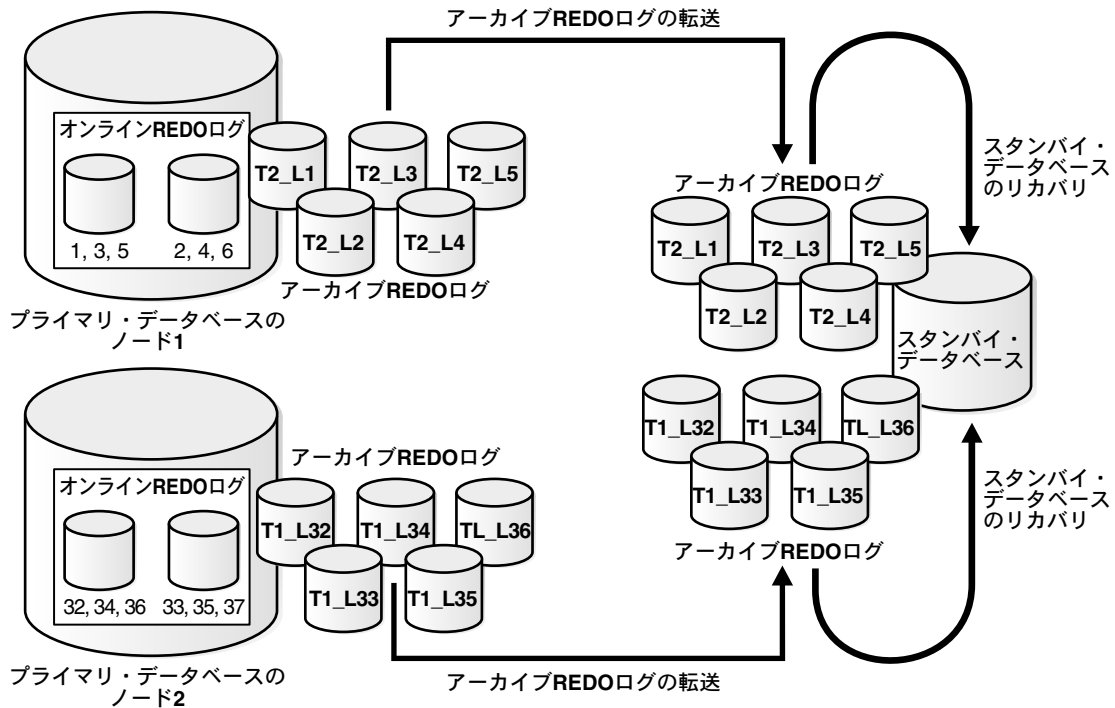
# Oracle Parallel Server 構成でのスタンバイ・データベースの使用

スタンバイ・データベースは Oracle Parallel Server（OPS）オプションと一緒に使用できます。次の表は、プライマリ・データベースおよびスタンバイ・データベースで使用可能なノードの組合せを示しています。

	単一インスタンス・スタンバイ・データベース	複数インスタンス・スタンバイ・データベース
単一インスタンス・プライマリ・データベース	はい	はい（読取り専用問合せのみ）
複数インスタンス・プライマリ・データベース	はい	はい

いずれの場合にも、プライマリ・データベースの各ノードは、自分のアーカイブ REDO ログのスレッドを、スタンバイ・データベースに転送します。たとえば、[図 3-3](#) は、単一インスタンス・スタンバイ・データベースに REDO ログを転送する 2 つのノードを持つ Oracle Parallel Server データベースを示しています。

図 3-3 複数インスタンス・プライマリ・データベースからのアーカイブ・ログの転送



この場合、プライマリ・データベースのノード1はログ1、2、3、4、5を転送し、ノード2はログ32、33、34、35、36を転送します。スタンバイ・データベースが管理リカバリ・モードの場合には、アーカイブ REDO ログを適用する正しい順序は自動的に判別されます。

プライマリ・データベースとスタンバイ・データベースの両方が Oracle Parallel Server 構成に含まれ、スタンバイ・データベースが管理リカバリ・モードの場合には、スタンバイ・データベースの1つのノードが、プライマリ・ノードから転送されたすべてのログ・セットに適用されます。この場合、管理リカバリの実行中は、REDO を適用していないスタンバイ・ノードを読取り専用モードにすることはできません。ほとんどの場合、リカバリに使用されていないノードはシャットダウンする必要がありますが、マウントしておくことも可能です。

**関連項目：** Oracle Parallel Server のデータベースの構成の詳細は、『Oracle8i Parallel Server セットアップおよび構成ガイド』を参照してください。



---

# スタンバイ・データベースのメンテナンスの実行

この章では、スタンバイ・データベースの典型的なメンテナンス操作の実行方法を説明します。この章は次のトピックから構成されています。

- スタンバイ・データベースに影響を与えるイベントのモニター
- スタンバイ・データベースに影響を与えるイベントへの応答
- スタンバイ・データベースのバックアップ

## スタンバイ・データベースに影響を与えるイベントのモニター

問題の発生を防止するためには、スタンバイ・データベースに影響を与えるイベントを認識し、これらのイベントをモニターする方法を知る必要があります。プライマリ・データベースの変更はほとんどの場合、アーカイブ REDO ログを通じて自動的にスタンバイ・データベースに波及するため、ユーザーの介入は必要ありません。ただし、プライマリ・データベースでの変更の種類によっては、スタンバイ・データベースに手動で介入する必要があることがあります。

この項は、次のトピックから構成されています。

- [プライマリ・データベースおよびスタンバイ・データベースのモニター](#)
- [スタンバイ・サイトで受け取ったアーカイブ・ログの判別](#)
- [スタンバイ・データベースに適用されたログの判別](#)

## プライマリ・データベースおよびスタンバイ・データベースのモニター

[表 4-1](#) は、コマンドが通常どおり波及するか、コマンドの完全な波及のための追加の管理作業が必要かを示しています。また、これらのイベントへの応答の方法も示しています。



表 4-1 コマンドの波及

プライマリ・データベースのイベント	プライマリ・サイトでの検出	スタンバイ・サイトでの検出	応答
アーカイブのエラー	<ul style="list-style-type: none"> <li>ERROR.V\$ARCHIVE_DESTINATION</li> <li>alert.log</li> <li>ARCHIVED.V\$LOG</li> <li>トレース・ファイルのアーカイブ</li> </ul>	リモート・ファイル・サーバー（RFS）プロセスのトレース・ファイル	エラーが発生した場合またはパフォーマンスが低下した場合に、アーカイブ REDO ログをプッシュまたはプルするスクリプトを作成する。4-10 ページの「 <a href="#">プライマリ・データベースへの表領域またはデータ・ファイルの追加</a> 」を参照。
スレッド・イベント	<ul style="list-style-type: none"> <li>alert.log</li> <li>V\$THREAD</li> </ul>	alert.log	スレッド・イベントはアーカイブ・ログによって自動的に波及するため、追加のアクションは必要ない。
REDO ログの変更	<ul style="list-style-type: none"> <li>alert.log</li> <li>V\$LOG および STATUS.V\$LOGFILE</li> </ul>	N/A	<p>REDO ログの変更は、REDO ログが消去されるか紛失しない限り、スタンバイ・データベースに影響しない。このような場合は、スタンバイ・データベースの再構築が必要になる。2-7 ページの「<a href="#">スタンバイ・データベース・ファイルの作成</a>」を参照。</p> <p>ALTER DATABASE CLEAR LOGFILE 文を使用し、スタンバイ・データベースのログを事前に消去すること。4-16 ページの「<a href="#">オンライン REDO ログの消去</a>」を参照。</p>
CREATE CONTROLFILE の発行	alert.log	パラメータ変更の影響を受ける REDO を検出するまではデータベースは正常に機能する。	スタンバイの制御ファイルを再作成する（4-15 ページの「 <a href="#">スタンバイ・データベースの制御ファイルのリフレッシュ</a> 」を参照）。プライマリ・データベースが RESETLOGS でオープンされている場合は、スタンバイ・データベースを再作成すること。
メディア・リカバリの実行	alert.log	N/A	RESETLOGS オプションが使用されている場合は、スタンバイ・データベースを再作成する。
表領域の状態の変更（読取り / 書込みまたは読取り専用に変更、あるいはオンラインまたはオフラインに変更）	<ul style="list-style-type: none"> <li>DBA_TABLESPACES</li> <li>alert.log</li> </ul>	<ul style="list-style-type: none"> <li>すべてのデータ・ファイルがオンラインになっていることを確認する。</li> <li>V\$RECOVER_FILE</li> </ul>	状態の変更は自動的に波及するため、応答は必要ない。データ・ファイルはオンラインのままになる。

表 4-1 ( 続き ) コマンドの波及

プライマリ・データベースのイベント	プライマリ・サイトでの検出	スタンバイ・サイトでの検出	応答
データ・ファイルの追加または表領域の作成	<ul style="list-style-type: none"><li>DBA_DATA_FILES</li><li>alert.log</li></ul>	<ul style="list-style-type: none"><li>ORA-283, ORA-1670, ORA-1157, ORA-1110</li><li>スタンバイのリカバリは停止する。</li></ul>	データ・ファイルを手動で作成し、リカバリを再起動する。4-10 ページの「 <a href="#">プライマリ・データベースへの表領域またはデータ・ファイルの追加</a> 」を参照。
表領域の削除	<ul style="list-style-type: none"><li>DBA_DATA_FILES</li><li>alert.log</li></ul>	alert.log	オペレーティング・システムからデータ・ファイルを削除する。
表領域またはデータ・ファイルがオフラインに変更またはデータ・ファイルがオフラインで削除	<ul style="list-style-type: none"><li>V\$RECOVER_FILE</li><li>alert.log</li></ul> 表領域またはデータ・ファイルをオンラインにするときにリカバリが必要になる。	<ul style="list-style-type: none"><li>すべてのデータ・ファイルがオンラインになっていることを確認する。</li><li>V\$RECOVER_FILE</li></ul>	データ・ファイルはオンラインのままになる。スタンバイ・データベースのアクティブ化後は表領域またはデータ・ファイルに問題はない。
データ・ファイルの改名	alert.log	N/A	N/A
ログされていない操作またはリカバリ不能な操作	<ul style="list-style-type: none"><li>ダイレクト・ローダーが、オンライン REDO ログ内のブロック範囲 REDO エントリを無効にする。V\$DATAFILE を検査。</li><li>V\$DATABASE</li></ul>	alert.log。ファイル・ブロックは、REDO の将来に含まれる場合はそのままになるが、そうでない限りは無効にされる。	ログされていない変更は、スタンバイ・データベースに波及しない。これらの変更を適用する場合は、4-13 ページの「 <a href="#">ダイレクト・パス処理の実行</a> 」を参照。
リカバリの進行	alert.log	<ul style="list-style-type: none"><li>V\$RECOVER_LOG</li><li>alert.log</li></ul>	スタンバイ・データベースがプライマリ・データベースから遅れていないかを確認する。

表 4-1 ( 続き ) コマンドの波及

プライマリ・データベースのイベント	プライマリ・サイトでの検出	スタンバイ・サイトでの検出	応答
データ・ファイルの自動拡張	alert.log	ディスク領域の不足のため、スタンバイ・データベースでの操作が失敗する可能性がある。	拡張したデータ・ファイルのための十分なディスク領域があることを確認すること。
OPEN RESETLOGS または CLEAR UNARCHIVED LOGFILES 文の発行	alert.log	スタンバイ・データベースは無効になる。	スタンバイ・データベースを再構築する。2-7 ページの「 <a href="#">スタンバイ・データベース・ファイルの作成</a> 」を参照。
初期化パラメータの変更	alert.log	変更されたパラメータに依存する REDO により障害が発生する可能性がある。	スタンバイのパラメータを動的に変更するか、スタンバイ・データベースをシャットダウンし、初期化パラメータ・ファイルを編集する。

## スタンバイ・サイトで受け取ったアーカイブ・ログの判別

次の表は、スタンバイ・サイトで受け取ったアーカイブ REDO ログに関する情報を入手するための、2 つの方法を示しています。

方法	利点	不利な点
スタンバイ・データベースの V\$ARCHIVED_LOG ビューにアクセスする。	<ul style="list-style-type: none"> <li>■ アクセスが容易</li> <li>■ 追加のパラメータを設定する必要がない</li> </ul>	<ul style="list-style-type: none"> <li>■ 最小限の情報しか表示されない</li> <li>■ アーカイブ・エラーを記録しない</li> </ul>
LOG_ARCHIVE_TRACE 初期化パラメータを使用し、プライマリ・サイトおよびスタンバイ・サイトのアーカイブ・トレースを設定する。	<ul style="list-style-type: none"> <li>■ トレース出力中の詳細度を管理できる</li> <li>■ 必要な場合は広範囲の情報を提供する</li> <li>■ アーカイブのエラーと成功を記録する</li> </ul>	初期化パラメータの設定およびトレース出力の解析が必要。

## V\$ARCHIVED\_LOG ビューのアクセス

スタンバイ・サイトで受け取った最新のアーカイブ・ログを判別する最も簡単な方法は、V\$ARCHIVED\_LOG ビューの問合せです。このビューは、スタンバイ・サイトがログの受取りを開始するまでは、プライマリの制御ファイルから生成されたアーカイブ・ログ・レコードを含んでいるため、スタンバイ・サイトでのログの受取りの開始後にしか役に立ちません。たとえば、次のスクリプトを実行できます (サンプル出力を含む)。

```
col name format a20
col thread# format 999
col sequence# format 999
col first_change# format 999999
col next_change# format 999999

SELECT thread#, sequence# AS "SEQ#", name, first_change# AS "FIRSTSCN",
       next_change# AS "NEXTSCN", archived, deleted, completion_time AS "TIME"
FROM   v$archived_log
/

SQL> @archived_script
```

THREAD#	SEQ#	NAME	FIRSTSCN	NEXTSCN	ARC	DEL	TIME
1	947	/arc_dest/arc_1_947	33113	33249	YES	NO	23-JUN-99

アーカイブ・トレースの設定

スタンバイ・サイトへの REDO ログのアーカイブの進捗を確認するには、プライマリおよびスタンバイの初期化パラメータ・ファイル内に、LOG\_ARCHIVE\_TRACE パラメータを設定します。

LOG_ARCHIVE_TRACE	Oracle による書込み	トレース・ファイル
プライマリ・データベース	プライマリ・データベースのアーカイブ・プロセス・アクティビティの監査証跡 (ARCn およびフォアグラウンド・プロセス)	USER_DUMP_DEST 初期化パラメータで指定されたファイル名を持つファイル
スタンバイ・データベース	スタンバイ・データベースのアーカイブ REDO ログに関連する RFS プロセス・アクティビティの監査証跡	USER_DUMP_DEST 初期化パラメータで指定されたファイル名を持つファイル

**トレース・ファイルの場所の判別** データベースのトレース・ファイルは、初期化パラメータ・ファイルの USER\_DUMP\_DEST パラメータで指定されたディレクトリに入っています。SQL\*Plus を使用してプライマリおよびスタンバイのインスタンスに接続し、SHOW 文を発行して場所を確認します。

```
SQL> SHOW PARAMETER user_dump_dest
```

NAME	TYPE	VALUE
user_dump_dest	string	?/rdlms/log

**ログ・トレース・パラメータの設定** アーカイブ・トレース・パラメータの書式は次のとおりです。この *trace\_level* は整数です。

```
LOG_ARCHIVE_TRACE=trace_level
```

プライマリ・データベース内の LOG\_ARCHIVE\_TRACE パラメータを使用可能、使用禁止または変更するには、次のいずれかを実行します。

- プライマリ・データベースをシャットダウンし、初期化パラメータ・ファイルを変更し、データベースを再起動します。
- データベースがオープンおよびマウントされている状態で、ALTER SYSTEM SET LOG\_ARCHIVE\_TRACE = *integer* を発行します。

読取り専用モードまたはリカバリ・モードのスタンバイ・データベース内の LOG\_ARCHIVE\_TRACE パラメータを使用可能、使用禁止または変更するには、次の SQL 文を発行します。

```
ALTER SYSTEM SET ...;
```

管理リカバリがアクティブな場合には、プライマリ・データベースから次のアーカイブ・ログを受け取ったときに、リモート・ファイル・サービス (RFS) によって生成されたトレース・ファイルに影響を与えるように、別のスタンバイ・セッションで ALTER SYSTEM 文を発行します。たとえば、次のように入力します。

```
SQL> ALTER SYSTEM SET log_archive_trace=32;
```

**整数値の選択** LOG\_ARCHIVE\_TRACE パラメータの整数値は、トレース・データのレベルを表します。一般的には、レベルが高いほど、情報は詳細になります。次の整数レベルを使用できます。

レベル	意味
0	アーカイブ・ログのトレースは使用禁止。デフォルトの設定値。
1	REDO ログ・ファイルのアーカイブ状況を追跡する。
2	アーカイブ・ログの宛先別にアーカイブ状態を追跡する。
4	アーカイブ操作フェーズを追跡する。
8	アーカイブ・ログの宛先のアクティビティを追跡する。
16	詳細なアーカイブ・ログの宛先のアクティビティを追跡する。
32	アーカイブ・ログの宛先パラメータの変更を追跡する。
64	ARCn プロセス状態アクティビティを追跡する。

LOG\_ARCHIVE\_TRACE パラメータの値を、個々のレベルの合計値に設定することにより、トレース・レベルを組み合わせることができます。たとえば、パラメータを 3 に設定すると、レベル 1 およびレベル 2 のトレース出力が生成されます。

次に示すのは、プライマリ・サイトから、サービス STANDBY1 およびローカル・ディレクトリ /vobs/oracle/dbs という 2 つの異なる宛先に REDO ログ 387 をアーカイブすることによって生成された ARC0 トレース・データの例です。

---

---

**注意：** レベル番号は、実際のトレース出力には表示されません。ここでは、明確にするために示しています。

---

---

Level	Corresponding entry content (sample)
-----	-----
( 1)	ARC0: Begin archiving log# 1 seq# 387 thrd# 1
( 4)	ARC0: VALIDATE
( 4)	ARC0: PREPARE
( 4)	ARC0: INITIALIZE
( 4)	ARC0: SPOOL
( 8)	ARC0: Creating archive destination 2 : 'standby1'
(16)	ARC0: Issuing standby Create archive destination at 'standby1'
( 8)	ARC0: Creating archive destination 1 : '/vobs/oracle/dbs/dlarc1_387.dbf'
(16)	ARC0: Archiving block 1 count 1 to : 'standby1'
(16)	ARC0: Issuing standby Archive of block 1 count 1 to 'standby1'
(16)	ARC0: Archiving block 1 count 1 to : '/vobs/oracle/dbs/dlarc1_387.dbf'
( 8)	ARC0: Closing archive destination 2 : standby1
(16)	ARC0: Issuing standby Close archive destination at 'standby1'
( 8)	ARC0: Closing archive destination 1 : /vobs/oracle/dbs/dlarc1_387.dbf
( 4)	ARC0: FINISH
( 2)	ARC0: Archival success destination 2 : 'standby1'
( 2)	ARC0: Archival success destination 1 : '/vobs/oracle/dbs/dlarc1_387.dbf'
( 4)	ARC0: COMPLETE, all destinations archived
(16)	ARC0: ArchivedLog entry added: /vobs/oracle/dbs/dlarc1_387.dbf
(16)	ARC0: ArchivedLog entry added: standby1
( 4)	ARC0: ARCHIVED
( 1)	ARC0: Completed archiving log# 1 seq# 387 thrd# 1
(32)	Propagating archive 0 destination version 0 to version 2
	Propagating archive 0 state version 0 to version 2
	Propagating archive 1 destination version 0 to version 2
	Propagating archive 1 state version 0 to version 2
	Propagating archive 2 destination version 0 to version 1
	Propagating archive 2 state version 0 to version 1
	Propagating archive 3 destination version 0 to version 1
	Propagating archive 3 state version 0 to version 1
	Propagating archive 4 destination version 0 to version 1

Propagating archive 4 state version 0 to version 1

```
(64) ARCH: changing ARC0 KCRNOARCH->KCRRSCHED
      ARCH: STARTING ARCH PROCESSES
      ARCH: changing ARC0 KCRRSCHED->KCRRSTART
      ARCH: invoking ARC0
      ARC0: changing ARC0 KCRRSTART->KCRRACTIVE
      ARCH: Initializing ARC0
      ARCH: ARC0 invoked
      ARCH: STARTING ARCH PROCESSES COMPLETE
      ARC0 started with pid=8
      ARC0: Archival started
```

次に示すのは、スタンバイ・サイトで、ディレクトリ /stby にアーカイブ・ログ 387 を受け取り、スタンバイ・データベースにログを適用することにより、RFS プロセスから生成されるトレース・データです。

```
level      trace output (sample)
-----
( 4)      RFS: Startup received from ARCH pid 9272
( 4)      RFS: Notifier
( 4)      RFS: Attaching to standby instance
( 1)      RFS: Begin archive log# 2 seq# 387 thrd# 1
(32)      Propagating archive 5 destination version 0 to version 2
(32)      Propagating archive 5 state version 0 to version 1
( 8)      RFS: Creating archive destination file: /stby/parc1_387.dbf
(16)      RFS: Archiving block 1 count 11
( 1)      RFS: Completed archive log# 2 seq# 387 thrd# 1
( 8)      RFS: Closing archive destination file: /stby/parc1_387.dbf
(16)      RFS: ArchivedLog entry added: /stby/parc1_387.dbf
( 1)      RFS: Archivelog seq# 387 thrd# 1 available 04/02/99 09:40:53
( 4)      RFS: Detaching from standby instance
( 4)      RFS: Shutdown received from ARCH pid 9272
```

## スタンバイ・データベースに適用されたログの判別

スタンバイ・データベースの、適用済の最新のログ順序番号を記録する V\$LOG\_HISTORY ビューを問い合わせます。たとえば、次の問い合わせを発行します。

```
SQL> SELECT thread#, max(sequence#) AS "LAST_APPLIED_LOG"
2> FROM   v$log_history
3> GROUP BY thread#;
```

```
THREAD# LAST_APPLIED_LOG
-----
1              967
```

この例では、最後に適用されたログは、ログ順序番号 967 のアーカイブ REDO ログです。

---

**注意：** V\$LOG はリカバリ中は更新されません。

---

## スタンバイ・データベースに影響を与えるイベントへの応答

一般的には、プライマリ・データベースの物理的な変更を行うと、スタンバイ・データベースでは手動の応答が必要になります。この項は、次のトピックから構成されています。

- [プライマリ・データベースへの表領域またはデータ・ファイルの追加](#)
- [プライマリ・データベースのデータ・ファイルの改名](#)
- [プライマリ・データベースの REDO ログの追加または削除](#)
- [プライマリ・データベースのアーカイブしていない REDO ログのリセットまたは消去](#)
- [プライマリ・データベースの制御ファイルの変更](#)
- [スタンバイ・データベースのデータ・ファイルをオフラインにする方法](#)
- [ダイレクト・パス処理の実行](#)
- [スタンバイ・データベースの制御ファイルのリフレッシュ](#)
- [オンライン REDO ログの消去](#)

## プライマリ・データベースへの表領域またはデータ・ファイルの追加

プライマリ・データベースに表領域またはデータ・ファイルを追加すると REDO が生成されますが、この REDO をスタンバイ・データベースで適用するときには、スタンバイの制御ファイルにデータ・ファイル名が自動的に追加されます。スタンバイ・データベースが、制御ファイル内で指定されたファイル名を持つファイルを検出すると、リカバリは続行されます。スタンバイ・データベースが、制御ファイル内で指定されたファイル名を持つファイルを検出できない場合は、リカバリは終了します。

プライマリ・データベース内に新しいデータ・ファイルを作成し、スタンバイ・データベースを更新するには次の手順の 1 つを実行してください。スタンバイ・データベースで新しいデータ・ファイルが必要ない場合には、次の構文を使用してデータ・ファイルを手動でオフラインにすることができます。

```
SQL> ALTER DATABASE DATAFILE 'filename' OFFLINE DROP;
```

### プライマリ・データベースに表領域またはデータ・ファイルを追加し、スタンバイ・データベースでデータ・ファイルを作成する

1. 通常の方法でプライマリ・データベースに表領域を作成します。たとえば、新しいデータ・ファイル `t_db2.f` を表領域 `tbs_2` 内に作成するには、次のコマンドを発行します。



```
SQL> CREATE TABLESPACE tbs_2 DATAFILE 't_db2.f' SIZE 2M;
```

2. スタンバイ・データベースがシャットダウンしている場合は、データベースをマウントせずに、スタンバイ・インスタンスを起動します。たとえば、次のように入力します。

```
SQL> STARTUP NOMOUNT pfile=/privatel/stby/initSTANDBY.ora
```

スタンバイ・データベースが現在管理リカバリ・モードになっている場合は、ステップ 4 まで進んでください。

3. スタンバイ・データベースをマウントし、管理リカバリ・モードにします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

4. スタンバイ・データベースへの REDO のアーカイブを開始するように、プライマリ・データベースの REDO ログを切り替えます。

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

スタンバイ・データベースのリカバリ処理が CREATE TABLESPACE 文を含む REDO ログを適用しようとする、スタンバイ・サイトには新しいデータ・ファイルが存在しないため、リカバリ処理は停止します。

5. 新しいデータ・ファイルを検出できないためにスタンバイ・データベースでリカバリが取消しになるのを待つか、管理リカバリの手動の取消しを行います。

```
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL;
```

CREATE TABLESPACE 文を含む REDO は、スタンバイの制御ファイルに新しいファイル名を追加することに注意してください。次の alert.log エントリが生成されます。

```
WARNING! Recovering datafile 2 from a fuzzy file. If not the current file it
might be an online backup taken without entering the begin backup command.
Successfully added datafile 2 to media recovery
Datafile #2: '/privatel/stby/t_db2.f'
```

6. スタンバイ・データベースにデータ・ファイルを作成します。たとえば、次のコマンドを発行します。

```
SQL> ALTER DATABASE CREATE DATAFILE '/privatel/stby/t_db2.f'
      AS '/privatel/stby/t_db2.f';
```

7. スタンバイ・データベースを管理リカバリ・モードに設定します。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

プライマリ・データベースで通常の処理を続行します。これで、プライマリ・データベースとスタンバイ・データベースが同期しました。

**関連項目：** オフラインのデータ・ファイルの変更の詳細は、4-13 ページの「[スタンバイ・データベースのデータ・ファイルをオフラインにする方法](#)」を参照してください。

## プライマリ・データベースのデータ・ファイルの改名

プライマリ・データベースでのデータ・ファイルの改名は、スタンバイ・データベースの制御ファイルのリフレッシュしない限り、スタンバイ・データベースでは有効になりません。プライマリ・データベースのデータ・ファイルを改名したときには、プライマリ・データベースおよびスタンバイ・データベースのデータ・ファイルの同期を保つために、スタンバイ・データベースでも同様の操作を実行してください。

## プライマリ・データベースの REDO ログの追加または削除

スタンバイ・データベースに影響を与えることなく、プライマリ・データベースに REDO ログ・ファイルのグループまたはメンバーを追加できます。同様に、スタンバイ・データベースに影響を与えることなく、プライマリ・データベースで REDO ログ・ファイルのグループまたはメンバーを削除できます。プライマリ・データベースでスレッドを使用可能または使用禁止にしても、スタンバイ・データベースには影響はありません。

プライマリ・データベースとスタンバイ・データベースで、オンライン REDO ログの構成を同じにするかどうかを検討してください。プライマリ・データベースとスタンバイ・データベースでオンライン REDO ログの構成が異なってもスタンバイ・データベースの機能に影響はありませんが、アクティブ化後のスタンバイ・データベースのパフォーマンスには影響します。たとえば、プライマリ・データベースに 10 の REDO ログがあり、スタンバイ・データベースに 2 つの REDO ログがある場合、新しくプライマリ・データベースとして機能させるためにスタンバイ・データベースをアクティブにすると、新しいプライマリ・データベースでは古いプライマリ・データベースよりも頻繁なアーカイブを強制されることになります。

スタンバイのアクティブ化後の問題を防止するために、プライマリ・データベースとスタンバイ・データベースでオンライン REDO ログの構成を同じにすることをお勧めします。プライマリ・データベースで ALTER DATABASE ENABLE THREAD 文を使用してログ・ファイル・スレッドを使用可能にしたときには、スタンバイ・データベースをアクティブにする前に、スタンバイ・データベースに新しい制御ファイルを作成する必要がありますので注意してください。手順は 4-15 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

## プライマリ・データベースのアーカイブしていない REDO ログのリセットまたは消去

ALTER DATABASE CLEAR UNARCHIVED LOGFILE 文を発行してプライマリ・データベースのログ・ファイルを消去した場合、または RESETLOGS オプションを使用してプライマリ・データベースをオープンした場合、スタンバイ・データベースは無効になります。これらの操作は両方ともにプライマリ・ログの順序番号を 1 にリセットするため、プライマ

リ・データベースから生成されたアーカイブ・ログを適用するためには、スタンバイ・データベースを再作成する必要があります。手順は 2-7 ページの「[スタンバイ・データベース・ファイルの作成](#)」を参照してください。追加情報は 5-43 ページの「[シナリオ 8: スタンバイ・データベースの再作成](#)」を参照してください。

## プライマリ・データベースの制御ファイルの変更

次のいずれかの操作を実行するために、プライマリ・データベースで CREATE CONTROLFILE 文を使用した場合、スタンバイ・データベースの制御ファイルが無効になることがあります。

- REDO ログ・ファイルのグループまたはメンバーの最大数の変更
- データベースを同時にマウントおよびオープンできるインスタンスの最大数の変更

プライマリ・データベースで RESETLOGS オプションを指定して CREATE CONTROLFILE 文を使用すると、プライマリ・データベースの次のオープン時にオンライン・ログがリセットされるため、スタンバイ・データベースが無効になります。

スタンバイ・データベースの制御ファイルが無効にした場合には、4-15 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」で説明する手順を使用し、ファイルを再作成してください。

## スタンバイ・データベースのデータ・ファイルをオフラインにする方法

プライマリ・データベースのデータ・ファイルのサブセットをサポートする手段として、スタンバイ・データベースのデータ・ファイルをオフラインにすることができます。たとえば、スタンバイ・データベースで、プライマリ・データベースの一時表領域のリカバリを行わないことがもできます。

スタンバイ・データベースで次の文を使用し、データ・ファイルをオフラインにします。

```
ALTER DATABASE DATAFILE 'filename' OFFLINE DROP;
```

この文を実行するときには、スタンバイ・データベースをオープンした後で、オフラインのファイルを含む表領域を削除する必要があります。

## ダイレクト・パス処理の実行

次のいずれかの結果としてダイレクト・ロードを実行した場合、パフォーマンスの向上は、プライマリ・データベースにのみ適用されます（スタンバイ・データベースでは、相当するリカバリ処理のパフォーマンス向上はありません）。

- ダイレクト・パス・ロード
- 副問合せによる CREATE TABLE
- プライマリ・データベースでの CREATE INDEX

スタンバイ・データベースのリカバリ処理は、リカバリ不能なダイレクト・ロードによって生成された REDO 情報の順次の読取りおよび適用を続行します。

### UNRECOVERABLE プロセスの手動の波及

UNRECOVERABLE オプションを使用するプライマリ・データベースのプロセスは、アーカイブ REDO ログに含まれないため、スタンバイ・データベースには波及しません。プライマリ・データベースで UNRECOVERABLE 操作を実行した後スタンバイ・データベースのリカバリを行うと、リカバリ中はエラー・メッセージは表示されず、スタンバイ・データベースの alert.log ファイルにエラー・メッセージが書き込まれます。次のようなエラー・メッセージが表示されます。

```
26040, 00000, " データ・ブロックが NOLOGGING オプションを使用してロードされました。¥n"
//* Cause: Trying to access data in block that was loaded without
//*          redo generation using the NOLOGGING/UNRECOVERABLE option
//* Action: Drop the object containing the block.
```

エラー・メッセージではブロックを含むオブジェクトを削除するように指示していますが、この操作は実行しないでください。そのかわりに、次の作業の 1 つを実行します。

- スタンバイ・データベースで影響を受けるデータ・ファイルをオフラインにし、アクティビ化後に表領域を削除します (4-13 ページの「[スタンバイ・データベースのデータ・ファイルをオフラインにする方法](#)」を参照)。
- 新しいデータベース・バックアップからスタンバイ・データベースを再作成します (2-7 ページの「[スタンバイ・データベース・ファイルの作成](#)」を参照)。
- 影響を受ける表領域をバックアップし、プライマリ・データベース中の現行のログをアーカイブし、スタンバイ・データベースにデータ・ファイルを転送して、スタンバイのリカバリを再開します。これは、UNRECOVERABLE 操作後にデータベースの通常のリカバリ可能性を確保するために実行する手順と同じです。

**関連項目：** 詳細は、4-13 ページの「[スタンバイ・データベースのデータ・ファイルをオフラインにする方法](#)」および 5-28 ページの「[シナリオ 4: NOLOGGING 句の指定後のリカバリ](#)」を参照してください。

### UNRECOVERABLE 操作後にバックアップが必要かどうかの判断

プライマリ・データベースで UNRECOVERABLE 操作を実行したときには、新しいバックアップが必要かどうかを判断してください。

#### 新しいバックアップが必要かどうかの判断

1. プライマリ・データベースで V\$DATAFILE ビューの問合せを行い、Oracle が最新の無効 REDO データを生成したときのシステム変更番号 (SCN) または時刻を確認します。
2. プライマリ・データベースで次の SQL 文を発行し、新しいバックアップの実行が必要かどうかを判断します。

```
SELECT unrecoverable_change#,  
       to_char(unrecoverable_time, 'mm-dd-yyyy hh:mi:ss')  
FROM   v$datafile;
```

3. 前述のステップの問合せで、データ・ファイルのリカバリ不能時間が最後のバックアップ以降と報告された場合には、問題のデータ・ファイルの新しいバックアップを作成してください。

**関連項目：** V\$DATAFILE ビューの詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

## スタンバイ・データベースの制御ファイルのリフレッシュ

プライマリ・データベースの制御ファイルで行った変更をリフレッシュする方法、または変更のコピーを作成する方法を説明します。ファイルの追加または削除など、プライマリ・データベースで大きな構造上の変更を行った後は、スタンバイ・データベースの制御ファイルをリフレッシュしてください。

### スタンバイ・データベースの制御ファイルのリフレッシュ

1. スタンバイ・インスタンスで SQL\*Plus セッションを開始し、スタンバイ・データベースで CANCEL 文を発行し、リカバリ処理を停止します。

```
SQL> RECOVER CANCEL # for manual recovery mode  
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL # for managed recovery mode
```

2. スタンバイ・インスタンスをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE
```

3. 本番インスタンスで SQL\*Plus セッションを開始し、スタンバイ・データベースの制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'filename';
```

4. バイナリ・ファイルに適したオペレーティング・システムのユーティリティを使用し、スタンバイの制御ファイルおよびアーカイブ・ログ・ファイルをスタンバイ・サイトに転送します。

5. スタンバイ・インスタンスに接続し、スタンバイ・データベースをマウントします（オープンはしません）。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

6. スタンバイ・データベースでリカバリ処理を再起動します。

```
SQL> RECOVER STANDBY DATABASE # recovers using location for logs  
                                # specified in initialization parameter file
```

```
SQL> RECOVER FROM 'location' STANDBY DATABASE # recovers from specified
# location
```

オンライン REDO ログの消去

スタンバイ・データベースを作成した後、次の文を発行し、スタンバイ・データベースのオンライン REDO ログを消去することにより、パフォーマンスを最適化できます。この *integer* はログ・グループの数を表します。

```
ALTER DATABASE CLEAR LOGFILE GROUP integer;
```

Oracle はアクティブ化時にログを「ゼロ設定」する必要がなくなるため、この文によりスタンバイのアクティブ化が最適化されます。ゼロ設定では、REDO ログ全体にゼロを書き込み、REDO ログを作成時の外見に戻すために新しいヘッダーを設定します。ゼロ設定は、RESETLOGS 操作中に発生します。

ログを手動で消去すると、Oracle はアクティブ化時にログがゼロになっていることを認識し、ゼロ設定を省略します。すべてのオンライン・ログにゼロを書き込むには長い時間がかかることがあるため、この最適化は重要です。メンテナンス中にこの操作を実行しないと、Oracle はアクティブ化時にオンライン・ログを自動的に消去します。

スタンバイ・データベースのバックアップ

必要であれば、スタンバイ・データベースをバックアップできますが、データベースが手動リカバリ・モードまたは管理リカバリ・モードにある間はバックアップできません。スタンバイ・データベースの管理リカバリ・モードを終了し、バックアップを作成してから、管理リカバリを再開する必要があります。データベースがシャットダウンしている、または読取り専用モードのときにバックアップを作成できます。

次の表は、それぞれの方法の利点および不利な点を示しています。

方法	利点	不利な点
スタンバイ・データベースをシャットダウンする。	データベースのシャットダウンが必要なその他のメンテナンス操作の実行後にデータベースをバックアップできる。	スタンバイ・データベースがアーカイブ・ログを受け取らないため、プライマリ・データベースでギャップ・シーケンスが作成されることがある。ギャップ・シーケンスが作成された場合は、スタンバイ・データベースを管理リカバリ・モードにする前に、手動リカバリを実行する必要がある。

方法	利点	不利な点
スタンバイ・データベースを 読取り専用モードにする。	スタンバイ・サイトはプライ マリ・データベースからの アーカイブ・ログを受け取り 続けるため、ギャップ・シー ケンスは作成されない。	

### 読取り専用モードのスタンバイ・データベースでの表領域のバックアップ

1. スタンバイ・データベースで SQL\*Plus セッションを開始し、データベースの管理リカバリ・モードまたは手動リカバリ・モードを終了します。

```
RECOVER MANAGED STANDBY DATABASE CANCEL    # for managed recovery
RECOVER CANCEL                               # for manual recovery
```

2. 読取り専用モードでデータベースをオープンします。

```
ALTER DATABASE OPEN READ ONLY
```

3. オペレーティング・システムのユーティリティを使用し、いくつかの表領域のバックアップを作成します。スタンバイの制御ファイルをバックアップしないでください。

データベースのダウン時間は最小限にします。たとえば、UNIX で表領域 TBS\_1 内のデータ・ファイル tbs11.f、tbs12.f および tbs13.f をバックアップするには、次のように入力します。

```
% cp /disk1/oracle/dbs/tbs11.f /disk2/backup/tbs11.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs12.bk
% cp /disk1/oracle/dbs/tbs13.f /disk2/backup/tbs13.bk
```

4. スタンバイ・データベースのアクティブなユーザー・セッションをすべて終了します。
5. データベースを手動リカバリ・モードまたは管理リカバリ・モードにします。

```
RECOVER MANAGED STANDBY DATABASE    # for managed recovery
RECOVER STANDBY DATABASE             # for manual recovery
```

6. オペレーティング・システムのユーティリティを使用し、プライマリ・データベースの制御ファイルをバックアップします。スタンバイ・データベースの制御ファイルではなく、プライマリ・データベースの制御ファイルをバックアップしてください。
7. データベース中の各表領域をバックアップするまで、前述のステップを繰り返します。

### シャットダウン中のスタンバイ・データベースでの表領域のバックアップ

1. スタンバイ・データベースで SQL\*Plus セッションを開始し、データベースの管理リカバリ・モードまたは手動リカバリ・モードを終了します。

```
RECOVER MANAGED STANDBY DATABASE CANCEL    # for managed recovery
RECOVER CANCEL                              # for manual recovery
```

2. データベースをシャットダウンします。

```
SHUTDOWN IMMEDIATE
```

3. オペレーティング・システムのユーティリティを使用し、いくつかの表領域のコールド・バックアップを作成します。データベースのダウン時間は最小限にします。たとえば、UNIX で表領域 TBS\_1 内のデータ・ファイル tbs11.f、tbs12.f および tbs13.f をバックアップするには、次のように入力します。

```
% cp /disk1/oracle/dbs/tbs11.f /disk2/backup/tbs11.bk
% cp /disk1/oracle/dbs/tbs12.f /disk2/backup/tbs12.bk
% cp /disk1/oracle/dbs/tbs13.f /disk2/backup/tbs13.bk
```

4. 必要であればパラメータ・ファイルを指定し、SQL\*Plus を使用し、スタンバイ・データベースをマウントせずに Oracle インスタンス起動します。

```
STARTUP NOMOUNT pfile = initSTANDBY.ora
```

5. データベースをマウントします。

```
ALTER DATABASE MOUNT STANDBY DATABASE
```

6. データベースを手動リカバリ・モードまたは管理リカバリ・モードにします。

```
RECOVER MANAGED STANDBY DATABASE    # for managed recovery
RECOVER STANDBY DATABASE             # for manual recovery
```

7. データベース中の各表領域をバックアップするまで、前述のステップを繰り返します。



---

## スタンバイ・データベースのシナリオ

この章では、次のスタンバイ・データベースのシナリオを説明します。

- シナリオ 1: 同一ホスト上のスタンバイ・データベースの作成
- シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成
- シナリオ 3: プライマリ・データベースの物理的変更への対応
- シナリオ 4: NOLOGGING 句の指定後のリカバリ
- シナリオ 5: 複数スタンバイ・データベース構成でのフェイルオーバー先のスタンバイ・データベースの決定
- シナリオ 6: クライアント・アプリケーション・フェイルオーバーの構成
- シナリオ 7: ネットワーク障害後のリカバリ
- シナリオ 8: スタンバイ・データベースの再作成
- シナリオ 9: 進行中のリカバリがないスタンバイ・データベース
- シナリオ 10: 時間差を設定したスタンバイ・データベース

## シナリオ 1: 同一ホスト上のスタンバイ・データベースの作成

このシナリオでは、プライマリ・データベース PROD1 と同じホスト上でのスタンバイ・データベース STANDBY1 の作成について説明します。ホストは、異なるコントローラ上の別のディスク構成にマウントされた 3 つのファイル・システムを持つ UNIX マシンです。スタンバイ・データベースをプライマリ・データベースとは異なるファイル・システムに配置することにより、プライマリ・データベースをハード・ディスク障害から保護できます。同一ホスト上のスタンバイ・データベースを管理リカバリ・モードにすることにより、スタンバイ・データベースが絶えず更新されるようにします。

ローカル・ホスト上のスタンバイ・データベースをセットアップした後、総合的な災害時保護のために、リモート・ホスト上にスタンバイ・データベースを作成することを計画します。この方法では、災害によりプライマリ・データベースのすべてのディスクがクラッシュしたり、破壊された場合でも、リモートのスタンバイ・データベースにフェイルオーバーすることにより、データベースをオープンしたままにしておくことができます。

### ステップ 1: スタンバイ・データベースの計画

ホスト・システムでは、それぞれが専用のコントローラを持つ専用のディスク・セット上で 3 つのファイル・システムを使用しているため、プライマリ・データベースのファイルを最初のファイル・システムにメンテナンスし、スタンバイ・データベースのファイルを 2 番目のファイル・システムにメンテナンスし、ORACLE\_HOME バイナリを 3 番目のファイル・システムにメンテナンスすることになります。プライマリ・データベースのディスクに障害が発生した場合は、スタンバイ・データベースに切り替えることができます。ORACLE\_HOME ディスクに障害が発生した場合には、リモートのスタンバイ・データベースに切り替えることができます。

プライマリ・データベースと同じマシンをスタンバイ・データベースのホストとするためには、スタンバイ・データベースの初期化パラメータ・ファイル内で次のパラメータを設定する必要があります。

- CONTROL\_FILES
- LOCK\_NAME\_SPACE
- DB\_FILE\_NAME\_CONVERT
- LOG\_FILE\_NAME\_CONVERT

プライマリ・データベースのほとんどのデータ・ファイルおよび REDO ログ・ファイルは、同じディレクトリに入れられ、一貫性のある名前になっています。ALTER DATABASE 文を使用して改名する必要のあるファイルも一部あります。

プライマリ・データベースは、メンテナンスのために毎週日曜日に 1 時間だけシャットダウンされるため、この時間を利用して、コールドの一貫性バックアップを作成できます。その後データベースを再起動し、スタンバイ・データベースで必要な構成作業を行うことができます。

## ステップ 2: スタンバイ・データベースの作成

この手順の次のステップでは、スタンバイ・データベースのベースとするバックアップを作成します。一貫性バックアップまたは非一貫性バックアップを作成できますが、メンテナンスのために毎週日曜日にデータベースをダウンさせるため、この機会に一貫性バックアップを作成し、これをスタンバイ・データベースで使用することにします。

### 1. データベース・ファイルを判別します。

日曜日に、プライマリ・データベースをシャットダウンする前に、データベースを問い合せて、データベースに含まれているデータ・ファイルを確認します。

```
SQL> SELECT name FROM v$datafile;  
NAME
```

```
-----  
/fs1/dbs/tbs_01.f  
/fs1/dbs/tbs_02.f  
/fs1/dbs/tbs_11.f  
/fs1/dbs/tbs_12.f  
/fs1/dbs/tbs_21.f  
/fs1/dbs/tbs_22.f  
/fs1/dbs/tbs_13.f  
/fs1/dbs/tbs_23.f  
/fs1/dbs/tbs_24.f  
/fs1/dbs/tbs_31.f  
/fs1/dbs/tbs_32.f  
/fs1/dbs/tbs_41.f  
/fs1/dbs2/tbs_42.f  
/fs1/dbs2/tbs_51.f  
/fs1/dbs2/tbs_52.f  
/fs1/dbs2/tbs_03.f  
/fs1/dbs3/tbs_14.f  
/fs1/dbs3/tbs_25.f  
/fs1/dbs3/tbs_33.f  
/fs1/dbs3/tbs_43.f  
/fs1/dbs3/tbs_53.f  
21 rows selected.
```

### 2. データ・ファイルをバックアップします。

データベースに含まれているデータ・ファイルを判別した後、IMMEDIATE オプションを指定してデータベースをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

この時点で、次の方法で、プライマリのすべてのデータ・ファイルを一時ディレクトリにバックアップすることに決定します。

```
% cp /fs1/dbs/* /fs1/temp
% cp /fs1/dbs2/* /fs1/temp
% cp /fs1/dbs3/* /fs1/temp
```

その他の日常的なメンテナンス操作をいくつか実行した後、次のようにデータベースを再起動します。

```
SQL> STARTUP PFILE=initPROD1.ora;
```

3. スタンバイ・データベースの制御ファイルを作成します。

数分後に、一貫性バックアップを格納したディレクトリと同じディレクトリに、スタンバイ・データベースの制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/fs1/temp/stbycf.f';
```

4. スタンバイのファイル・システムにファイルを転送します。

スタンバイ・データベースの制御ファイルが正常に作成されると、データ・ファイルおよびスタンバイ・データベースの制御ファイルを、プライマリのファイル・システムからスタンバイのファイル・システムにコピーできるようになります。

データ・ファイルの転送には長い時間がかかることがあるため、最初に制御ファイルをコピーし、データ・ファイルのコピーを開始してから、その他の作業（たとえばネットワーク構成）にとりかかります。たとえば、UNIX コマンド・シェルに次のように入力します。

```
% cp /fs1/temp/stbycf.f /fs2/dbs/cf1.f
% cp /fs1/temp/tbs* /fs2/dbs
```

## ステップ 3: ネットワーク・ファイルの構成

スタンバイ・データベースを管理スタンバイ環境で実行するためには、スタンバイ・サービスに REDO ログをアーカイブできるように、プライマリ・データベースとスタンバイ・データベースの間に Net8 接続を構成する必要があります。

プライマリ・データベースとスタンバイ・データベースは同じホスト上にあるため、両ホストを接続するために IPC プロトコルを使用します。Oracle Names Server を使用していないため、プライマリ・データベースのための tnsnames.ora エントリと、スタンバイ・データベースのための listener.ora エントリの両方を作成する必要があります。

1. tnsnames.ora ファイルを構成します。

次のステップでは、テキスト・エディタを使用して tnsnames.ora ファイルをオープンします。

```
% vi /fs3/oracle/network/admin/tnsnames.ora
```

現在のところ、このファイルには 1 つのサービス名のエントリしか存在していません。それは、PROD1 データベースへの TCP/IP 接続です。

```
prod1 = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (PORT=1512) (HOST=dlsun183))
  (CONNECT_DATA= (SID=prod1))
)
```

プライマリ・データベースとスタンバイ・データベースの間に IPC 接続を定義するためには、次の書式でエントリを追加します。

```
standby_service_name = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=ipc) (KEY=keyhandle))
  (CONNECT_DATA= (SID=standby_sid)))
```

次の例で示すように、standby\_service\_name、keyhandle および standby\_sid を適切な値で置換します。

```
standby1 = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=ipc) (KEY=kstandby1))
  (CONNECT_DATA= (SID=standby1)))
```

## 2. listener.ora ファイルを構成します。

次のステップでは、ファイル・システム /fs3 にある listener.ora ファイルを開きます。

```
% vi /fs3/oracle/network/admin/listener.ora
```

次のアドレスのリスト（リスナーがホストのどこをリスニングするか）および SID（リスナーがどの接続をリスニングするか）が検出されます。

```
LISTENER = (ADDRESS_LIST=
  (ADDRESS= (PROTOCOL=tcp) (PORT=1512) (HOST=dlsun183))
)
SID_LIST_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=PROD1) (ORACLE_HOME=/fs3/oracle))
)
```

現在のところ、リスナーはデータベース PROD1 に関して、ホスト dlsun183 のポート 1512 をリスニングしています。

listener.ora ファイルを編集し、次の書式で2つのエントリを追加する必要があります。

```
STANDBY_LISTENER = (ADDRESS_LIST= (ADDRESS= (PROTOCOL=ipc)
  (KEY=keyhandle)))

SID_LIST_STANDBY_LISTENER = (SID_LIST=
  (SID_DESC= (SID_NAME=standby_sid) (ORACLE_HOME=/oracle_home)))
```

listener.ora ファイルは、通常はスタンバイ・サイトの \$ORACLE\_HOME/network/admin ディレクトリに入っています。次の例で示すように、keyhandle、standby\_sid および oracle\_home を適切な値で置換します。

```
STBY1_LISTENER = (ADDRESS_LIST=(ADDRESS=(PROTOCOL=ipc) # same node as primary
                        (KEY=kstdby1))) # ORACLE_SID standby instance is started
with
```

```
SID_LIST_STBY1_LISTENER = (SID_LIST=
                        (SID_DESC=(SID_NAME=stdby1) (ORACLE_HOME=/fs3/oracle)))
```

listener.ora ファイルを編集したため、ファイル内の変更が認識されるように、リスナーを再起動する必要があります。

```
% lsnrctl
LSNRCTL for Solaris: Version 8.1.6.0.0 - Production on 05-APR-99 11:39:41
```

```
(c) Copyright 1998 Oracle Corporation. All rights reserved.
```

```
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> start stby1_listener
```

この項で説明したステップのかわりに、Net8 Assistant グラフィカル・ユーザー・インタフェースを使用して、ネットワーク・ファイルを構成することもできます。追加情報は、『Oracle8i Net8 管理者ガイド』を参照してください。

## ステップ 4: プライマリ・データベースのパラメータ・ファイルの構成

ネットワーク・ファイルが構成できたため、これでプライマリ・データベースの初期化パラメータ・ファイルを編集できます。プライマリ・データベースは現在稼動中であるため、インスタンスを再起動するか、ALTER SESSION または ALTER SYSTEM 文を発行しない限り、変更は有効になりません。

ファイルに行う必要のある変更は、スタンバイ・サービスへのアーカイブに関連するもののみです。プライマリ・データベースの現在のパラメータ・ファイルは次のようになっています。

```
db_name=prod1
control_files=(/fs1/dbs/cf1.f,/fs1/dbs/cf2.f)
compatible=8.1.6
log_archive_start = TRUE
log_archive_dest_1 = 'LOCATION=/fs1/arc_dest/ MANDATORY REOPEN=60'
log_archive_dest_state_1 = ENABLE
log_archive_format = log_%t_%s.arc
audit_trail=FALSE
o7_dictionary_accessibility=FALSE
global_names=FALSE
```

```
db_domain=regress.rdbms.dev.us.oracle.com
remote_login_passwordfile = exclusive
```

```
# default parameters for instance 1
processes=30
sessions=30
transactions=21
transactions_per_rollback_segment=21
distributed_transactions=10
db_block_buffers=1000
db_files=200
shared_pool_size=10000000
```

1. スタンバイのアーカイブ先を指定します。

現在は、ローカル・ディレクトリという 1 つの宛先のみアーカイブしています。ローカル・スタンバイ・データベースで管理リカバリ・モードを使用するためには、サービス名を使用して、新しいアーカイブ場所を指定する必要があります。

テキスト・エディタを使用してプライマリ・データベースの初期化パラメータ・ファイルをオープンし、現行のアーカイブ場所および書式を確認します。

```
log_archive_dest_1 = 'LOCATION=/fs1/arc_dest/ MANDATORY REOPEN=60'
log_archive_dest_state_1 = ENABLE
log_archive_format = log_%t_%s.arc
```

パラメータ / オプション	意味
LOG_ARCHIVE_DEST_1	アーカイブ先を示す
LOCATION	ローカル・ディレクトリを示す
LOG_ARCHIVE_DEST_STATE_1	LOG_ARCHIVE_DEST_1 アーカイブ先の状態を示す
ENABLE	Oracle が宛先にアーカイブできることを示す
LOG_ARCHIVE_FORMAT	ログ・ファイルのファイル名の書式を示す

2. サービス STANDBY1 のスタンバイ・データベースにアーカイブするため、ファイルを編集し、次のエントリを追加します。

```
log_archive_dest_2 = 'SERVICE=standby1 OPTIONAL REOPEN=180'
log_archive_dest_state_2 = ENABLE
```

パラメータ / オプション	意味
LOG_ARCHIVE_DEST_2	新しいアーカイブ先を示す。LOG_ARCHIVE_DEST_1 は /fs1/arc_dest/ へのローカル・アーカイブ用にすでに確保されています。
SERVICE	スタンバイ・データベースのサービス名を示す。
OPTIONAL	この宛先で失敗しても、Oracle がオンライン REDO ログを再使用できることを示す。
REOPEN	失敗した宛先へのアーカイブを再試行するまでにアーカイブ・プロセスが待つ秒数を示す。
LOG_ARCHIVE_DEST_STATE_2	LOG_ARCHIVE_DEST_2 アーカイブ先の状態を示す。
ENABLE	Oracle が宛先にアーカイブできることを示す。

プライマリ・データベースの初期化パラメータ・ファイルを編集した後、スタンバイ・データベースで使用するためのコピーを作成します。

```
% cp /fs1/temp/initPROD1.ora /fs3/oracle/dbs/initSTANDBY1.ora
```

プライマリ・データベースの初期化パラメータ・ファイルに IFILE パラメータが含まれている場合には、IFILE パラメータが参照するファイルもスタンバイ・サイトにコピーし、必要であれば、適切な変更を行う必要があります。

## ステップ 5: スタンバイ・データベースのパラメータ・ファイルの構成

表 5-1 に示した初期化パラメータ・ファイルはスタンバイ・データベースのリカバリ処理に重要な役割を果たすことを認識し、このファイルを編集することになります。



表 5-1 スタンバイ・データベースの初期化パラメータの構成

パラメータ	設定値
COMPATIBLE	このパラメータは、プライマリ・データベースとスタンバイ・データベースで同じにする必要があります。プライマリ・データベースのパラメータ・ファイルですでに 8.1.6 に設定されているため、スタンバイの設定値はこのままにしておきます。
CONTROL_FILES	このパラメータは、プライマリ・データベースとスタンバイ・データベースで異なる値にする必要があります。制御ファイルは /fs2/dbfs ディレクトリに入れることにします。
DB_FILE_NAME_CONVERT	スタンバイ・データベースのファイル名をプライマリ・データベースのファイル名と区別するときに設定します。ほとんどの（すべてではない）データ・ファイルは /fs1/dbfs ディレクトリに入っています。このパラメータを /fs2/dbfs に設定し、/dbfs サブディレクトリ内のファイルは自動的に変換し、その他のファイルは ALTER DATABASE RENAME FILE 文を使用して変換します。
DB_FILES	プライマリ・データベースで許可するファイルの数と同じ数のファイルをスタンバイ・データベースで許可するために、DB_FILES は両方のデータベースで同じ設定にする必要があります。このため、このパラメータはそのままにします。DB_FILES が MAXDATAFILES と等しいか、これよりも大きくない限り、インスタンスがデータベースをマウントすることはできません。
DB_NAME	この値は、本番データベースのパラメータ・ファイル内の DB_NAME と同じにする必要があります。このため、このパラメータはそのままにします。
LOCK_NAME_SPACE	分散ロック・マネージャ（DLM）がロック名の作成に使用するネーム・スペースを指定します。スタンバイ・データベースとプライマリ・データベースでホストを共有する場合は、この値を設定してください。名前を STANDBY1 に設定することにします。
LOG_ARCHIVE_DEST_1	このパラメータは、アーカイブ REDO ログの場所を指定します。手動リカバリを実行するときにはこのディレクトリを使用する必要があります。この値は /fs2/arc_dest/ に設定することにします。
LOG_FILE_NAME_CONVERT	スタンバイ・データベースの REDO ログのファイル名をプライマリ・データベースの REDO ログのファイル名と区別するときに設定します。プライマリの REDO ログは /fs1/dbfs に入れられるため、スタンバイのログは /fs2/dbfs に入れることにします。
STANDBY_ARCHIVE_DEST	Oracle は、プライマリ・サイトから受け取ったログの名前を作成するためにこの値を使用します。この値は /fs2/stdby/ に設定することにします。

スタンバイ・データベースのパラメータ・ファイルを次のように編集します（編集した値は太字で示す）。

```
db_name = prod1                                #The same as PRMYinit.ora
control_files = (/fs2/dbfs/cf1.f)
```

```
compatible = 8.1.6
log_archive_start = TRUE
log_archive_dest_1='LOCATION=/fs2/arc_dest/'
log_archive_dest_state_1 = ENABLE
log_archive_format = log_%t_%s.arc
standby_archive_dest = /fs2/stdby/
db_file_name_convert = ('/fs1/dbs','/fs2/dbs')
log_file_name_convert = ('/fs1/dbs','/fs2/dbs')
lock_name_space = standby1
audit_trail=FALSE
o7_dictionary_accessibility=FALSE
global_names=FALSE
db_domain=regress.rdbms.dev.us.oracle.com
remote_login_passwordfile = exclusive

# default parameters for instance 1
processes=30
sessions=30
transactions=21
transactions_per_rollback_segment=21
distributed_transactions=10
db_block_buffers=1000
db_files=200
shared_pool_size=10000000
```

## ステップ 6: 管理リカバリの準備としてのスタンバイ・インスタンスの起動

ネットワーク・ファイルおよびパラメータ・ファイルをすべて構成できたため、これでスタンバイ・データベースへのアーカイブを使用可能にできます。

1. 次のように、ORACLE\_SID 環境変数を、プライマリ・サイトの tnsnames.ora ファイルおよびスタンバイ・サイトの listener.ora ファイル内の SID パラメータと同じ値に設定します。

```
% setenv ORACLE_SID stdby1
```

2. インスタンスを起動します。

最初に、次の例で示すように、スタンバイ・データベースの制御ファイルをマウントせずに、スタンバイ・データベース・インスタンスを起動します。

```
SQL> CONNECT sys/change_on_install@standby1
SQL> STARTUP NOMOUNT PFILE=/fs3/oracle/dbs/initSTANDBY1.ora;
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

3. データ・ファイルを手動で改名します。

次に、DB\_FILE\_NAME\_CONVERT パラメータで取り込めないデータ・ファイルを改名するために SQL スクリプトを作成します。

```

ALTER DATABASE RENAME FILE /fs1/dbs2/tbs_42.f,
                             /fs1/dbs2/tbs_51.f,
                             /fs1/dbs2/tbs_52.f,
                             /fs1/dbs2/tbs_03.f,
                             /fs1/dbs3/tbs_14.f,
                             /fs1/dbs3/tbs_25.f,
                             /fs1/dbs3/tbs_33.f,
                             /fs1/dbs3/tbs_43.f,
                             /fs1/dbs3/tbs_53.f,
TO
                             /fs2/dbs/tbs_42.f,
                             /fs2/dbs/tbs_51.f,
                             /fs2/dbs/tbs_52.f,
                             /fs2/dbs/tbs_03.f,
                             /fs2/dbs/tbs_14.f,
                             /fs2/dbs/tbs_25.f,
                             /fs2/dbs/tbs_33.f,
                             /fs2/dbs/tbs_43.f,
                             /fs2/dbs/tbs_53.f
/

```

4. プライマリ・データベースのパラメータ・ファイルの変更を有効にします。

最後に、プライマリ・データベースのパラメータ・ファイルで行った変更を使用可能にし、スタンバイ・データベースがアーカイブ REDO ログの受取りを開始できるようにします。

```

SQL> CONNECT sys/change_on_install@prod1 as sysdba
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 OPTIONAL
REOPEN=180';
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = ENABLE;

```

## ステップ 7: ギャップ・シーケンス内のログの識別

プライマリ・データベースのパラメータ・ファイルの変更を有効にしたので、プライマリ・データベースはスタンバイのサービス名に対してアーカイブできるようになりました。しかし管理リカバリを実行するには、プライマリ・データベースのバックアップの作成以降、スタンバイ・データベースが最初のログを受け取るまでの間に行われた変更を含むログを適用して、スタンバイ・データベースを同期化する必要があります。

次の SQL スクリプトを作成し、スタンバイ・データベースで実行します。

```

SELECT high.thread#, "LowGap#", "HighGap#"
FROM
(
  SELECT thread#, MIN(sequence#)-1 "HighGap#"
  FROM
  (
    SELECT a.thread#, a.sequence#
    FROM

```

```

        (
            SELECT *
            FROM v$archived_log
        ) a,
        (
            SELECT thread#, MAX(sequence#) gap1
            FROM v$log_history
            GROUP BY thread#
        ) b
        WHERE a.thread# = b.thread#
        AND a.sequence# > gap1
    )
    GROUP BY thread#
) high,

(
    SELECT thread#, MIN(sequence#) "LowGap#"
    FROM
    (
        SELECT thread#, sequence#
        FROM v$log_history, v$datafile
        WHERE checkpoint_change# <= next_change#
        AND checkpoint_change# >= first_change#
    )
    GROUP BY thread#
) low
WHERE low.thread# = high.thread#;

```

この問合せの出力は次のとおりです。

```

SQL> @gap
THREAD#    LowGap#    HighGap#
-----
      1      250      252

```

これにより、管理リカバリを開始するためには、ログ順序 250、251 および 252 を適用してスタンバイ・データベースを同期化する必要があることがわかります。

## ステップ 8: ギャップ・シーケンス内のログのスタンバイのファイル・システムへの適用

スタンバイ・データベースに対するギャップ・シーケンスの問合せによって生成されたアーカイブ・ログのファイル名は、初期化パラメータ・ファイル内の LOG\_ARCHIVE\_DEST\_1 および LOG\_ARCHIVE\_FORMAT パラメータによって生成されたものです。プライマリ・サイトからスタンバイ・サイトにログを転送する前に、正しいファイル名を判断する必要があります。

最初に、プライマリ・データベースでアーカイブされた、ギャップ中のログのファイル名を判別します。SQL\*Plus を使用してプライマリ・データベースに接続した後、次の SQL 問合せを発行して名前を入手します。

```
SQL> CONNECT sys/change_on_install@prod1
SQL> SELECT name FROM v$archived_log WHERE sequence# IN (250,251,252);
```

```
NAME
```

```
-----
/fs1/arc_dest/log_1_250.arc
/fs1/arc_dest/log_1_251.arc
/fs1/arc_dest/log_1_252.arc
```

この場合のギャップ・シーケンスは log\_1\_250.arc、log\_1\_251.arc および log\_1\_252.arc から構成されています。スタンバイ・データベースのパラメータ・ファイル内の LOG\_ARCHIVE\_DEST\_1 および LOG\_ARCHIVE\_FORMAT の設定は次のようになります。

```
LOG_ARCHIVE_DEST_1 = 'LOCATION=/fs2/arc_dest/'
LOG_ARCHIVE_FORMAT = log_%t_%s.arc
```

スタンバイ・サイトの LOG\_ARCHIVE\_DEST\_1 および LOG\_ARCHIVE\_FORMAT 初期化パラメータの値に従って改名しながら、プライマリのファイル・システムからスタンバイのファイル・システムにギャップ・シーケンスのログを移動します。

```
% cp /fs1/arc_dest/log_1_250.arc /fs2/arc_dest/log_1_250.arc
% cp /fs1/arc_dest/log_1_251.arc /fs2/arc_dest/log_1_251.arc
% cp /fs1/arc_dest/log_1_252.arc /fs2/arc_dest/log_1_252.arc
```

## ステップ 9: ギャップ・シーケンス内のログのスタンバイ・データベースへの適用

これで、RECOVER AUTOMATIC STANDBY DATABASE 文を使用してギャップ・シーケンスのログを適用できます。この文は、LOG\_ARCHIVE\_DEST\_1 および LOG\_ARCHIVE\_FORMAT パラメータの値を使用して、ターゲットのファイル名を構成します。ギャップ・シーケンスのログの適用により、スタンバイ・データベースはプライマリ・データベースと同期するため、スタンバイ・データベースを管理リカバリ・モードにすることができます。

SQL\*Plus でスタンバイ・データベースに接続中に、AUTOMATIC オプションを使用してデータベースのリカバリを行います。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
ORA-00279: 変更 35083 (08/16/1999 14:08:37 で生成) にはスレッド番号 2 が必要です。
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
CANCEL
Media recovery cancelled.
SQL>
```

ギャップ・シーケンスのログのリカバリ後に、Oracle では存在しないログの名前のプロンプトを表示します。これは、リカバリ処理が、プライマリ・データベースによって STANDBY サービスにアーカイブされたログを認識できないためです。この時点でリカバリの取消しを行います。

```
SQL> CANCEL
```

### ステップ 10: スタンバイ・データベースの管理リカバリ・モード設定

これで、RECOVER MANAGED STANDBY DATABASE 文を使用して管理リカバリを使用可能にできます。Oracle が指定された分数だけ待ってから、スタンバイ・データベースの制御ファイルのディレクトリに要求されたアーカイブ・ログ・エントリを書き込むように、RECOVER 文の TIMEOUT オプションを使用して、20 分という時間間隔を指定します。要求されたアーカイブ・ログ・エントリが、指定された時間内にスタンバイ・データベースの制御ファイル・ディレクトリに書き込まれない場合は、リカバリ操作は取り消されます。

SQL\*Plus を使用してスタンバイ・データベースに接続中に、スタンバイ・データベースを管理リカバリ・モードにします。

```
SQL> RECOVER MANAGED STANDBY DATABASE TIMEOUT 20;
```

これにより、Oracle は管理リカバリを開始します。プライマリ・データベースからスタンバイ・サイトに REDO ログがアーカイブされるたびに、スタンバイ・データベースはログを自動的に適用します。

## シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成

このシナリオでは、リモート・ホスト上にスタンバイ・データベース STANDBY1 を作成します。このシナリオでは次のような仮定を行っています。

- 一貫性バックアップを実行できる
- プライマリ・データベースとスタンバイ・データベースの接続に TCP/IP を使用する
- スタンバイ・データベースは管理リカバリ環境の一部
- リモート・ホスト名は STBYHOST
- PRMYinit.ora はプライマリ・データベースの初期化パラメータ・ファイル
- STBYinit.ora はスタンバイ・データベースの初期化パラメータ・ファイル

### ステップ 1: プライマリ・データベースのデータ・ファイルのバックアップ

スタンバイ・データベースのベースとするバックアップを作成します。

1. プライマリ・データベースを問い合せてデータ・ファイルを確認します。SQL\*Plus を起動し、次の例で示すように、V\$DATAFILE ビューを問い合せて、プライマリ・データベースのデータ・ファイルのリストを入手します。

```
SQL> SELECT name FROM v$datafile;  
NAME
```

```
-----
/vobs/oracle/dbs/dbf_1.f
1 row selected.
```

2. プライマリ・データベースをシャットダウンし、データ・ファイルの一貫性バックアップを作成します。

```
SQL> SHUTDOWN IMMEDIATE;
```

3. 次の例で示すように、プライマリ・データベースのデータ・ファイルを一時的な場所 (/backup) にコピーします。

```
% cp /vobs/oracle/dbs/dbf_1.f /backup
```

4. 次のようにプライマリ・データベースをリオープンします。

```
SQL> STARTUP PFILE=PRMyinit.ora;
```

## ステップ 2: スタンバイ・データベースの制御ファイルの作成

1. スタンバイ・データベースの制御ファイルを作成する前に、プライマリ・データベースが ARCHIVELOG モードになっていることと、自動アーカイブが使用可能になっていることを確認します。ARCHIVE LOG LIST 文を発行します。

```
SQL> ARCHIVE LOG LIST
```

ARCHIVE LOG LIST 文の出力で "No Archive Mode" と表示された場合は、次のステップを実行します。

- a. 次のようにプライマリ・データベースをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

- b. プライマリ・データベース・インスタンスをオープンせずに、起動およびマウントします。

```
SQL> STARTUP MOUNT PFILE=PRMyinit.ora;
```

- c. ログ・アーカイブ・モードを次のように設定します。

```
SQL> ALTER DATABASE ARCHIVELOG;
```

- d. プライマリ・データベースをオープンします。

```
SQL> ALTER DATABASE OPEN;
```

2. 次の文を発行し、スタンバイ・データベースの制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/backup/stbycf.f'
```

スタンバイ・サイトへのコピーを容易にするために、スタンバイ・データベースの制御ファイルとプライマリ・データベースのデータ・ファイルは、プライマリ・サイトの同じ一時的な場所に置かれています。

### ステップ 3: データ・ファイルおよび制御ファイルのスタンバイ・サイトへの転送

次の例で示すように、プライマリ・データベースのデータ・ファイルおよびスタンバイの制御ファイルを、プライマリ・サイトの一時的な場所からスタンバイ・サイトにコピーします。

```
% rcp /backup/* STBYHOST:/fs2/oracle/stdby
```

### ステップ 4: ネットワーク・ファイルの構成

このシナリオでは、プライマリ・データベースとスタンバイ・データベースの接続に TCP/IP ネットワーク・プロトコルを使用するものと想定しています。このステップでは、次のパラメータ・ファイルを編集します。

- プライマリ・サイトの tnsnames.ora ファイル
- スタンバイ・サイトの listener.ora ファイル

1. tnsnames.ora ファイルを構成します。

tnsnames.ora ファイルを編集し、次の書式でエントリを 1 つ追加する必要があります。

```
standby_service_name = (DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcp) (PORT=port_number) (HOST=host_name))
  (CONNECT_DATA= (SID=standby_sid)))
```

tnsnames.ora ファイルは、通常はプライマリ・サイトの \$ORACLE\_HOME/network/admin ディレクトリに入っています。次の例で示すように、standby\_service\_name、port\_number、host\_name および standby\_sid を適切な値で置換します。

```
standby1 = (DESCRIPTION= (ADDRESS= (PROTOCOL=tcp)
  (PORT=5112) (HOST=STBYHOST))
  (CONNECT_DATA= (SID=stdby1)))
```

2. listener.ora ファイルを構成します。

listener.ora ファイルを編集し、次の書式で 2 つのエントリを追加する必要があります。

```
STANDBY_LISTENER = (ADDRESS_LIST= (ADDRESS= (PROTOCOL=tcp)
  (PORT=port_number) (HOST=host_name)))
```

```
SID_LIST_STANDBY_LISTENER = (SID_LIST=
```



```
(SID_DESC=(SID_NAME=standby_sid) (ORACLE_HOME=/oracle_home)))
```

listener.ora ファイルは、通常はスタンバイ・サイトの \$ORACLE\_HOME/network/admin ディレクトリに入っています。次の例で示すように、port\_number、host\_name、standby\_sid および oracle\_home を適切な値で置換します。

```
STDBY1_LISTENER = (ADDRESS_LIST=(ADDRESS=(PROTOCOL=tcp)
(PORT=5112) (HOST=STBYHOST)))
```

```
SID_LIST_STDBY1_LISTENER = (SID_LIST=
(SID_DESC=(SID_NAME=standby1) (ORACLE_HOME=/oracle)))
```

listener.ora ファイル内の SID\_NAME が tnsnames.ora ファイル内の SID と一致することを確認します。また、2つのファイルで PORT および HOST の値が同じになっていることを確認します。新しいリスナーを追加したり、既存のリスナーに新しいアドレスを追加するためのオプションもあります。

## ステップ 5: スタンバイ・サイトでのリスナーの起動

ステップ 4 で追加した 2 つのエントリは、ポート 5112 でスタンバイ・データベース stdby1 への接続のリスニングを行うリスナーを定義します。このステップでは、stdby1\_listener リスナーを起動する必要があります。たとえば、

```
% lsnrctl start stdby1_listener;
```

## ステップ 6: スタンバイの初期化パラメータ・ファイルの構成

1. プライマリ・データベースの初期化パラメータ・ファイルを、プライマリ・サイトからスタンバイ・サイトにコピーします。スタンバイ・サイトで、次のようなコマンドを発行します。

```
% rcp /vobs/oracle/dbs/PRMyinit.ora STBYHOST:/fs2/oracle/stdby/STBYinit.ora
```

2. スタンバイの初期化パラメータ・ファイル (STBYinit.ora) を編集します。次のパラメータを編集します。

パラメータ	値
CONTROL_FILES	stbycf.f
STANDBY_ARCHIVE_DEST	/fs2/oracle/stdby/
LOG_ARCHIVE_DEST_1	/fs2/oracle/stdby/
LOG_ARCHIVE_FORMAT	stdby_%t_%s
DB_FILE_NAME_CONVERT	(' /vobs/oracle/dbs','/fs2/oracle/stdby')
LOG_FILE_NAME_CONVERT	(' /vobs/oracle/dbs','/fs2/oracle/stdby')

パラメータ	値
LOG_ARCHIVE_START	FALSE

STBYinit.ora ファイルは次のようになります。

```
#
#parameter file STBYinit.ora
#

db_name=prod1                                #The same as PRMYinit.ora

# The following two parameters have been changed from PRMYinit.ora
control_files=/fs2/oracle/stdby/stbycf.f
lock_name_space=stdby1

# The following parameters are the same as PRMYinit.ora
audit_trail=FALSE
o7_dictionary_accessibility=FALSE
global_names=FALSE
db_domain=regress.rdbms.dev.us.oracle.com
commit_point_strength=1

processes=30
sessions=30
transactions=21
transactions_per_rollback_segment=21
distributed_transactions=10
db_block_buffers=100
shared_pool_size=4000000
ifile=/vobs/oracle/work/tkinit.ora # Verify that file exists on the standby site
                                   # and that the file specification is valid

# specific parameters for standby database
log_archive_format = stdby_%t_%s.arc
standby_archive_dest=/fs2/oracle/stdby/
log_archive_dest_1='LOCATION=/fs2/oracle/stdby/'
db_file_name_convert=(' /vobs/oracle/dbs', '/fs2/oracle/stdby')
log_file_name_convert=(' /vobs/oracle/dbs', '/fs2/oracle/stdby')
log_archive_start=FALSE
log_archive_trace=127
```

## ステップ 7: スタンバイの初期化パラメータ・ファイルのコピー

1. 次のコマンドを発行し、STBYinit.ora ファイルのコピーを作成します。

```
% cp STBYinit.ora Failover.ora
```

スタンバイ・データベース stdby1 にフェイルオーバーしたときに、新しいプライマリ・データベースの初期化パラメータ・ファイルとして Failover.ora ファイルを使用できるように、Failover.ora を編集します。LOG\_ARCHIVE\_DEST\_*n* パラメータで正しい値が使用されていることを確認してください。

2. スタンバイ・データベースへのフェイルオーバーが発生した場合に備えて、スタンバイ・サイトの tnsnames.ora ファイルを編集します。tnsnames.ora ファイルの構成方法は、ステップ 4 を参照してください。

## ステップ 8: スタンバイ・データベースの起動

スタンバイ・データベースを起動し、アーカイブを使用可能にします。

1. 次のように、ORACLE\_SID 環境変数を、プライマリ・サイトの tnsnames.ora ファイルおよびスタンバイ・サイトの listener.ora ファイル内の SID パラメータと同じ値に設定します。

```
% setenv ORACLE_SID stdby1
```

2. SQL\*Plus を起動します。

```
SQL> CONNECT sys/sys_password as sysdba
```

3. データベースをマウントせずにスタンバイ・データベース・インスタンスを起動します。

```
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
```

4. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

## ステップ 9: プライマリの初期化パラメータ・ファイルの構成

1. PRMYinit.ora ファイルに次のエントリを追加し、アーカイブ先を指定します。

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 MANDATORY REOPEN=60'
```

2. PRMYinit.ora ファイルに次のエントリを追加し、アーカイブ先の状態を使用可能にします。

```
LOG_ARCHIVE_DEST_STATE_2 = ENABLE
```

3. このステップで設定した初期化パラメータが有効になるように、次の文を発行します。

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2='SERVICE=standby1 MANDATORY REOPEN=60';
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;
```

## ステップ 10: ギャップ・シーケンス内のログの適用

1. プライマリ・データベースで、次のように現行の REDO ログをアーカイブします。
2. スタンバイ・データベースで次の SQL スクリプトを実行し、ギャップ・シーケンス内のアーカイブ REDO ログを識別します。

```
SELECT high.thread#, "LowGap#", "HighGap#"
FROM
(
  SELECT thread#, MIN(sequence#)-1 "HighGap#"
  FROM
  (
    SELECT a.thread#, a.sequence#
    FROM
    (
      SELECT *
      FROM v$archived_log
    ) a,
    (
      SELECT thread#, MAX(next_change#) gap1
      FROM v$log_history
      GROUP BY thread#
    ) b
    WHERE a.thread# = b.thread#
    AND a.next_change# > gap1
  )
  GROUP BY thread#
) high,

(
  SELECT thread#, MIN(sequence#) "LowGap#"
  FROM
  (
    SELECT thread#, sequence#
    FROM v$log_history, v$datafile
    WHERE checkpoint_change# <= next_change#
    AND checkpoint_change# >= first_change#
  )
  GROUP BY thread#
) low
```

```
WHERE low.thread# = high.thread#;
```

THREAD#	LowGap#	HighGap#
1	90	92

3. プライマリ・データベースで V\$ARCHIVED\_LOG ビューへの問合せを次のように実行し、ギャップ・シーケンス内のログのファイル名を入手します。

```
SELECT name FROM v$archived_log
WHERE thread#=1 AND sequence#<=92 AND sequence#>=90;
```

```
NAME
-----
/vobs/oracle/dbs/r_1_90.arc
/vobs/oracle/dbs/r_1_91.arc
/vobs/oracle/dbs/r_1_92.arc
```

4. ギャップ・シーケンス内のログを、プライマリ・データベースからスタンバイ・データベースに次のように転送します。

```
% rcp /vobs/oracle/dbs/r_1_90.arc STBYHOST:/fs2/oracle/stdby/stdby_1_90.arc
% rcp /vobs/oracle/dbs/r_1_91.arc STBYHOST:/fs2/oracle/stdby/stdby_1_91.arc
% rcp /vobs/oracle/dbs/r_1_92.arc STBYHOST:/fs2/oracle/stdby/stdby_1_92.arc
```

5. スタンバイ・データベースで次の SQL 文を発行し、ギャップ・シーケンス内のログを手動で適用します。

```
SQL> RECOVER STANDBY DATABASE;
```

## ステップ 11: スタンバイ・データベースの管理リカバリ・モード設定

スタンバイ・データベースで次の SQL 文を発行し、管理リカバリを使用可能にします。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

## シナリオ 3: プライマリ・データベースの物理的変更への対応

このシナリオでは、プライマリ・データベースで物理的変更が行われたときに必要な手順を説明します。次のトピックを説明します。

- [プライマリ・データベースへのデータ・ファイルの追加](#)
- [プライマリ・データベースのデータ・ファイルの改名](#)
- [プライマリ・データベースのデータ・ファイルまたは表領域の削除](#)

- オンライン REDO ログの追加または削除
- 制御ファイルの変更
- スタンバイ・データベースの制御ファイルのリフレッシュ
- スタンバイ・データベースの再構築が必要な物理的変更

## プライマリ・データベースへのデータ・ファイルの追加

プライマリ・データベースにデータ・ファイルを追加したときには、スタンバイ・データベースにも対応するデータ・ファイルを追加して一貫性を保つ必要があります。そうしておかないと、プライマリ・データベース内の新しいデータ・ファイルに関連するオンライン REDO ログに含まれる変更は、スタンバイ・データベースに適用されません。

状況	影響
プライマリ・データベースに新しいデータ・ファイルを作成	スタンバイ・データベースの中に新しいデータ・ファイルを作成します。
既存のデータ・ファイルを別のデータベースからプライマリ・データベースに転送	プライマリ・データベースからスタンバイ・データベースにデータ・ファイルをコピーします。データ・ファイルをコピーするときには、データ・ファイルの内容をメンテナンスします。

### スタンバイ・データベースの中に新しいデータ・ファイルを作成

1. プライマリ・データベースで次のようにデータ・ファイルを作成するものとします。

```
SQL> CREATE TABLESPACE tbs_4 DATAFILE 'tbs_4.f' SIZE 2M;
```

2. スタンバイ・データベースを起動します。

```
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
```

3. プライマリ・サイトで、次のように強制的にログ・スイッチします。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

4. スタンバイ・データベースを管理リカバリ・モードにします。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

プライマリ・サイトにあるデータ・ファイルがスタンバイ・サイトに存在しないため、この SQL 文により、Oracle はアーカイブ REDO ログの適用を停止します。REDO ログをアーカイブしようとする、次のようなメッセージが表示されます。

ORA-00283: エラーによってリカバリ・セッションは取り消されました。

ORA-01157: データ・ファイル 4 を識別 / ロックできません。DEWR トレース・ファイルを参照してください。

ORA-01110: データ・ファイル 4: '/vobs/oracle/dbs/stdby/tbs\_4.f'

このエラー・メッセージは、データ・ファイルがスタンバイ・データベースの制御ファイルに追加されているが、データ・ファイルがまだ作成されていないことを示しています。

5. スタンバイ・サイトで、次の文を発行して新しいデータ・ファイルを作成します。

```
SQL> ALTER DATABASE CREATE DATAFILE '/vobs/oracle/dbs/stdby/tbs_4.f'  
      AS '/vobs/oracle/dbs/stdby/tbs_4.f';
```

6. スタンバイ・サイトで、次の文を発行してアーカイブ REDO ログの適用を再開します。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

## プライマリ・データベースからデータ・ファイルをコピー

プライマリ・データベースで、別のデータベースからプライマリ・データベースにデータ・ファイルを転送したものとします。

1. プライマリ・データベースで、新しいデータ・ファイルをバックアップします。

```
SQL> ALTER TABLESPACE tbs_4 BEGIN BACKUP;
```

2. データ・ファイルを一時ディレクトリにコピーします。

```
% cp tbs_4.f /backup
```

3. プライマリ・データベースで次の文を発行します。

```
SQL> ALTER TABLESPACE tbs_4 END BACKUP;
```

4. バックアップの一時ディレクトリからスタンバイ・サイトに tbs\_4.f データ・ファイルをコピーします。

```
% cp /backup/tbs_4.f STBYHOST:/vobs/oracle/dbs/stdby
```

5. スタンバイ・サイトで、次の文を発行してアーカイブ REDO ログの適用を再開します。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

## プライマリ・データベースにデータ・ファイルを追加した後でスタンバイ・データベースの制御ファイルをリフレッシュ

プライマリ・データベースにデータ・ファイルを追加し、スタンバイ・データベースで対応するデータ・ファイルを作成またはコピーすると、V\$DATAFILE ビューの状態フィールドには、"RECOVER" という値が含まれるようになります。"RECOVER" という値は、データ・

ファイルのリカバリが必要であることを示します。この場合の "RECOVER" 値は正しくありません。V\$DATAFILE ビューの状態フィールドに正しい値を入れるために、スタンバイ・データベースの制御ファイルのリフレッシュする必要があります。追加情報は 5-26 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

---

**注意：** スタンバイ・データベースの制御ファイルのリフレッシュすることにより正確な値が得られる一方で、これにより管理リカバリが無効になることがあります。管理リカバリが無効になった場合の唯一のオプションは手動リカバリです。

---

## プライマリ・データベースのデータ・ファイルの改名

プライマリ・データベースの 1 つまたは複数のデータ・ファイルを改名した場合は、スタンバイ・データベースでも対応するデータ・ファイルを改名する必要があります。

1. プライマリ・サイトでデータ・ファイルを改名します。

```
SQL> ALTER TABLESPACE tbs_4 OFFLINE;
SQL> ALTER TABLESPACE tbs_4 RENAME DATAFILE 'tbs_4.f' TO 'tbs_x.f';
SQL> ALTER DATABASE RECOVER TABLESPACE tbs_4;
SQL> ALTER TABLESPACE tbs_4 ONLINE;
```

2. プライマリ・サイトで、次の文を発行し、スタンバイ・データベースの制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'stbycf.f';
```

3. 次の文を発行し、スタンバイ・データベースでオンライン REDO ログがすべて適用されていることを確認します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

4. IMMEDIATE オプションを指定してスタンバイ・データベースをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

5. プライマリ・サイトからスタンバイ・サイトにスタンバイ・データベースの制御ファイルをコピーし、スタンバイ・サイトの既存の制御ファイルを上書きします。

```
% rcp stbycf.f STBYHOST:/fs2/oracle/stdby/
```

6. スタンバイ・サイトでデータ・ファイルを改名します。

```
% mv tbs_4.f tbs_x.f
```

7. データベースをマウントせずにスタンバイ・データベース・インスタンスを起動します。



```
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
```

8. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

9. ギャップ・シーケンス内のログを識別し、適用します。

10. スタンバイ・データベースで次の文を発行し、管理リカバリを使用可能にします。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

スタンバイ・サイトで対応するファイルを改名せずにスタンバイ・データベースの制御ファイルをリフレッシュしようとする、スタンバイ・データベースは改名されたデータ・ファイルを使用しようとしませんが、改名されたデータ・ファイルを見つけることができません。このため、次のようなエラー・メッセージが表示されます。

ORA-00283: エラーによってリカバリ・セッションは取り消されました。

ORA-01157: データ・ファイル 4 を識別 / ロックできません。DBWR トレース・ファイルを参照してください。

ORA-01110: データ・ファイル 4: '/vobs/oracle/dbs/stdby/tbs\_x.f'

## プライマリ・データベースのデータ・ファイルまたは表領域の削除

プライマリ・データベースの 1 つまたは複数のデータ・ファイルまたは表領域を削除した場合は、スタンバイ・データベースでも対応するデータ・ファイルまたは表領域を削除する必要があります。また、スタンバイ・データベースの制御ファイルもリフレッシュする必要があります。

1. 次のように、プライマリ・サイトでデータ・ファイルを削除します。

```
SQL> DROP TABLESPACE tbs_4;
% rm tbs_4.f
```

2. スタンバイ・データベースの制御ファイルをリフレッシュします。ステップは、5-26 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

3. 次のように、スタンバイ・サイトで対応するデータ・ファイルを削除します。

```
% rm tbs_4.f
```

## オンライン REDO ログの追加または削除

DBA が使用できるチューニングの方法の 1 つに、オンライン REDO ログのサイズおよび数を変更する方法があります。このため、プライマリ・サイトでオンライン REDO ログを追加または削除したときには、スタンバイ・データベースの制御ファイルをリフレッシュすることが重要になります。

1. オンライン REDO ログを追加または削除するには次のようにします。

```
SQL> ALTER DATABASE ADD LOGFILE 'prmy3.log' SIZE 100K;
```

または

```
SQL> ALTER DATABASE DROP LOGFILE 'prmy3.log';
```

2. スタンバイ・データベースの制御ファイルをリフレッシュします。ステップは、5-26 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

## 制御ファイルの変更

プライマリ・データベースで CREATE CONTROLFILE 文を使用して 1 つまたは複数のデータベース・パラメータを変更したときには、スタンバイ・データベースの制御ファイルをリフレッシュする必要があります。この文を使用して変更できるパラメータには、REDO ログ・ファイル・グループの最大数、REDO ログ・ファイル・メンバーの最大数、アーカイブ REDO ログ・ファイルの最大数、データ・ファイルの最大数またはデータベースを同時にマウントおよびオープンできるインスタンスの最大数が含まれます。

1. プライマリ・データベースで次のように制御ファイルを変更します。

```
SQL> CREATE CONTROLFILE REUSE DATABASE stby1 NORESETLOGS
      LOGFILE 'prmy1.log' SIZE 100K, 'prmy2.log' SIZE 100K
      DATAFILE 'dbf_1.f' SIZE 10M MAXLOGFILES 52 ARCHIVELOG;
```

2. スタンバイ・データベースの制御ファイルをリフレッシュします。ステップは、5-26 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

CREATE CONTROLFILE 文の RESETLOGS 句を使用すると、スタンバイ・データベースは無効になります。スタンバイ・データベースが無効になった場合には、スタンバイ・データベースを再作成する以外に方法がありません。

## スタンバイ・データベースの制御ファイルのリフレッシュ

プライマリ・データベースに物理的変更を行った場合には、スタンバイ・データベースで対応する変更を行うことに加えて、スタンバイ・データベースの制御ファイルをリフレッシュする必要があることもあります。オンライン REDO ログは、プライマリ・データベースの制御ファイルの変更は記録しません。この項では、スタンバイ・データベースの制御ファイルをリフレッシュする必要のある状況と、そのためのステップを説明します。

次の場合には、スタンバイ・データベースの制御ファイルをリフレッシュする必要があります。

- データ・ファイルを改名した場合
- データ・ファイルを削除した場合

- 表領域を削除した場合
- 1 つまたは複数のオンライン REDO ログを追加した場合
- 1 つまたは複数のオンライン REDO ログを削除した場合
- 制御ファイルを変更した場合

スタンバイ・データベースの制御ファイルとプライマリ・データベースの制御ファイルの同期を保つために、次のステップを実行します。

1. プライマリ・サイトで、次の文を発行し、スタンバイ・データベースの制御ファイルを作成します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'stbycf.f';
```

2. スタンバイ・データベースが管理リカバリ・モードの場合には、次の文を発行してリカバリの取消しを行う必要があります。

```
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL;
```

3. IMMEDIATE オプションを指定してスタンバイ・データベースをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

4. スタンバイ・データベースの制御ファイルを、プライマリ・サイトからスタンバイ・サイトにコピーし、スタンバイ・サイトの既存の制御ファイルを上書きします。

```
% rcp stbycf.f STBYHOST:/fs2/oracle/stdby/
```

5. データベースをマウントせずにスタンバイ・データベース・インスタンスを起動します。

```
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
```

6. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

7. ギャップ・シーケンス内のログを識別し、適用します。

8. スタンバイ・データベースで次の文を発行し、管理リカバリを使用可能にします。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

管理リカバリを使用可能にしようとした場合、次のようなエラー・メッセージが表示されることがあります。

ORA-00308: アーカイブ・ログ 'standby1' をオープンできません。

ORA-27037: ファイル・ステータスを取得できません。

SVR4 Error: 2: No such file or directory

Additional information: 3

ORA-01547: 警告: RECOVER は成功しましたが OPEN RESETLOGS が次のエラーを受け取りました。

ORA-01152: ファイル 1 は十分に古いバックアップからリストアされていません。

ORA-01110: データ・ファイル 1: '/vobs/oracle/dbs/stdby/tbs\_1.f'

ORA-00308 エラーが表示された場合には、次の文を発行してリカバリの取消しを行います。

```
SQL> CANCEL
```

これらのエラー・メッセージは、ギャップ・シーケンス内の 1 つまたは複数のログが正常に適用されなかった場合に発行されます。このようなエラーが発行された場合は、ステップ 7 および 8 を繰り返してください。

## スタンバイ・データベースの再構築が必要な物理的変更

プライマリ・データベースで行った物理的変更によって、スタンバイ・データベースが無効になることがあります。スタンバイ・データベースが無効になった場合は、再構築する以外に方法はありません。

ALTER DATABASE 文の次の句は、スタンバイ・データベースを無効にします。

- CLEAR UNARCHIVED LOGFILE
- OPEN RESETLOGS

## シナリオ 4: NOLOGGING 句の指定後のリカバリ

SQL 文の中には、REDO ログ・ファイルにデータベース操作をログしないことを指示したり、NOLOGGING 句を指定できるものがあります。ユーザーが NOLOGGING 句を指定した場合でも、REDO ログ・レコードは REDO ログに書き込まれます。しかし、REDO ログ・ファイルがスタンバイ・サイトに転送され、スタンバイ・データベースに適用される時には、データ・ファイルの一部は使用不可能になり、リカバリ不能というマークが付けられます。スタンバイ・データベースをアクティブにするか、読取り専用オプションでスタンバイ・データベースをオープンし、"UNRECOVERABLE" とマークされたブロック範囲を読み取ろうとすると、次のようなエラー・メッセージが表示されます。

ORA-01578: ファイル番号 1, ブロック番号 2521 で ORACLE データ・ブロックに障害が発生しました。

ORA-01110: データ・ファイル 1: '/vobs/oracle/dbs/stdby/tbs\_1.f'

ORA-26040: データ・ブロックが NOLOGGING オプションを使用してロードされました。

NOLOGGING 句の指定後のリカバリのためには、REDO ログに記録されないデータを含むデータ・ファイルを、プライマリ・サイトからスタンバイ・サイトにコピーする必要があります。次のステップを実行します。

1. コピーする必要のあるデータ・ファイルを判別します。

- a. プライマリ・データベースで次の問合せを発行します。

```
SQL> SELECT name, unrecoverable_change# FROM v$datafile;
NAME                                     UNRECOVERA
-----
/vobs/oracle/dbs/tbs_1.f                5216
/vobs/oracle/dbs/tbs_2.f                0
/vobs/oracle/dbs/tbs_3.f                0
/vobs/oracle/dbs/tbs_4.f                0
4 rows selected.
```

- b. スタンバイ・データベースで次の問合せを発行します。

```
SQL> SELECT name, unrecoverable_change# FROM v$datafile;
NAME                                     UNRECOVERA
-----
/vobs/oracle/dbs/standby/tbs_1.f        5186
/vobs/oracle/dbs/standby/tbs_2.f        0
/vobs/oracle/dbs/standby/tbs_3.f        0
/vobs/oracle/dbs/standby/tbs_4.f        0
4 rows selected.
```

- c. プライマリ・データベースとスタンバイ・データベースの照会結果を比較します。

両方の照会結果で `unrecoverable_change#` 列の値を比較します。プライマリ・データベースの `unrecoverable_change#` 列の値が、スタンバイ・データベースの同じ列の値よりも大きい場合には、プライマリ・サイトからスタンバイ・サイトにデータ・ファイルをコピーする必要があります。

この例では、プライマリ・データベースの `unrecoverable_change#` の値は、`tbs_1.f` データ・ファイルではより大きくなっているため、`tbs_1.f` データ・ファイルをスタンバイ・サイトにコピーする必要があります。

2. プライマリ・サイトで、スタンバイ・サイトにコピーする必要のあるデータ・ファイルを次のようにバックアップします。

```
SQL> ALTER TABLESPACE system BEGIN BACKUP;
SQL> EXIT;
% cp tbs_1.f /backup
SQL> ALTER TABLESPACE system END BACKUP;
```

3. スタンバイ・データベースの新しい制御ファイルを作成します。

プライマリ・データベースで次の文を発行します。

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/backup/stbycf.f';
```

4. スタンバイ・データベースをシャットダウンします。

スタンバイ・データベースで次の文を発行します。

```
SQL> SHUTDOWN IMMEDIATE;
```

5. データ・ファイルとスタンバイ・データベースの制御ファイルを、プライマリ・サイトからスタンバイ・サイトに次のようにコピーします。

```
% rcp /backup/* STBYHOST:/fs2/oracle/stdby/
```

6. データベースをマウントせずにスタンバイ・データベース・インスタンスを起動します。

```
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
```

7. スタンバイ・データベースをマウントします。

```
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

8. ギャップ・シーケンス内のログを識別し、適用します。

9. スタンバイ・データベースで次の文を発行し、管理リカバリを使用可能にします。

```
SQL> RECOVER MANAGED STANDBY DATABASE;
```

管理リカバリを使用可能にしようとした場合、次のようなエラー・メッセージが表示されることがあります。

ORA-00308: アーカイブ・ログ 'standby1' をオープンできません。

ORA-27037: ファイル・ステータスを取得できません。

SVR4 Error: 2: No such file or directory

Additional information: 3

ORA-01547: 警告 : RECOVER は成功しましたが OPEN RESETLOGS が次のエラーを受け取りました。

ORA-01152: ファイル 1 は十分に古いバックアップからリストアされていません。

ORA-01110: データ・ファイル 1: '/vobs/oracle/dbs/stdby/tbs\_1.f'

ORA-00308 エラーが表示された場合には、次の文を発行してリカバリの取消しを行います。

```
SQL> CANCEL
```

これらのエラー・メッセージは、ギャップ・シーケンス内の 1 つまたは複数のログが正常に適用されなかった場合に発行されます。このようなエラーが発行された場合は、ステップ 8 および 9 を繰り返してください。

## シナリオ 5: 複数スタンバイ・データベース構成でのフェイルオーバー先のスタンバイ・データベースの決定

すべてのスタンバイ・データベースは、ただ1つのプライマリ・データベースと関連付けられます。しかし、1つのプライマリ・データベースは、複数のスタンバイ・データベースをサポートできます。このシナリオでは、複数のスタンバイ・データベース内のどのデータベースをアクティブにするかを決定するために必要な情報を識別します。

複数のスタンバイ・データベースを持つ構成で考慮する必要のある重要な事項の1つに、アーカイブ先を必須にするか、オプションにするかということがあります。次の表は、それぞれの宛先の利点および不利な点を示しています。

宛先	利点	不利な点
MANDATORY	すべてのアーカイブ REDO ログ・ファイルは、必須のアーカイブ先にアーカイブされます。スタンバイ・サイトでアーカイブ REDO ログ・ファイルを適用した後、スタンバイ・データベースが最新の状態になっているかどうか確認できます。さらに、最小限のデータの損失で、スタンバイ・データベースを新しい本番データベースとしてアクティブにできます。	場合によっては（ネットワーク障害など）、アーカイブ REDO ログ・ファイルは必須のアーカイブ先に達することができないため、REDO ログ・ファイルのアーカイブが停止することがあります。最悪の場合には、オンライン REDO ログ・ファイルが満杯で、アーカイブが行えない場合、プライマリ・データベース・インスタンスの稼動が停止することがあります。  オンライン REDO ログへのスイッチを行えないためにプライマリ・データベースが停止したかどうかを確認するために、次の SQL 問合せを発行できます。  <pre>SELECT decode(count(*),0,'NO','YES') "switch_possible" FROM v\$log WHERE archived='YES';</pre> 問合せの出力が "Yes" を示している場合には、ログ・スイッチは可能です。出力が "No" を示している場合には、ログ・スイッチは不可能です。

宛先	利点	不利な点
OPTIONAL	スタンバイ・サイトのオプションのアーカイブ先への REDO ログのアーカイブが割込みを受けても、プライマリ・データベースは通常の稼動を続けます。	場合によっては（ネットワーク障害など）、アーカイブ REDO ログ・ファイルはオプションのアーカイブ先に達することができないため、REDO ログ・ファイルのアーカイブが停止することがあります。スタンバイ・データベースに複数のギャップ・シーケンスがある可能性があります。自動アーカイブを再開するためには、ギャップを解決し、プライマリ・サイトからスタンバイ・サイトに REDO ログを手動で転送する必要があります。

複数のスタンバイ・データベースの構成では、次の推奨事項を考慮してください。

- 必須のアーカイブ先として、少なくとも1つのリモート・スタンバイ・データベースを指定してください。理想的には、最も安定したネットワーク・リンクを持ち、最も安定したシステム構成のスタンバイ・データベースを選択してください。
- ローカルのアーカイブ先を必須として指定してください。これにより、ネットワーク障害が発生した場合もプライマリ・データベースはローカルにアーカイブできます。

プライマリ・データベースがサンフランシスコにあり、次のような5つのスタンバイ・データベースをサポートしているとします。

スタンバイ	場所	タイプ	説明
1	ローカル・ディレクトリ	必須	アーカイブ REDO ログのローカル・コピー
2	サンフランシスコ	必須	プライマリ・サイトが物理的に損傷した場合は、このスタンバイ・データベースにフェイルオーバーします。このスタンバイ・サイトは、ローカル・エリア・ネットワークによってプライマリ・サイトと接続しています。
3	ボストン	オプション	サンフランシスコで災害が発生した場合は、このスタンバイ・データベースにフェイルオーバーします。
4	ロサンジェルス	オプション	このスタンバイ・サイトはアーカイブ REDO ログを受け取りますが、適用はしません。このタイプの構成の詳細は、5-46 ページの「シナリオ 9: 進行中のリカバリがないスタンバイ・データベース」を参照してください。



スタンバイ	場所	タイプ	説明
5	サンフランシスコ	オプション	このスタンバイ・サイトはアーカイブ REDO ログを受け取り、8 時間後にログを適用します。このタイプの構成の詳細は、5-50 ページの「シナリオ 10: 時間差を設定したスタンバイ・データベース」を参照してください。

プライマリ・サイトのあるサンフランシスコで災害が発生し、プライマリ・ホストが損傷を受けたとします。スタンバイ・データベースの 1 つをアクティブにする必要があります。複数のスタンバイ・データベース構成をセットアップしたデータベース管理者（DBA）に、どのスタンバイ・データベースにフェイルオーバーするかを相談できるとは限りません。このため、プライマリ・サイトおよび各スタンバイ・サイトに、災害リカバリ計画を備えておく必要があります。災害リカバリ・チームの各メンバーは、災害リカバリ計画を理解し、従うべき手順を知っておく必要があります。このシナリオでは、アクティブにするスタンバイ・データベースを決定するときに、決定者が必要とする情報の種類を識別します。災害リカバリ・チームに情報を伝達する方法の 1 つに、各スタンバイ・サイトに README ファイルを用意する方法があります。

各サイトの README ファイルでは、次の方法を説明する必要があります。

- DBA としてローカル・データベースにログオンする方法。
- スタンバイ・データベースのある各システムにログオンする方法。  
システム間にはファイアウォールがあることがあります。README ファイルでは、ファイアウォールを通り抜ける方法を説明する必要があります。
- DBA としてその他のデータベースにログオンする方法。
- 一番新しいスタンバイ・データベースを識別する方法。
- スタンバイ・データベースをアクティブにする方法。
- クライアント・アプリケーションが、元のプライマリ・データベースではなく、新しくアクティブにされたデータベースにアクセスするようにネットワーク設定値を構成する方法。

次の例は、サンプルの README ファイルの内容を示しています。

----- スタンバイ・データベース災害リカバリ ReadMe ファイル -----

警告：

\*\*\*\*\*  
この手順は、プライマリ・データベースの障害時にのみ、スタンバイ・データベースへのフェイルオーバーの担当者が実行してください。

このファイルに記載された手順を不必要に行った場合、データベース・システム全体が破損する可能性があります。

\*\*\*\*\*

複数スタンバイ・データベース構成：

No.	Location	Type	IP Address
1	San Francisco	Primary	128.1.124.25
2	San Francisco	Standby	128.1.124.157
3	Boston	Standby	136.132.1.55
4	Los Angeles	Standby	145.23.82.16
5	San Francisco	Standby	128.1.135.24

現在使用しているのは、Bostonにあるシステム No.3 です。

最新の状態にある、使用可能なスタンバイ・データベースにフェイルオーバーするために、次の手順を実行します。

1. ローカルのスタンバイ・データベースに DBA としてログオンします。

- a) 次のユーザー名とパスワードを使用してログオンします。

ユーザー名： Standby3  
パスワード： zkc722Khn

- b) 次のように SQL\*Plus を起動します。

% sqlplus

- c) 次のように DBA として接続します。

CONNECT sys/password AS SYSDBA

2. 可能な限り多くのリモート・システムと接続してください。最大 4 つのシステムに接続できます。システム No.4 にはファイアウォールがないため、直接接続できます。システム No.1、2 および 5 は同じファイアウォール・ホストを共有しています。最初にファイアウォール・ホストに接続してから、各システムに接続する必要があります。ファイアウォール・ホストの IP アドレスは 128.1.1.100 です。次のユーザー名およびパスワードを使用してください。

ユーザー名： Disaster  
パスワード： 821hsIW32

3. 次のユーザー名およびパスワードを使用し、可能な限り多くのリモート・システムにログオンしてください。

ログイン情報：

No.	Location	IP Address	username	password
-----	-----	-----	-----	-----

```

1  San Francisco  128.1.124.25  Oracle8i      sdd290Ec
2  San Francisco  128.1.124.157 Standby2      ei23nJHb
3                      (L o c a l)
4  Los Angeles    145.23.82.16  Standby4      23HHoe2a
5  San Francisco  128.1.135.24  Standby5      snc#$dnc

```

4. ログオン可能な各リモート・システムで、次のように SQL\*Plus を起動します。

```
% sqlplus
```

5. 次のように各リモート・データベースに接続します。

```
CONNECT sys/password AS SYSDBA
```

それぞれの場所の DBA パスワードは次のとおりです。

No.	Location	Password
1	San Francisco	x2dwlsd91
2	San Francisco	a239s1DAq
3	(L o c a l)	
4	Los Angeles	owKL(@as23
5	San Francisco	sad_KS13x

6. システム No.1 にログオンできる場合は、SQL\*Plus を起動し、次の文を発行します。

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP PFILE=PRMYinit.ora;
```

注：STARTUP 文を正常に実行できる場合には、プライマリ・データベースは破損していません。  
この手順を続行しないでください。

7. 接続できた各スタンバイ・データベース（このマシン上のこのデータベースを含む）で次の SQL 文を発行します。

```
SQL> RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP NOMOUNT PFILE=STBYinit.ora;
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
      *** press the Return key for each of the prompts
SQL> SELECT THREAD#, MAX(SEQUENCE#) FROM V$LOG_HISTORY GROUP BY THREAD#;
```

各スタンバイ・データベースからの問合せの結果を比較します。最も大きな順序番号を持つスタンバイ・データベースをアクティブにしてください。

8. 順序番号が最大のスタンバイ・データベースにフェイルオーバーします。

順序番号が最大のスタンバイ・データベースで、SQL\*Plus を起動し、次の SQL 文を発行します。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;  
SQL> SHUTDOWN IMMEDIATE;  
SQL> STARTUP PFILE=Failover.ora;
```

----- スタンバイ・データベース災害リカバリ ReadMe ファイルの終わり -----

## シナリオ 6: クライアント・アプリケーション・フェイルオーバーの構成

スタンバイ・データベースはアクティブ化すると本番データベースになるため、スタンバイ・データベースとして機能できなくなります。クライアント・アプリケーションは、元のプライマリ・データベースから、新しくアクティブにされた本番データベースに、接続をリダイレクトする必要があります。このシナリオでは、クライアントのフェイルオーバーをセットアップする次の方法を説明します。

- ローカル TNS 構成
- Oracle Names Server 構成
- 透過的アプリケーション・フェイルオーバー (TAF) 構成
- 手動のネットワーク構成

### ローカル TNS 構成

tnsnames.ora ファイルは、複数のアドレスを含めて構成できます。ローカル TNS 構成では、アドレスの少なくとも 1 つはスタンバイ・サイトのアドレスにする必要があります。各クライアント・サイトで tnsnames.ora ファイルを変更し、スタンバイ・サイトのアドレスが指定されていることを確認します。tnsnames.ora ファイルは、通常は \$ORACLE\_HOME/network/admin ディレクトリに入っています。1 つの TNS 名に複数のアドレスを割り当て、FAILOVER オプションを使用できます。複数のアドレスの設定方法および FAILOVER オプションの使用方法は、『Oracle8i Net8 管理者ガイド』を参照してください。次の例は、プライマリ・サイトのアドレスに加えて、スタンバイ・サイトのアドレスを含むエントリを示しています。

```
ProductDB = ( DESCRIPTION=  
              (FAILOVER=ON)  
              (LOAD_BALANCE=OFF)  
              (ADDRESS= (PROTOCOL=tcp) (PORT=1521) (HOST=PRMYHOST.foo.com))  
              (ADDRESS= (PROTOCOL=tcp) (PORT=1521) (HOST=STBYHOST.foo.com))  
              (CONNECT_DATA= (SID=db1))  
            )
```

この例では、プライマリ・データベースは PRMYHOST.foo.com にあり、スタンバイ・データベースは STBYHOST.foo.com にあります。クライアント・アプリケーションは ProductDB に接続すると、最初に PRMYHOST.foo.com への接続要求を送信します。応答がない場合には、クライアント・アプリケーションは STBYHOST.foo.com に別の接続要求を送信しようとします。プライマリ・データベースがダウンし、スタンバイ・データベースがアクティブになっているときには、クライアント・アプリケーションは新しい本番データベースに自動的に接続できます。

接続が確立された後でプライマリ・データベースに障害が発生した場合、クライアント・アプリケーションは、残りの要求を、新しくアクティブにされた本番データベースに自動的にリダイレクトすることはありません。ProductDB データベースへの接続を再度確立する必要があります。

## Oracle Names Server 構成

Oracle Names Server を使用する場合は、サーバー上で TNS 名の設定値を変更できます。1 つの TNS 名に複数のアドレスを割り当て、FAILOVER オプションを使用できます。Oracle Names Server での複数のアドレスの設定方法および FAILOVER オプションの使用方法は、『Oracle8i Net8 管理者ガイド』を参照してください。Oracle Names server で TNS 名を設定するための書式は、ローカルの tnsnames.ora ファイルの書式と同じです。このため、[ローカル TNS 構成](#)の例は、Oracle Names server 構成でも適用できます。

## 透過的アプリケーション・フェイルオーバー（TAF）構成

次の構成では、プライマリ・データベースがスタンバイ・データベースへフェイルオーバーした場合、クライアントはデータベース・サーバーに再接続する必要があります。

- ローカル TNS 構成
- Oracle Names Server 構成
- DNS 構成

ただし、Oracle Call Interface（OCI）クライアントを使用している場合には、透過的アプリケーション・フェイルオーバー（TAF）を使用できます。TAF は、接続の障害時に、データベースに自動的に再接続するためのアプリケーションの機能です。クライアント・アプリケーションがデータベース・トランザクション関与していない場合は、ユーザーはプライマリ・データベース・サーバーの障害に気が付かないことがあります。TAF の構成方法の詳細は、『Oracle8i パフォーマンスのための設計およびチューニング・ガイド』を参照してください。次の例は、ProductDB データベースおよび Standby1 データベースのアドレス情報を示しています。

```
ProductDB=( DESCRIPTION=
              (ADDRESS=(PROTOCOL=tcp) (PORT=1521) (HOST=PRMYHOST.foo.com))
              (CONNECT_DATA=(SID=db1) (FAILOVER_MODE=(BACKUP=Standby1)
                                                         (TYPE=session)
                                                         (METHOD=basic))))
```

```

    )
Standby1 =( DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (PORT=1521) (HOST=STBYHOST.foo.com) )
    (CONNECT_DATA= (SID=db1))
    )

```

イベントの順序

1. クライアント・アプリケーションが ProductDB データベースに接続します。
2. PRMYHOST.foo.com のプライマリ・データベースで障害が発生します。
3. スタンバイ・データベースが新しい本番データベースとしてアクティブにされます。
4. クライアント・アプリケーションは PRMYHOST.foo.com への接続に失敗すると、BACKUP パラメータの FAILOVER\_MODE 句で指定されているデータベースを使用し、STBYHOST.foo.com に自動的に接続します。

## 手動のネットワーク構成

クライアントが新しい本番データベースに自動的にフェイルオーバーできるように構成を設定するかわりに、スタンバイ・データベースをアクティブにした後に、ネットワーク設定を手動で変更できます。

ローカルの tnsnames.ora ファイルを変更できます。元の本番データベースを指している TNS 名を、新しくアクティブにされた本番データベースにリダイレクトします。たとえば、元の TNS 設定が次のようになっているとします。

```

ProductDB=( DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (PORT=1521) (HOST=PRMYHOST.foo.com) )
    (CONNECT_DATA= (SID=db1))
    )

```

PRMYHOST.foo.com 上のプライマリ・データベースに障害が発生したため、STBYHOST.foo.com のスタンバイ・データベースがアクティブにされ、これが新しい本番データベースになった場合は、tnsnames.ora ファイルを編集し、次のようにエントリを変更する必要があります。

```

ProductDB=( DESCRIPTION=
    (ADDRESS= (PROTOCOL=tcp) (PORT=1521) (HOST=STBYHOST.foo.com) )
    (CONNECT_DATA= (SID=db1))
    )

```

クライアント・アプリケーションを変更する必要はありません。本番データベースへのその後の接続は、STBYHOST.foo.com 上の新しい本番データベースに送信されます。

Oracle Names Server を使用する場合は、対応するエントリで同様の変更を行います。この Oracle Names Server を使用するすべてのクライアントは、その後の接続を、STBYHOST.foo.com 上の新しい本番データベースに送信します。

DNS サーバーで設定を変更することもできます。クライアントが本番データベースを見つけるために使用する、ドメイン名の設定を変更します。

たとえば、次のように仮定します。

項目	値
ドメイン名	ProductDB.foo.com
tnsnames.ora エントリ	ProductDB=(DESCRIPTION= (ADDRESS=(PROTOCOL=tcp) (PORT=1521) (HOST=ProductDB.foo.com)) (CONNECT_DATA=(SID=db1) )
ドメイン名の DNS サーバー・エントリ	ProductDB.foo.com IN A 136.1.23.15
STBYHOST.foo.com の IP アドレス	128.3.151.63

DNS サーバーのドメイン名の DNS サーバー・エントリを次のように変更します。

```
ProductDB.foo.com IN A 128.3.151.63
```

DNS の設定を変更した場合、すべてのクライアントが変更を即時に認識するわけではありません。古い DNS 設定がキャッシュ中に残っている場合があるため、クライアントは古い設定値を使用し続ける可能性があります。古い設定値を、新しい設定値で置換する必要があります。

## シナリオ 7: ネットワーク障害後のリカバリ

スタンバイ・データベースが管理リカバリ・モードになっている場合には、スタンバイ・データベースはプライマリ・データベースから受け取ったアーカイブ REDO ログを自動的に適用します。ネットワークがダウンした場合、プライマリ・データベースからスタンバイ・データベースへの自動アーカイブは停止します。

スタンバイ・データベースがオプションのアーカイブ先として指定されている場合には、プライマリ・データベースは通常稼働を続けます。

ネットワークが回復すると、プライマリ・データベースからスタンバイ・データベースへのアーカイブ REDO ログの自動アーカイブが再開されます。しかし、スタンバイ・データベースがオプションのアーカイブ先として指定され、プライマリ・サイトでログ・スイッチが発生した場合には、スタンバイ・データベースには、ネットワークがダウンしていた間のギャップ・シーケンスがあります。ギャップ・シーケンスが解決されない限り、スタンバイ・データベースはスタンバイ・サイトのアーカイブ REDO ログを適用できません。





2. ネットワーク障害を解決します。

ステップ 1 の照会結果は TNS: リスナーがありません。というエラーを示しています。スタンバイ・サイトのリスナーが起動されているか確認する必要があります。リスナーが停止している場合は、リスナーを起動します。

3. 可能であれば、プライマリ・データベースが停止しないように防止します。

ネットワークの問題を迅速に解決できず、スタンバイ・データベースが必須の宛先として指定されている場合には、次のいずれかを実行し、データベースの停止を防止する必要があります。

- 必須のアーカイブ先を使用禁止にします。

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = DEFER;
```

ネットワークの問題が解決すると、アーカイブ先を使用可能に戻すことができます。

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2 = ENABLE;
```

- アーカイブ先を必須からオプションに変更します。

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 OPTIONAL
REOPEN=60';
```

ネットワークの問題が解決すると、アーカイブ先をオプションから必須に戻すことができます。

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_2 = 'SERVICE=standby1 MANDATORY
REOPEN=60';
```

問題を解決する責任が他のユーザーにあることも考えられます。V\$ARCHIVE\_DESTビューを定期的に問い合せて、問題が解決されたかどうかを確認できます。

4. プライマリ・データベースで、現行の REDO ログをアーカイブします。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
```

5. ギャップ・シーケンス内のログを識別し、適用します。

ネットワークが回復し、新しいアーカイブ・ログがプライマリ・データベースからスタンバイ・データベースに転送されても、ギャップ・シーケンス内のログが手動で適用されない限り、新しいログがスタンバイ・データベースに適用されることはありません。

スタンバイ・データベースで次の SQL スクリプトを実行し、ギャップ・シーケンス内のアーカイブ REDO ログを識別します。

```
SELECT high.thread#, "LowGap#", "HighGap#"
FROM
(
```

```

SELECT thread#, MIN(sequence#)-1 "HighGap#"
FROM
(
  SELECT a.thread#, a.sequence#
  FROM
  (
    SELECT *
    FROM v$archived_log
  ) a,
  (
    SELECT thread#, MAX(next_change#) gap1
    FROM v$log_history
    GROUP BY thread#
  ) b
  WHERE a.thread# = b.thread#
  AND a.next_change# > gap1
)
GROUP BY thread#
) high,

(
  SELECT thread#, MIN(sequence#) "LowGap#"
  FROM
  (
    SELECT thread#, sequence#
    FROM v$log_history, v$datafile
    WHERE checkpoint_change# <= next_change#
    AND checkpoint_change# >= first_change#
  )
  GROUP BY thread#
) low
WHERE low.thread# = high.thread#;

```

THREAD#	LowGap#	HighGap#
1	90	92

6. プライマリ・データベースで V\$ARCHIVED\_LOG ビューへの問合せを次のように実行し、ギャップ・シーケンス内のログのファイル名を入手します。

```

SELECT name FROM v$archived_log
WHERE thread#=1 AND sequence#<=92 AND sequence#>=90;

NAME
-----
/vobs/oracle/dbs/r_1_90.arc

```

```
/vobs/oracle/dbs/r_1_91.arc
/vobs/oracle/dbs/r_1_92.arc
```

7. ギャップ・シーケンス内のログを、プライマリ・データベースからスタンバイ・データベースに次のように転送します。

```
% rcp /vobs/oracle/dbs/r_1_90.arc STBYHOST:/fs2/oracle/standby/standby_1_90.arc
% rcp /vobs/oracle/dbs/r_1_91.arc STBYHOST:/fs2/oracle/standby/standby_1_91.arc
% rcp /vobs/oracle/dbs/r_1_92.arc STBYHOST:/fs2/oracle/standby/standby_1_92.arc
```

8. スタンバイ・データベースで次の文を発行し、ギャップ・シーケンスを手動で適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

9. ギャップがなくなるまでステップ 5 から 8 を繰り返します。

ネットワーク障害により、スタンバイ・データベースに複数のギャップが生まれることがあります。ギャップは一度に 1 つずつ解決する必要があります。

すべてのアーカイブ REDO ログがプライマリ・データベースと同じシステム上にあるように、プライマリ・サイトのローカル・ディレクトリを必須のアーカイブ先として指定しておく必要があります。プライマリ・システムにアクセスできない場合、プライマリ・データベースが複数のスタンバイ・データベースを持つ構成の一部である場合には、他のスタンバイ・サイトにあるアーカイブ REDO ログを識別してみることができます。

## シナリオ 8: スタンバイ・データベースの再作成

このシナリオでは、スタンバイ・データベースにフェイルオーバーし、スタンバイ・データベースを通常の本番データベースとして使用し始めたケースを紹介します。しばらく後に、元の本番システムにフェイルバックし、本番データベースをスタンバイ・データベースに戻すことに決定します。一般的なステップは次のとおりです。

1. 元のプライマリ・サイトにスタンバイ・データベースを作成します。
2. 元のプライマリ・サイトのスタンバイ・データベースにフェイルオーバーします。
3. 元のスタンバイ・サイトに新しいスタンバイ・データベースを作成します。

詳細なステップは次のようになります。

1. 元のスタンバイの初期化パラメータ・ファイルを、次のように、元のスタンバイ・サイトから元のプライマリ・サイトにコピーします。

```
% rcp STBYinit.ora PRMYHOST:fallback.ora
```

fallback.ora ファイルは、元のプライマリ・サイトのスタンバイ・データベースのスタンバイ初期化パラメータ・ファイルになります。

2. 元のプライマリ・サイトで fallback.ora ファイルを構成します。次のパラメータを変更する必要があります。

パラメータ	値
LOCK_NAME_SPACE	fallback
LOG_ARCHIVE_FORMAT	r_%t_%s.arc
STANDBY_ARCHIVE_DEST	/vobs/oracle/dbs/
LOG_ARCHIVE_DEST_1	/vobs/oracle/dbs/
DB_FILE_NAME_CONVERT	(' /fs2/oracle/stdby','/vobs/oracle/dbs')
LOG_FILE_NAME_CONVERT	(' /fs2/oracle/stdby','/vobs/oracle/dbs')

fallback.ora ファイルは次のようになります。

```
#
#parameter file fallback.ora
#

db_name=prod1                #The same as PRMYinit.ora

control_files=/fs2/oracle/stdby/stbycf.f
lock_name_space=fallback;

audit_trail=FALSE
o7_dictionary_accessibility=FALSE
global_names=FALSE
db_domain=regress.rdbms.dev.us.oracle.com
commit_point_strength=1

processes=30
sessions=30
transactions=21
transactions_per_rollback_segment=21
distributed_transactions=10
db_block_buffers=100
shared_pool_size=4000000
ifile=/vobs/oracle/work/tkinit.ora

# specific parameters for standby database
log_archive_format = r_%t_%s.arc
standby_archive_dest=/vobs/oracle/dbs/
log_archive_dest_1='LOCATION=/vobs/oracle/dbs/'
log_archive_dest_state_1 = ENABLE
db_file_name_convert=(' /fs2/oracle/stdby','/vobs/oracle/dbs')
```

```
log_file_name_convert=('/fs2/oracle/stdby','/vobs/oracle/dbs')
log_archive_start=FALSE
log_archive_trace=127
```

3. 元のプライマリ・サイトで、プライマリの初期化パラメータ・ファイルを作成または変更します。

LOG\_ARCHIVE\_DEST\_1 および LOG\_ARCHIVE\_DEST\_STATE\_1 パラメータに適切な値を指定する必要があります。

**注意：**「シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成」のステップ 7 は、スタンバイ・データベースの初期化パラメータ・ファイルをコピーすることを提案しています。コピーを作成してある場合には、このステップでこのコピーを変更できます。

4. 元のスタンバイ・サイトで、本番データベースのデータ・ファイルをバックアップします。
5. 元のスタンバイ・サイトで、スタンバイ・データベースの制御ファイルを作成します。
6. 本番データベースのデータ・ファイルおよびスタンバイ・データベースの制御ファイルを、元のスタンバイ・サイトから元のプライマリ・サイトにコピーします。
7. これ以上変更されないように、次のように現行のオンライン REDO ログをアーカイブし、本番データベースをシャットダウンします。

```
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;
SQL> SHUTDOWN IMMEDIATE;
```

8. 本番データベースのデータ・ファイルのバックアップ作成後に生成されたアーカイブ REDO ログを、元のスタンバイ・サイトから元のプライマリ・サイトに次のようにコピーします。

```
% rcp /fs2/oracle/stdby/stdby_1_102.arc PRMYHOST:/vobs/oracle/dbs/r_1_102.arc
% rcp /fs2/oracle/stdby/stdby_1_103.arc PRMYHOST:/vobs/oracle/dbs/r_1_103.arc
```

9. 元のプライマリ・サイトで、スタンバイ・データベースを起動およびマウントします。

```
SQL> CONNECT sys/sys_password AS SYSDBA
SQL> STARTUP NOMOUNT PFILE=fallback.ora;
SQL> ALTER DATABASE MOUNT STANDBY DATABASE;
```

10. ギャップ・シーケンス内のログを適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

11. スタンバイ・データベースをアクティブにします。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
SQL> SHUTDOWN IMMEDIATE;
```

12. 元のプライマリ・サイトで本番データベースを起動します。

```
SQL> STARTUP PFILE=PRMYinit.ora;
```

13. クライアント・アプリケーションが本番データベースにアクセスできるようにネットワーク設定を構成します。
14. 元のスタンバイ・サイトに新しいスタンバイ・データベースを作成します。スタンバイ・データベースの作成のステップは、5-14 ページの「[シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成](#)」を参照してください。

## シナリオ 9: 進行中のリカバリがないスタンバイ・データベース

このシナリオでは、アーカイブ・オンライン REDO ログがスタンバイ・サイトに自動的に転送されても、スタンバイ・データベースに自動的に適用されないときに、DBA が行う必要のある作業について説明します。

たとえば、Web サーバーを実行するホスト上にスタンバイ・データベースがセットアップされているとします。このホストの作業負荷は非常に重いため、プライマリ・サイトから REDO ログが到着したときに自動的にログが適用されることによる、ホストの作業負荷の増加を避ける とします。ホストがアーカイブ REDO ログを受け取っても、ログを適用しないようにします。これにより、REDO ログの適用のためのシステム・リソースを節約できます。その後、問合せを実行するために読取り専用データベースとしてスタンバイ・データベースをオープンしたり、スタンバイ・データベースを本番データベースとしてアクティブにするときには、Web サーバーを停止し、すべての REDO ログを一度に適用できます。

このシナリオでは、スタンバイ・データベースがすでにセットアップされているものと想定しています。5-14 ページの「[シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成](#)」のステップ 1 から 10 では、スタンバイ・データベースのセットアップ方法を説明しています。

スタンバイ・データベースのリスナーを起動する必要があります。スタンバイ・データベースを起動およびマウントする必要がありますが、リカバリ・モードは使用できません。（これにより、REDO ログの適用のための時間およびリソースを節約できます。）

このシナリオは「[シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成](#)」と似ています。シナリオ 2 では、スタンバイ・データベースは管理リカバリ・モードになっています。管理リカバリ・モードのスタンバイ・データベースでは、プライマリ・サイトから受け取った REDO ログは自動的に適用されます。

## 進行中のリカバリがないスタンバイ・データベースの管理

アーカイブ REDO ログが即時にスタンバイ・データベースに適用されない場合には、スタンバイ・サイトでは進行中のリカバリがありません。この項では、進行中のリカバリがないスタンバイ・データベースの管理において DBA が考慮する必要のある作業を説明します。この項では、次のトピックを説明します。

- プライマリ・データベースへのデータ・ファイルの追加
- プライマリ・データベースの制御ファイルの変更の反映
- NOLOGGING 句が指定されていた場合の対処
- ネットワークの問題の対処

## プライマリ・データベースへのデータ・ファイルの追加

プライマリ・サイトにデータ・ファイルが追加されたときに、スタンバイ・サイトで実行するアクションを決定する必要があります。次の方法を使用できます。

- データ・ファイルが追加されたときに、プライマリ・サイトからスタンバイ・サイトに新しいデータ・ファイルをコピーする方法。
- プライマリ・サイトから新しいデータ・ファイルをコピーしない方法。

REDO ログをアーカイブしようとする、次のようなメッセージが表示されます。

```
ORA-00283: エラーによってリカバリ・セッションは取り消されました。  
ORA-01157: データ・ファイル 4 を識別 / ロックできません。DBWR トレース・ファイルを参照してください。  
ORA-01110: データ・ファイル 4: '/vobs/oracle/dbs/stby/tbs_4.f'
```

次の文を発行し、対応する新しいデータ・ファイルを作成します。

```
SQL> ALTER DATABASE CREATE DATAFILE '/vobs/oracle/dbs/stby/tbs_4.f' AS  
      '/vobs/oracle/dbs/stby/tbs_4.f';
```

## プライマリ・データベースの制御ファイルの変更の反映

プライマリ・サイトの制御ファイルが変更された場合に、スタンバイ・サイトでこの変更を反映したい場合には、次の文を発行し、スタンバイ・サイトで REDO ログを適用する必要があります。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

スタンバイ・サイトに REDO ログを適用した後、スタンバイ・データベースの制御ファイルをリフレッシュします。ステップは、5-26 ページの「[スタンバイ・データベースの制御ファイルのリフレッシュ](#)」を参照してください。

## NOLOGGING 句が指定されていた場合の対処

プライマリ・サイトで NOLOGGING 句が指定された後のリカバリのためには、次の作業を実行する必要があります。

1. 次の文を使用し、スタンバイ・サイトで REDO ログを適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

2. 影響を受けたデータ・ファイルおよびスタンバイ・データベースの制御ファイルをスタンバイ・サイトにコピーします。

使用する手順は、5-28 ページの「シナリオ 4: NOLOGGING 句の指定後のリカバリ」を参照してください。

プライマリ・サイトで NOLOGGING を指定した結果、変更が失われてもよい場合は、何もしないでもかまいません。

### ネットワークの問題の対処

スタンバイ・サイトが必須の場合には、スタンバイ・サイトでは何も行う必要はありません。スタンバイ・サイトがオプションの場合には、REDO ログを適用し、ただちにギャップを解決する必要があります。時間をかけずにギャップを解決することが重要です。そうしないと、プライマリ・サイトにアクセスできなくなった場合に、アーカイブ REDO ログにアクセスできなくなるというリスクが生じます。

## 進行中のリカバリがないスタンバイ・データベースのアクティブ化

スタンバイ・データベースをアクティブにする前に、適切な REDO ログを適用する必要があります。次のステップで示すように、スタンバイ・データベースをアクティブにする前に、REDO ログ順序内のギャップをすべて解決しておく必要があります。

1. 次の SQL スクリプトを実行し、ギャップ・シーケンス内のアーカイブ REDO ログを識別します。

```
SELECT high.thread#, "LowGap#", "HighGap#"
FROM
(
  SELECT thread#, MIN(sequence#)-1 "HighGap#"
  FROM
  (
    SELECT a.thread#, a.sequence#
    FROM
    (
      SELECT *
      FROM v$aarchived_log
    ) a,
    (
      SELECT thread#, MAX(sequence#) gap1
      FROM v$log_history
      GROUP BY thread#
    ) b
    WHERE a.thread# = b.thread#
    AND a.sequence# > gap1
  )
  GROUP BY thread#
) high,
```



```
(
  SELECT thread#, MIN(sequence#) "LowGap#"
  FROM
  (
    SELECT thread#, sequence#
    FROM v$log_history, v$datafile
    WHERE checkpoint_change# <= next_change#
    AND checkpoint_change# >= first_change#
  )
  GROUP BY thread#
) low
WHERE low.thread# = high.thread#;
```

THREAD#	LowGap#	HighGap#
1	90	92

ギャップ・シーケンスは LowGap# から HighGap# までです。この例では、スレッド 1 のギャップ・シーケンスは 90、91 および 92 です。このステップでギャップ・シーケンスを選択しない場合は、ステップ 5 に進んでください。

2. プライマリ・サイトからスタンバイ・サイトに転送する必要があるアーカイブ REDO ログを識別します。

V\$ARCHIVED\_LOG ビューへの問合せを次のように実行し、ギャップ・シーケンス内のログのファイル名を入手します。

```
SELECT name FROM v$archived_log
WHERE thread#=1 AND sequence#<=92 AND sequence#>=90;
```

NAME
/vobs/oracle/dbs/r_1_90.arc
/vobs/oracle/dbs/r_1_91.arc
/vobs/oracle/dbs/r_1_92.arc

3. ギャップ・シーケンス内のログを、プライマリ・データベースからスタンバイ・データベースに次のように転送します。

```
% rcp /vobs/oracle/dbs/r_1_90.arc STBYHOST:/fs2/oracle/stdby/stdby_1_90.arc
% rcp /vobs/oracle/dbs/r_1_91.arc STBYHOST:/fs2/oracle/stdby/stdby_1_91.arc
% rcp /vobs/oracle/dbs/r_1_92.arc STBYHOST:/fs2/oracle/stdby/stdby_1_92.arc
```

すべてのアーカイブ REDO ログがプライマリ・データベースと同じシステム上にあるように、プライマリ・サイトのローカル・ディレクトリを必須のアーカイブ先として指

定しておく必要があります。プライマリ・システムにアクセスできない場合、プライマリ・データベースが複数のスタンバイ・データベースを持つ構成の一部である場合には、他のスタンバイ・サイトにあるアーカイブ REDO ログを識別してみることができます。

4. スタンバイ・データベースで次の文を発行し、ギャップ・シーケンス内のログを手動で適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

ギャップがなくなるまでステップ 1 から 4 を繰り返します。

5. 次の文を発行し、スタンバイ・データベースをアクティブにします。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

6. 次のようにスタンバイ・データベース・インスタンスをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

7. 新しい本番インスタンスを起動します。

必要であれば、新しい本番インスタンスのパラメータ・ファイルを作成します。このファイルは、スタンバイ・データベースのパラメータ・ファイルを使用して作成できます。次に、スタンバイ・データベースで次の文を発行します。

```
SQL> STARTUP PFILE=FailOver.ora;
```

## シナリオ 10: 時間差を設定したスタンバイ・データベース

管理リカバリ・モードでは、スタンバイ・データベースはプライマリ・データベースから REDO ログが到着すると、ログを自動的に適用します。ただし、ログを即時に適用しないようにすることも考えられます。プライマリ・サイトでの REDO ログのアーカイブと、スタンバイ・サイトでのログの適用との間に時間差を設けた場合です。時間をあけることにより、破損したデータまたはエラーのあるデータが、プライマリ・サイトからスタンバイ・サイトに転送されることを防止できます。

たとえば、プライマリ・データベースで毎晩バッチ・ジョブを実行するとします。このバッチ・ジョブを間違えて 2 回実行し、2 回目のバッチ・ジョブが完了するまでその誤りに気が付かなかったとします。理想的には、バッチ・ジョブの開始前の時点までデータベースをロールバックする必要があります。時間差を設けた（たとえば 8 時間）スタンバイ・データベースを持つプライマリ・データベースであれば、リカバリは簡単です。時間差が設定されたスタンバイ・データベースをアクティブにし、このデータベースを新しい本番データベースとして使用します。

時間差を設定したスタンバイ・データベースを作成するには、管理リカバリ・モードではなく、手動リカバリ・モードを使用します。この場合も、管理リカバリ・モードの場合と同様に、オンライン REDO ログ・ファイルは自動的に転送できます。しかし、ログ・ファイルはスタンバイ・データベースにすぐには適用されません。定期的にスクリプトを実行し、各

REDO ログが、目的の時間差を生じさせるだけの古さのものかどうかを確認できます。このスクリプトで、十分な古さの REDO ログを手動リカバリ・ディレクトリに移動させた後、手動でログを適用します。

このシナリオの例では、4 時間の時間差を設定し、次のトピックを説明します。

- [時間差を設定したスタンバイ・データベースの作成](#)
- [時間差を設定したスタンバイ・データベースの管理](#)
- [データベースの指定時刻へのロールバック](#)
- [時間差をバイパスしたスタンバイ・データベースのアクティブ化](#)

典型的なスタンバイ・データベースの作成手順を確認してから、このシナリオを読んでください。このシナリオでは、ステップの詳細な説明を省略しています。通常のスタンバイ・データベースのセットアップの詳細は、5-14 ページの「[シナリオ 2: リモート・ホスト上のスタンバイ・データベースの作成](#)」を参照してください。

## 時間差を設定したスタンバイ・データベースの作成

次のステップを実行し、時間差を設定したスタンバイ・データベースを作成します。

1. データ・ファイルをバックアップし、プライマリ・サイトにスタンバイ・データベースの制御ファイルを作成します。
2. データ・ファイルおよびスタンバイ・データベースの制御ファイルをスタンバイ・サイトに転送します。
3. `tnsnames.ora` および `listener.ora` ネットワーク・ファイルを構成します。
4. スタンバイ・サイトでリスナーを起動します。
5. スタンバイの初期化パラメータ・ファイルを構成します。

`STANDBY_ARCHIVE_DEST` および `LOG_ARCHIVE_DEST_1` パラメータを編集します。

`LOG_ARCHIVE_DEST_1` パラメータは、アーカイブ REDO ログの場所を指定します。手動リカバリを実行するときにはこのディレクトリを使用する必要があります。管理リカバリを実行するときには、`STANDBY_ARCHIVE_DEST` パラメータで指定されたディレクトリを使用する必要があります。

たとえば、これらのパラメータを次のような値に設定したとします。

```
STANDBY_ARCHIVE_DEST=/fs2/oracle/stdby_log/  
LOG_ARCHIVE_DEST_1   ='LOCATION=/fs2/oracle/stdby/'
```

6. スタンバイ・データベースをマウントします。データベースをオープンしたり、管理リカバリ・モードにしたりしないでください。
7. プライマリの初期化パラメータ・ファイルを構成します。

8. ギャップ・シーケンス内のログを識別し、適用します。
9. 管理リカバリ・ディレクトリ内のログ・ファイルをチェックし、指定されたタイムスタンプのあるログ・ファイルを手動リカバリ・ディレクトリに移動するために定期的に行えるスクリプトを作成します。REDO ログが新しく移動された場合には、手動リカバリ・モードを開始し、新しく移動した REDO ログを適用します。

次の PERL スクリプトは、このステップで説明した作業を実行します。

```
#!/usr/local/bin/perl

#How many hours the standby database should lag behind the primary database
$LAG_HOUR = 4;

#The manual recovery directory
$DEST_DIR = '/fs2/oracle/stby/';

#The flag for whether there are new logs to be applied.
$needApply = 0;

#Check the managed recovery directory
while ( </fs2/oracle/stby_log/*.arc> ) {
    # Get the timestamp of the file
    $file_time = (stat($_))[9];
    # See if the file is "old enough"
    if ( time-$file_time > $LAG_HOUR*60*60 ) {
        print "mv $_ $DEST_DIR\n";
        system "mv $_ $DEST_DIR";
        $needApply = 1;
    }
}
#If redo logs were moved in this round, apply them
if ( $needApply == 1 ) {
    system "/usr/Lagged_Standby/ApplyLog";
}
REDO ログの適用に使用される SHELL スクリプト (/usr/Lagged_Standby/ApplyLog) は、次のように構成されています。

sqlplus internal << EOF

    recover automatic standby database;
    cancel
    exit

EOF
```

10. 1 日の特定の時間にトリガーを受け取るジョブを作成する方法については、使用しているプラットフォームのドキュメントを参照してください。

たとえば UNIX の場合には、CRON ジョブ・ファイルを作成できます。CRON ジョブ・ファイルの作成方法のヘルプが必要な場合は、UNIX コマンド・シェルで **man crontab** コマンドを発行します。

ジョブのスケジュール頻度を決定する必要があります。ジョブの実行頻度が多いほど、きめ細かい作業が実行され、一度に適用されるログの量は少なくなります。最小の単位は 1 つの REDO ログ・ファイルです。

たとえば、10 分間隔でジョブを実行する場合は、次のような CRON ジョブ・ファイルを作成できます（ステップ 9 で作成したスクリプト・ファイルが /usr/Lagged\_Standby/lag4.pl であるとして）。

```
0,10,20,30,40,50 * * * * /usr/Lagged_Standby/lag4.pl
```

前述の CRON ジョブ・ファイルが /usr/Lagged\_Standby/jobs.cron であるとして。次のコマンドを発行し、CRON ジョブをスケジュールできます。

```
% crontab /usr/Lagged_Standby/jobs.cron
```

lag4.pl スクリプトは 10 分間隔で実行されます。正確な時間差は、指定された時間の後に表示されます（この場合は 4 時間）。

## 時間差を設定したスタンバイ・データベースの管理

オンライン REDO ログの適用が遅れるため、時間差を設定したスタンバイ・データベースのランタイム・シナリオは、時間差のないスタンバイ・データベースのシナリオとは多少異なります。

時間差の設定されたスタンバイ・データベースを管理するときには、DBA としては次の作業を考慮しておく必要があります。

- プライマリ・サイトにデータ・ファイルが追加されたときには、新しいデータ・ファイルをスタンバイ・サイトにコピーする必要があります。
- スタンバイ・データベースの制御ファイルをリフレッシュする前に、スタンバイ・サイトに転送されているが、スタンバイ・データベースに適用されていないすべてのアーカイブ REDO ログを適用しておく必要があります。次のステップを実行します。

1. 次の例で示すように、すべてのアーカイブ REDO ログを、手動で手動リカバリ・ディレクトリに移動します。

```
% mv /fs2/oracle/stdby_log/*.arc /fs2/oracle/standby/
```

2. すべてのアーカイブ REDO ログを次のように適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

3. スタンバイ・データベースの制御ファイルをリフレッシュします。

この方法では、スタンバイ・データベースの制御ファイルは更新されますが、時間差はなくなります。時間差を設けるためには、指定された時間（たとえば 4 時間）の間待つ必要があります。

- プライマリ・サイトで NOLOGGING 句が指定された後のリカバリのためには、最初にはすべてのアーカイブ REDO ログを適用しておく必要があります。プライマリ・サイトで NOLOGGING を指定した結果として失われた変更があってもよい場合は、何もしないでかまいません。

1. すべてのアーカイブ REDO ログを手動で、手動リカバリ・ディレクトリに移動します。次に例を示します。

```
% mv /fs2/oracle/stdby_log/*.arc /fs2/oracle/standby/
```

2. すべてのアーカイブ REDO ログを適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

3. 影響を受けたデータ・ファイルおよびスタンバイ・データベースの制御ファイルをコピーします。

この方法では、スタンバイ・データベースの制御ファイルは更新されますが、時間差はなくなります。時間差を設けるためには、指定された時間（たとえば 4 時間）の間待つ必要があります。

- ネットワークに問題がある場合

スタンバイ・サイトが必須の場合には、スタンバイ・サイトでは何も行う必要はありません。スタンバイ・サイトがオプションの場合には、次のステップを実行し、ただちにギャップを解決できます。

1. すべてのアーカイブ REDO ログを手動で手動リカバリ・ディレクトリに移動します。次に例を示します。

```
% mv /fs2/oracle/stdby_log/*.arc /fs2/oracle/standby/
```

2. すべてのアーカイブ REDO ログを適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

3. ギャップ・シーケンス内のログを識別し、適用します。

この方法では、スタンバイ・データベースの制御ファイルは更新されますが、時間差はなくなります。時間差を設けるためには、指定された時間（たとえば 4 時間）の間待つ必要があります。

## データベースの指定時刻へのロールバック

このシナリオの最初に説明したケースでは、時間差を利用し、現行の本番データベースよりも指定された時間（たとえば 4 時間）だけ前の状態の本番データベースを得ることができました。適切な時間差のあるスタンバイ・データベースをアクティブにするには、次のようにします。

1. 次の文を発行し、スタンバイ・データベースをアクティブにします。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

2. スタンバイ・データベース・インスタンスをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

3. 新しい本番インスタンスを起動します。

```
SQL> STARTUP PFILE=Failover.ora;
```

## 時間差をバイパスしたスタンバイ・データベースのアクティブ化

時間差を利用しない場合には、次のように、時間差のない通常のスタンバイ・データベースとしてスタンバイ・データベースをアクティブにできます。

1. すべてのアーカイブ REDO ログを手動で手動リカバリ・ディレクトリに移動します。次に例を示します。

```
% mv /fs2/oracle/stdby_log/*.arc /fs2/oracle/standby/
```

2. すべてのアーカイブ REDO ログを適用します。

```
SQL> RECOVER AUTOMATIC STANDBY DATABASE;
```

3. 次の文を発行し、スタンバイ・データベースをアクティブにします。

```
SQL> ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

4. スタンバイ・データベース・インスタンスをシャットダウンします。

```
SQL> SHUTDOWN IMMEDIATE;
```

5. 新しい本番インスタンスを起動します。

```
SQL> STARTUP PFILE=Failover.ora;
```





## 数字

---

- 8.1.5 の新機能
  - スタンバイ・データベース, x
- 8.1.6 の新機能
  - スタンバイ・データベース, x

## A

---

- ALTER DATABASE 文, 1-27
  - CLEAR LOGFILE GROUP 句, 4-16
  - CLEAR UNARCHIVED LOGFILE 句, 4-13, 5-28
  - CREATE STANDBY CONTROLFILE 句, 4-15, 5-15
  - DATAFILE OFFLINE DROP 句, 4-13
  - ENABLE THREAD 句, 4-12
  - MOUNT STANDBY DATABASE 句, 3-4, 4-15
  - OPEN READ ONLY 句, 3-17, 3-18, 4-17
  - OPEN RESETLOGS 句, 5-28
  - RESETLOGS パラメータ, 3-7
- ALTER SYSTEM 文
  - ARCHIVE LOG CURRENT 句, 3-20, 5-41
  - SWITCH LOGFILE 句, 4-11
- ARCHIVE LOG CURRENT 句
  - ALTER SYSTEM 文, 3-20, 5-41

## C

---

- CLEAR LOGFILE GROUP 句
  - ALTER DATABASE 文, 4-16
- CLEAR UNARCHIVED LOGFILE 句
  - ALTER DATABASE 文, 4-13, 5-28
- COMPATIBLE 初期化パラメータ, 5-9
- CONTROL\_FILES 初期化パラメータ, 2-19, 5-2, 5-9

- CREATE CONTROLFILE 文
  - スタンバイ・データベース、影響, 4-13, 5-26
- CREATE STANDBY CONTROLFILE 句
  - ALTER DATABASE 文, 4-15

## D

---

- DATAFILE OFFLINE DROP 句
  - ALTER DATABASE 文, 4-13
- DB\_FILE\_NAME\_CONVERT 初期化パラメータ, 2-20, 2-21, 5-2, 5-9
  - プライマリ・データベースおよびスタンバイ・データベース, 2-19
- DB\_FILES 初期化パラメータ, 5-9
- DB\_NAME 初期化パラメータ, 5-9

## E

---

- ENABLE THREAD 句
  - ALTER DATABASE 文, 4-12
- ENABLE オプション
  - LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 5-8

## L

---

- LISTENER.ORA ファイル
  - 構成, 2-11, 2-13, 5-5
  - 書式, 5-5, 5-16
- LOCK\_NAME\_SPACE 初期化パラメータ, 5-2, 5-9
- LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 3-4, 5-8, 5-9
  - ENABLE オプション, 5-8
  - MANDATORY オプション, 2-17
  - OPTIONAL オプション, 2-17, 5-8
  - REOPEN オプション, 2-18, 5-8

SERVICE オプション, 5-8  
宛先の指定に使用, 5-7, 5-8  
LOG\_ARCHIVE\_DEST 初期化パラメータ, 3-4  
LOG\_ARCHIVE\_FORMAT 初期化パラメータ, 5-7  
LOG\_ARCHIVE\_START 初期化パラメータ, 5-18  
LOG\_ARCHIVE\_TRACE 初期化パラメータ, 4-6  
LOG\_FILE\_NAME\_CONVERT 初期化パラメータ,  
5-2, 5-17

## M

---

MANDATORY オプション  
LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 2-17  
MOUNT STANDBY DATABASE 句  
ALTER DATABASE 文, 3-4, 4-15

## N

---

NOLOGGING 句, 5-28, 5-47

## O

---

OPEN READ ONLY 句  
ALTER DATABASE 文, 3-17, 3-18, 4-17  
OPEN RESETLOGS 句  
ALTER DATABASE 文, 5-28  
OPTIONAL オプション  
LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 2-17,  
5-8  
Oracle Parallel Server (OPS)  
スタンバイ・データベース, 3-22

## R

---

RECOVER 文  
FROM 'location' 句, 4-15  
TIMEOUT オプションの指定, 5-14  
REDO ログ  
削除, 4-12, 5-25  
消去, 4-13, 4-16  
追加, 4-12, 5-25  
リセット, 4-13  
REOPEN オプション  
LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 2-18,  
5-8  
RESETLOGS パラメータ  
ALTER DATABASE 文, 3-7

## S

---

SERVICE オプション  
LOG\_ARCHIVE\_DEST\_*n* 初期化パラメータ, 5-8  
SQL 文  
ALTER DATABASE, 1-27, 4-13, 4-16  
ALTER SYSTEM, 4-11  
CREATE CONTROLFILE, 4-13  
スタンバイ・データベース関連, 1-27  
STANDBY\_ARCHIVE\_DEST 初期化パラメータ, 2-21,  
5-9  
SWITCH LOGFILE 句  
ALTER SYSTEM 文, 4-11

## T

---

TAF  
「透過的アプリケーション・フェイルオーバー」を  
参照  
TNSNAMES.ORA ファイル, 2-17  
構成, 2-11, 2-12, 5-4  
書式, 5-5, 5-16

## U

---

UNRECOVERABLE 操作, 5-28  
後のバックアップ, 4-14  
USER\_DUMP\_DEST 初期化パラメータ, 4-6

## V

---

V\$ARCHIVE\_DEST ビュー, 2-33, 5-40  
V\$ARCHIVED\_LOG ビュー, 2-24, 4-5, 5-21, 5-42  
V\$DATABASE ビュー, 2-9  
V\$DATAFILE ビュー, 4-14, 5-14, 5-24  
V\$INSTANCE ビュー, 3-15  
V\$LOG\_HISTORY ビュー, 1-25, 3-15, 4-9, 5-12,  
5-20

## あ

---

アーカイブ  
宛先の状態, 2-17  
失敗した宛先, 2-18  
アーカイブ REDO ログ  
宛先  
失敗の再アーカイブ, 2-18

宛先の状態, 2-17  
状態のチェック, 1-25  
情報のアクセス, 4-5, 4-9  
スタンバイ・データベース, 4-5  
多重化, 2-16, 5-7  
転送, 3-13

アーカイブ REDO ログの宛先の状態, 2-17  
アーカイブ REDO ログの転送, 3-13  
アーカイブ先, 2-17, 5-31  
アーカイブ・トレース  
    スタンバイ・データベース, 4-6  
アーカイブ先  
    MANDATORY, 5-31  
    OPTIONAL, 5-32  
アクティブ化  
    スタンバイ・データベース, 1-11, 3-19  
    結果, 1-12

## お

---

オフライン  
    スタンバイ・データベースのデータ・ファイル,  
        4-13  
オンライン・ログの消去  
    スタンバイ・データベースの含意, 4-16

## か

---

改名  
    データ・ファイル, 4-12, 5-24  
        自動, 2-21  
        手動, 2-31  
管理リカバリ  
    スタンバイ・データベース, 2-5, 3-13  
        開始, 3-13  
        準備, 2-5  
        取消し, 3-15  
管理リカバリ・モード, 1-20, 3-13

## き

---

ギャップ・シーケンス  
    原因, 3-5  
    スタンバイ・データベースへのログの適用, 3-12,  
        5-13, 5-20  
    説明, 1-16  
    ～内のログのコピー, 3-10, 5-12

～内のログの識別, 3-8, 5-11, 5-41  
防止, 3-7

## く

---

クライアント・アプリケーション・フェイルオーバー  
構成, 5-36

## こ

---

構成  
    listener.ora ファイル, 5-5  
    tnsnames.ora ファイル, 5-4  
    クライアント・アプリケーション・フェイルオー  
        バー, 5-36  
    初期化パラメータ・ファイル, 5-6, 5-8, 5-17,  
        5-19  
    スタンバイ環境, 1-18  
    ネットワーク・ファイル, 2-27, 5-4  
構成オプション  
    スタンバイ・データベース, 1-2  
コマンド  
    スタンバイ・データベース, 1-27

## さ

---

削除  
    REDO ログ, 5-25  
    データ・ファイル, 5-25  
    表領域, 5-25  
作成  
    スタンバイ・データベース, 2-7, 2-12, 2-13,  
        2-14, 2-32  
        概要, 1-15  
    制御ファイル, 5-15  
    データ・ファイル, 5-22

## し

---

時間差  
    スタンバイ・データベース, 5-51 ～ 5-55  
手動リカバリ  
    スタンバイ・データベース, 1-16, 2-6  
        準備, 2-6  
手動リカバリ・モード, 1-7  
準備  
    スタンバイ・データベース, 2-2

## 初期化パラメータ

- COMPATIBLE, 5-9
- CONTROL\_FILES, 2-19, 5-2, 5-9
  - プライマリ・データベースおよびスタンバイ・データベース, 2-19
- DB\_FILE\_NAME\_CONVERT, 2-20, 2-21, 5-2, 5-9
  - プライマリ・データベースおよびスタンバイ・データベース, 2-19
- DB\_FILES, 5-9
- DB\_NAME, 5-9
- LOCK\_NAME\_SPACE, 5-2, 5-9
- LOG\_ARCHIVE\_DEST, 3-4
- LOG\_ARCHIVE\_DEST\_*n*, 3-4, 5-7
- LOG\_ARCHIVE\_DEST\_STATE\_*n*, 2-17
- LOG\_ARCHIVE\_DEST\_*n*, 5-9
- LOG\_ARCHIVE\_FORMAT, 5-7
- LOG\_ARCHIVE\_START, 5-18
- LOG\_ARCHIVE\_TRACE, 4-6
- LOG\_FILE\_NAME\_CONVERT, 5-2, 5-17
- STANDBY\_ARCHIVE\_DEST, 2-21, 5-9
- USER\_DUMP\_DEST, 4-6
- プライマリ・データベースおよびスタンバイ・データベース, 2-19

## 初期化パラメータ・ファイル

- 構成, 2-27, 5-6, 5-8, 5-9, 5-17, 5-19

## す

---

### スタンバイ・データベース

- 8.1.5 の新機能, x
- 8.1.6 の新機能, x
- REDO ログ・ファイル
  - 変更, 4-12
- アーカイブ REDO ログ
  - 転送, 3-13
- アーカイブ REDO ログの転送, 3-13
- 影響を与える変更への応答, 4-10
- 環境の構成, 1-18, 2-2
- 管理リカバリ・モード, 1-8, 3-13
- 起動, 1-11, 1-18, 3-19
- 構成オプション, 1-2
- コマンド, 1-27
- 再作成, 5-43
- 作成
  - 手順, 2-7, 2-12, 2-13, 2-14, 2-32
  - 同一ホスト上, 5-2

- リモート・ホスト上, 5-14
- 作成の準備, 2-2
- 時間差
  - 管理, 5-53
  - 作成, 5-51
  - データベースのロールバック, 5-55
  - バイパス, 5-55
- 手動リカバリ・モード, 1-7
  - 手順, 2-31, 3-4
- 初期化パラメータ, 2-19, 5-9
- 制御ファイル
  - 作成, 2-8, 5-15
  - 変更, 5-26
  - リフレッシュ, 4-15, 5-26
- 制御ファイルの変更, 4-13, 5-26
- 説明, 1-2
- ダイレクト・パス処理, 4-13
- データ・ファイル
  - オフライン, 4-13
  - 改名, 4-12
  - 作成, 4-10
- テスト, 1-14
- バックアップ, 4-16
- フェイルオーバー, 1-11, 3-19
- プライマリ・データベースの変更の影響, 1-26
- 無効, 3-7
- モニター, 4-2
- 読取り専用モード, 1-9, 3-16
- リカバリが継続されない, 5-46
- リカバリ・モードの維持, 1-8
- 利点, 1-2

## スタンバイ・データベースのモニター, 4-2

## せ

---

### 制御ファイル

- CONTROL\_FILES 初期化パラメータ
  - プライマリ・データベースおよびスタンバイ・データベース, 2-19
- 作成, 2-8, 5-15
- スタンバイ・データベースのリフレッシュ, 4-15, 5-26
- スタンバイ・データベースへの影響, 4-13
- 変更, 4-13, 5-26

## た

---

ダイレクト・パス処理  
    スタンバイ・データベース, 4-13  
多重化  
    アーカイブ REDO ログ, 2-16, 5-7

## つ

---

追加  
    REDO ログ, 4-12, 5-25  
    データ・ファイル, 4-10, 5-22, 5-47

## て

---

データベース  
    スタンバイ  
        REDO ログ・ファイル、変更, 4-12  
        アーカイブ REDO ログの転送, 3-13  
        作成手順, 2-7, 2-12, 2-13, 2-14, 2-32  
        手動リカバリ・モード, 1-7, 2-31, 3-4  
        初期化パラメータ, 2-19  
        制御ファイル, 4-13, 4-15  
        ダイレクト・パス処理, 4-13  
        データ・ファイル、オフラインにする方法, 4-13  
        データ・ファイルの改名, 4-12  
        フェイルオーバー, 1-11, 1-12, 3-19  
        読取り専用モード, 1-9, 3-16  
        リカバリ・モードの維持, 1-8  
        利点, 1-2  
    物理構造の変更  
        スタンバイ・データベース、影響, 1-26  
データ・ファイル  
    改名, 5-24  
        スタンバイ・データベースへの影響, 4-12  
    削除, 5-25  
    作成, 5-22  
    プライマリ・データベースへの追加, 4-10, 5-22, 5-47  
テスト  
    スタンバイ・データベース, 1-14

## と

---

透過的アプリケーション・フェイルオーバー, 5-37  
トレース・ファイル  
    場所, 4-6

## ね

---

ネットワーク  
    障害, 5-39  
ネットワーク・ファイル  
    構成, 2-27, 5-4

## は

---

バックアップ  
    UNRECOVERABLE 操作後, 4-14  
    スタンバイ・データベース, 4-16  
    スタンバイ・データベースの作成元, 1-19  
    スタンバイ・データベースの表領域, 4-17  
    プライマリ・データベース, 1-19  
    累積増分, 3-6, 3-13, 3-23

## ひ

---

ビュー  
    V\$ARCHIVE\_DEST, 2-33, 5-40  
    V\$ARCHIVED\_LOG, 2-24, 4-5, 5-21, 5-42  
    V\$DATABASE, 2-9  
    V\$DATAFILE, 4-14, 5-14, 5-24  
    V\$INSTANCE, 3-15  
    V\$LOG\_HISTORY, 1-25, 3-15, 4-9, 5-12, 5-20  
表領域  
    削除, 5-25  
    スタンバイ・データベースでのバックアップ  
        シャットダウン中, 4-17  
        読取り専用モード, 4-17  
    プライマリ・データベースへの追加  
        スタンバイへの影響, 4-10

## ふ

---

フェイルオーバー  
    スタンバイ・データベース, 1-11, 1-18, 2-4, 3-19  
    結果, 1-12  
    接続方法, 2-4

## へ

---

変更  
    制御ファイル, 4-13, 5-26

## め

---

メンテナンス  
スタンバイ・データベース  
概要, 1-25

## よ

---

読取り専用モード  
スタンバイ・データベース, 1-9, 3-16

## ら

---

ライフ・サイクル  
スタンバイ・データベース, 1-15

## り

---

リカバリ  
ネットワーク障害, 5-39  
リカバリ・モードの維持, 1-8  
リフレッシュ  
スタンバイ・データベースの制御ファイル, 4-15,  
5-26