

Oracle: 組み込み環境における Oracle データベースの開発

Mughees A. Minhas, Oracle Corporation

Anuj Goe, Oracle Corporation

概要

業務アプリケーションが様々なビジネスで使用されていますが、それぞれのアプリケーションを実行するデータベース・ソフトウェアを直接管理するのに必要な技術的ノウハウは不足しています。これらのアプリケーションは、業務を成功させるのに不可欠なものです。特に中小型の企業には、IT や DBA スタッフがほとんどしなかったり、まったくいない場合もあり、一般的にこうしたノウハウを習得するためのリソースが不足しています。このため、中小型の企業には、インストールが簡単な自己管理型の、DBA 以外の人でも簡単に使用できる業務アプリケーションが必要です。これらの企業は、よく、組み込みデータベースが実装されている統合ターンキー・ソリューションを購入します。

組み込みインストールによって、ISV は、クリティカル・テクノロジーを直接自社製品に組み込み、この業界のニーズに的を絞ったソリューションを開発することができます。これらのソリューションには、データベース・ソフトウェアは業務アプリケーションやデバイスに完全統合されているため、エンド・ユーザーはデータベース・ソフトウェアが存在することをほとんど知りません。実際に、データベースは、アプリケーション・ユーザーには、全く見えません。これにより、ISV は、インストールから管理、サポートまで、さらに大きな完全なアプリケーションのコントロールが可能になります。

Oracle 10g によって、まさにオラクル社は、組み込み可能なデータベースの作成に向けて飛躍的に進歩しました。この文書では、データベースの組み込みに関連した技術的問題と、これらの問題が Oracle 10g によってどのように解決されるか説明します。また、この文書では、Oracle 10g を一般的なアプリケーションに組み込む詳細な手順も説明しています。

組み込みによって得られる利点

組み込みデータベースは、業務アプリケーションの開発を行う ISV、およびエンド・ユーザーまたは顧客の両方にとって多くの利点があります。

ISV に対する利点

ISV がデータベースを自社のアプリケーションに組み込むことで、次の利点が得られます。

- データベースが自社のアプリケーションまたはデバイスに不可欠なコンポーネントとなるため、顧客のデータベース環境を完全にコントロールし理解できます。
- 自社のアプリケーションに最適なデータベース構成をテストした後、その構成を使用して顧客のデータベースを構成できます。
- 顧客のデータベースを管理、更新できます。組み込みデータベースは、ISV のアプリケーションによってインストール、構成および管理されるため、インストールと構成は簡単かつ高速で行われ、またエラーがおこりにくいいためサポート・コストが著しく減少します。
- ソフトウェアを割引価格で購入できます。ISV は割引価格で購入できるため、その分を中小型企业のニーズに対応するための費用にあてられます。

顧客に対する利点

ISV に対する利点のほとんどが、そのまま顧客にとっての利点となります。組み込みソリューションを使用して、顧客は次のことを実現できます。

- 理論的には 1 ベンダー、1 製品を使用することで、サポート上の問題を削減できます。
- DBA あるいはデータベースを管理する専門家でなければならない機能をなくし、IT コストを削減できます。
- アプリケーション・インストール時間を短縮できます。データベース・ソフトウェアとアプリケーション・ソフトウェアを別々にインストールする必要はありません。アプリケーション・ソフトウェアをたった 1 回インストールするだけです。
- 自社のアプリケーションに適したデータベース環境を構成できるため、高いパフォーマンスが得られます。

これらは、ISV とエンド・ユーザーが、自分のアプリケーションに Oracle Database 10g を組み込んだときに利用できる利点のほんの一例です。

組み込みデータベースの問題

データベースのアプリケーションへの組み込みには、重要な技術的問題がいくつかあります。これらの問題は、次のカテゴリに分類できます。

1 配置

組み込みデータベースのインストールと構成は、ユーザーからはまったくみえず、ユーザーのアプリケーションと完全に統合されている必要があります。データベースは、アプリケーション・パッケージのコンポーネントとしてパッケージ化し、アプリケーション・インストールの一部としてインストールする必要があります。

2 管理

組み込みデータベースは、自己管理型でなければなりません。すべてのルーチン管理業務を自動化する必要があります。さらに、データベースは、システム環境の変更に適合し、外部の介入を要求することなく自動的に通常の問題を解決できる必要があります。万が一エラーあるいは問題が発生した場合には、データベースから、問題を診断して実行可能な対策を薦める簡単な方法が提供されなければなりません。

3 ソフトウェア・メンテナンスとサポート

ソフトウェア・メンテナンスは、すべてのソフトウェア・アプリケーションが取り組まなければならない問題です。組み込みデータベースでは、更新やソフトウェアへのパッチの適用などのタスクが非常に簡単に実施でき、できるかぎり自動化されていることが不可欠です。

では、Oracle データベースを業務アプリケーションに組み込める各種の Oracle Database 10g ソリューションについて説明します。

組込み配置

Oracle 10g インストールの強化点

Oracle 10g のインストールと構成は非常に簡単になりました。インストールは、より高速に、より軽量化されました。たった1枚のCDからインストールできるようになり、このCDには、事前に作成されたシード・データベースと一緒に Personal, Standard, Enterprise Editions 用のバイナリが含まれています。Oracle のインストールが強化され、ターゲット・システムに OS、パッチ、メモリー、CPU などが搭載されていることを確認する前提条件チェックが可能となりました。

Oracle 10g インストールは、ソフトウェアのインストールとアンインストールを本格的なサイレント・モードで実行できます。サイレント・インストールと合わせて、Oracle 10g はサイレント・モード・アンインストールもサポートしています。どのプログラムでも Oracle インストーラをサイレント・モードで呼び出して、アンインストールする応答ファイルからコンポーネント・リストを提供できます。Oracle 10g には、サイレント・インストール用応答ファイルを作成するレコード・アンド・プレイ機能があります。このモードは、Oracle インストーラと Database Creation Assistant (DBCA) の問題に対するユーザー応答を収集し、収集された応答はその後サイレント・モード・インストール用の入力として使用できます。Oracle Universal Installer は、応答ファイルの格納値を上書きすることによって、コマンド・ライン引数を取ってインストール時間の値を受け入れることができます。現在はインストール時のアプリケーション・インストール・プロセスで、どの Oracle コンポーネントを、どこに、どのような構成でインストールするか簡単にコントロールできるため、この機能は非常に重要です。

この他に Oracle 10g の重要な強化点は、Oracle クライアント (ドライバのみ) を Oracle Home にインストールする必要がなくなったことです。アプリケーションに、Oracle ドライバをパッケージ化し任意の場所にコピーしてから、Unix または Windows プラットフォームのレジストリ・エントリ、あるいはその両方で、必要な環境変数を設定できます。

Oracle 10g インストールで、Oracle ソフトウェア、リスナーおよびデータベースすべてを1つの統合されたプロセスに構成することもできます。事前に作成されたシード・データベースを自分のアプリケーション固有のデータベースに置き換えられます。この機能によって、アプリケーション・インストールの時間を短縮できます。ISV は、データベースを作成、構成し、自分のラボのアプリケーション固有のメタ・データを使用してロードした後、アプリケーションの統合コンポーネントとしてパッケージ化し、メディアにのせて出荷できます。エンド・ユーザーが選択したデータファイルの記憶場所など、さらにカスタマイズされたデータベースを作成する場合に備えて、Oracle データベースは、コマンド・ライン引数に基づいてデータベースを作成、構成できるフレキシビリティを備えています。

組込み方法

Oracle Database の組込みは、2つのフェーズに区分できます。第1のフェーズはパッケージ化と呼ばれます。このフェーズで、ISV は顧客に引き渡すインストール用の統合ソリューションを作成します。第2のフェーズでは、顧客が、あなたのパッケージを使用して、組込みデータベースを搭載したアプリケーションを顧客自身のマシンにインストールします。このフェーズはインストールと呼ばれます。各顧客の環境はそれぞれ違うため、インストールには顧客独自の問題が存在します。

パッケージ化

次の5つのコンポーネントは、パッケージ化する必要があります。

- 1 アプリケーション・ソフトウェア
- 2 Oracle Universal Installer (OUI)
- 3 Oracle ソフトウェア/バイナリ
- 4 事前に作成されたシード・データベース (データファイルあり、またはなし)
- 5 インストール・スクリプト (任意)

多様な引渡し機構によって、どのコンポーネントでも提供できます。CDなどの配布媒体でパッケージ全体を出荷するか、または自分のウェブサイトにホスティングしてダウンロードできるようにするか、あるいはその両方を利用することができます。

1 アプリケーション・ソフトウェア

Install Shield、Microsoft Installer または自分のアプリケーション・ソフトウェアをパッケージ化するためのスクリプトなど、どんなインストール・アプリケーション・ソフトウェアでも使用できます。Oracle Universal Installer (OUI) は、Oracle のインストール・エンジンです。このエンジンは非プラットフォーム依存型で、サポートされたプラットフォーム上で同じ動作を行います。OUI には、サイレント・モードで Oracle インストールを行うための OS レベル・コマンド・ライン・インタフェースが搭載されています。したがって、OS コールアウト機能を使用して、OUI を簡単に、アプリケーション・インストール・パッケージまたは直接スクリプトから呼び出せます。

2 Oracle Universal Installer (OUI)

OUI は、Oracle 標準メディアでパッケージ化されています。ほとんどの場合、別個にパッケージ化する必要はありません。しかし、Web ベースのインストールの場合、顧客のマシンでは、OUI をローカル・マシンから実行して、Oracle バイナリを直接 Web サーバーからダウンロードしてインストールする必要があります。OUI エンジンは、Oracle Technology Network (OTN) Web サイト (<http://otn.oracle.com>) からダウンロードできます。

3 Oracle ソフトウェア/バイナリ

Oracle 10g ソフトウェアは、OTN (<http://otn.oracle.com>) からダウンロードできます。ソフトウェア CD の場合は、Oracle Store (<http://otn.oracle.com>) から注文できます。Oracle パートナの場合、Oracle Partner Program (OPP) Membership の会員として、Oracle ソフトウェアは通常出荷されます。

Oracle ソフトウェアをパッケージ化する最初の手順として、応答ファイルを作成します。このファイルで、サイレント・モードでは監視されない Oracle ソフトウェアをインストールできます。レコード・モードで OUI を使用して、まず顧客の一般的インストールをシミュレートする必要があります。次のコマンドを使用して、レコード・モードで OUI を実行します。

```
setup.exe -record -destinationFile <filename>      (Windows プラットフォームの場合)
runInstaller -record -destinationFile <filename>   (Unix および Linux プラットフォームの場合)
```

「サマリー」ページまでの応答を記録し続け、「キャンセル」ボタンをクリックしてレコード・モードを終了します。

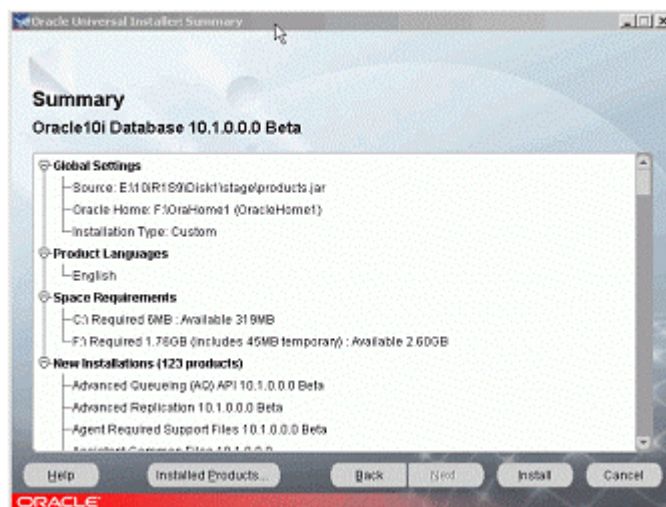


図 1: OUI のサマリー・ページ

ここで、記録された応答を顧客応答としてリプレイするために、最終パッケージに含める必要のある応答ファイル(保存先ファイルともいう)が作成されます。これで、顧客は Oracle インストーラの質問に答える必要がなくなり、すべての顧客インストールに同じ Oracle ソフトウェア・コンポーネントが入ります。

注: 前述の応答ファイルの作成前に、次項「事前に作成されたデータベース(データファイルあり、またはなし)」をよく読んで、アプリケーション要件を正しく理解し、データベースがインストール時にどのように作成されるか理解してください。

4 事前に作成されたシード・データベース(データファイルあり、またはなし)

次の2つのオプションがあります。

- i. Oracle 事前構成シード(汎用)データベースを使用します。このデータベースは自分のアプリケーションに使用できます。これは、Oracle データベースをパッケージ化しインストールする最も簡単な方法です。この方法では、前述の項「オラクル・ソフトウェア」に自分の応答ファイルを記録しながら、Oracle シード(汎用)データベースを作成するオプションを選択するだけです。
- ii. 自分のカスタム・データベースを作成し、自分のアプリケーション要件に合わせて構成してから、顧客のマシンにインストールするデータベースをパッケージ化します。このオプションには、事前に作成されたデータベースをパッケージ化してインストールするための、さらに2つのオプションが用意されています。
 - a. 自分のカスタム・データベースに Oracle シード(汎用)データベースを置き換える。
このデータベースを使用し、Oracle が提供するデータベースを自分のカスタム・データベースに置き換え、1回だけコールして Oracle ソフトウェアと一緒にインストールします。オプション(i)と同様、この方法でも、前述の項「オラクル・ソフトウェア」に自分の応答ファイルを記録しながら、Oracle シード(汎用)データベース作成のオプションを選択するだけです。

b. カスタマイズ・インストールにデータベースをパッケージ化。

ISV は、インストール時にデータベースのカスタマイゼーションを行うことができます。たとえば、エンド・ユーザーからのデータファイルに違う記憶場所を入力し、これらの入力値を使用してデータベースの構成、レイアウトを行います。この場合、OUI で Oracle ソフトウェアのインストールのみが行われ、データベースの作成は必要なくなります。データベースの作成は、自分のアプリケーション・インストール・ソフトウェアから Oracle Database Configuration Assistant (DBCA) への別のコールによって行われます。

(iia)、(iib)いずれのオプションでも、まず、既存のデータベースをパッケージ化する必要があります。これは、Oracle Database Configuration Assistant (DBCA) を使用して行われます。DBCA は、Oracle データベースの作成、構成およびパッケージ化を行う Oracle の GUI ツールです。パッケージ化の場合、DBCA ではテンプレートの概念が使用されます。DBCA ツールの使用で、データベースはどのようなマシンに対してでも転送用 DBCA テンプレートにパッケージ化することができます。ターゲット・マシン上に、DBCA によってテンプレートに基づく新しいデータベースを作成できます。DBCA テンプレートは、2 種類のファイルで構成されています。1 つはテンプレート定義ファイルで、もう 1 つはオプションで複数のデータベース・ファイルが 1 つのファイルに圧縮されています。既存のデータベースは、まず、DBCA テンプレートにパッケージ化する必要があります。DBCA でデータベースをパッケージ化する手順については、付録 A に説明します。

データファイルのないデータベース・テンプレートを作成した場合、ISV は、オプション(iib)のみを使用でき(iia)は使用できません。オプション(iib)を実装するには、データベースをカスタム・インストール用にパッケージ化する必要があります。このオプションを実装する詳細手順については、付録 B に説明します。しかし、データファイルがテンプレートに含まれている場合は、次の手順として、Oracle に実装されているシード・データベースをカスタム・パッケージ化されたシード・データベース・テンプレートに置き換えてください。Oracle に実装されているシード・データベースを置き換える詳細手順については、付録 C に説明します。

5 インストール・スクリプト (任意)

プリインストールまたはポストインストール・タスクが必要となる場合の任意のスクリプトを含めることができます。たとえば、必須の UNIX グループ、ユーザー、カーネル・パラメータ設定などを作成するには、UNIX および Linux プラットフォームにプリインストール・スクリプトを作成する必要があります。

これで、組込み配置の第 1 部「パッケージ化」の説明を終わります。では、第 2 部「インストール手順」について説明します。

インストール

組込みプロセスの第 2 部では、顧客がアプリケーション・パッケージから組込みデータベースをインストールする手順について説明します。インストール手順は、パッケージ化されている機能とパッケージ化でどのオプションを選択したかによって異なります。インストールでは、パッケージ化された同じ 5 つのコンポーネントが処理されます。これらのコンポーネントは次のとおりです。

- 1 アプリケーション・ソフトウェア
- 2 Oracle Universal Installer (OUI)
- 3 Oracle ソフトウェア/バイナリ
- 4 事前に作成されたシード・データベース (データファイルあり、またはなし)
- 5 インストール・スクリプト

1 アプリケーション・ソフトウェア

アプリケーション・ソフトウェアのパッケージ化に使用されるインストール・ソフトウェアによって、インストール・プロセスは異なります。しかし、すべてのインストール・ソフトウェアには OS にコールさせるコールアウト機能が搭載されており、システムはコールの完了を待ちます。このため、自分のアプリケーション・インストール・ソフトウェアのコールアウト機能を使用して、OUI にコールを行って Oracle ソフトウェア・インストールを開始してください。また、ユーザー入力 (ORACLE_HOME など) を集めて、必ず自分のインストール・ソフトウェア変数に入れる必要があります。これで、これらの入力値を OUI に渡すことができます。

2 Oracle Universal Installer (OUI)

OUI は、Oracle パッケージを読み込むランタイム・エンジンで、このインストーラによって Oracle パッケージがターゲット・マシンにインストールされます。OUI をインストールする必要はありません。自分のアプリケーション・インストーラを使用して、通常メディア自体からこのプログラムにコールできるようにしさえすればよいのです。

3 Oracle ソフトウェア/バイナリ

組込みインストール用 Oracle ソフトウェアをインストールするには、サイレント・モードで OUI を呼び出し、引数として自分の応答ファイルに引き渡す必要があります。他の引数もすべて引き渡して、応答ファイル内でハードコードされた値を上書きすることもできます。この機能は、ORACLE_HOME、FROM_LOCATION などインストール固有の変数の値を提供するのに使用します。FROM_LOCATION は、ファイル・システムまたは HTTP を使用した Web サーバーなど、Oracle ソフトウェア・パッケージにアクセスできる記憶場所です。

```
setup.exe -nowait -silent -responseFile <filename> ORACLE_HOME="F:\OraHome10g"
ORACLE_HOME_NAME="OracleHome10g"    (Windows プラットフォームの場合)

setup.exe -nowait -silent -responseFile <filename> ORACLE_HOME="F:\OraHome10g"
ORACLE_HOME_NAME="OracleHome10g"    (Unix および Linux プラットフォームの場合)
```

事前に作成されたシード・データベースのインストールを選択した場合、このコールで、すべての Oracle バイナリのインストールと Oracle データベースが作成されます。

4 事前に作成されたシード・データベース (データファイルあり、またはなし)

この手順は、データベースのカスタム・インストールを選択した場合のみ必要です。自分のデータベースをインストールするには、まず、自分のテンプレート・ファイルをターゲット・マシンの ORACLE_HOME/assistants/dbca/template ディレクトリにコピーしてください。次に、別のコールを行って、次のコマンドで dbca を開始します。

```
dbca -silent -createDatabase -templateName "<templatename>" -gdbname
"<dbname>" -sid "<sidname>" -sysPassword <password for sys> -systemPassword
<password for system>
```

例:

```
dbca -silent -createDatabase -templateName "embedded_db.dbc" -gdbname
"my10db1.us.oracle.com" -sid my10db1" -sysPassword sysoracle -systemPassword
systemoracle
```

これで、組込みデータベースの最初の最も重要な問題をどう解決するか、つまり、Oracle データベースをアプリケーション・インストールと完全統合して静かに配置する機能についての説明を終わります。

自己管理データベースの設計

データベースの第2の問題として、データベースは、毎日の操作でオンサイト・メンテナンスのいらぬ自己管理型である必要があります。これは、一般に、組込み環境には、データベース・システムを積極的に管理するオンサイト DBA または技術的ノウハウがないからです。ほとんどの場合、ユーザーは、使用しているアプリケーションについて多少の技術的知識はあっても、アプリケーションを実行しているデータベース・システムについての知識はまったくありません。こうした環境では、いったん配置されたデータベースは、すべてのルーチン管理業務が完全自動化された自己管理型である必要があります。第2に、外部の介入をいっさい必要とせずに、環境の変化に適応できる必要もあります。第3に、問題が発生した場合にも外部の介入をいっさい必要とせずに、データベース自体で問題を修正できる必要があります。

ルーチン管理業務の自動化

1. バックアップとリカバリの管理

データ損失をリカバリするために通常バックアップを実施することは、最も重要な管理業務といえます。バックアップの自動化は、組込みデータベースにとって不可欠です。オラクル社は、次の機能を使用してバックアップを作成することをお勧めします。

- Recovery Manager (RMAN) を使用して、バックアップおよびリカバリ操作を行います。RMAN は、データベースのバックアップの管理、さらに重要なのはリカバリを管理する Oracle のユーティリティです。RMAN の操作は簡単で、データベースの優れたパフォーマンスと可用性が利用できます。RMAN によって、要求されたバックアップ、リストアまたはリカバリ操作を最も効率的に実行できる方法が決定され、これらの操作は Oracle データベース・サーバーと共に実行されます。RMAN に、増分バックアップ、バックアップ処理を行いながら破損ブロックの検出、ブロック・メディア・リカバリ、および Oracle データベースのバックアップとリカバリ管理を簡単にするその他の機能が実装されています。
- 新しいフラッシュ・リカバリ領域機能を使用したディスクへのバックアップを実施します。フラッシュ・リカバリ領域は、Oracle データベース内にある、あらゆるリカバリ関連ファイルとアクティビティが一体化された格納場所です。1つの初期化パラメータを定義することで、すべての RMAN バックアップ、アーカイブ・ログ、コントロール・ファイル・オートバックアップおよびデータファイル・コピーが自動的に指定のディスク記憶場所に書き込まれます。フラッシュ・リカバリ領域は完全自己管理型です。Oracle サーバーは、自動的に古いバックアップとアーカイブ・ログ・ファイルを取り除き、現在のバックアップとアーカイブ・ログ・ファイルのみを維持します。ユーザーは、もう、古いバックアップを削除してフラッシュ・リカバリ領域サイズを小さく維持しようとする必要がなく、リカバリに必要なアーカイブ・ログを追跡する必要もなくなりました。これにより、リカバリ・アクティビティに必要なすべてのファイルが1つの場所に配置され、その場所がデータベースにはわかっているため、リカバリ・プロセスの時間が短縮されます。テープ・アクセスに比べてディスク・アクセスは非常に早く効率的に行われるため、フラッシュ・リカバリ領域のバックアップとリカバリ・アクティビティは極めて高速に行われます。現在のディスク上では、データがディスクのどこにあっても 2,3 ミリ秒でアクセスできます。

フラッシュ・リカバリ領域は、次の2つの初期化パラメータを設定して定義します。

DB_RECOVERY_FILE_DEST: リカバリ領域の記憶場所を設定します。

DB_RECOVERY_FILE_DEST_SIZE: リカバリ領域のバイトでサイズを設定します。

これらの2つのパラメータは動的で、データベースがオンラインの状態、ALTER SYSTEM SET コマンドを使用して変更または無効にできます。

- いったん完全データベース・バックアップを行ってから、増分夜間バックアップを行います。Oracle 10g の増分バックアップは、さらに効率的です。増分バックアップは変更されたブロックをバックアップするだけですが、Oracle9i と異なり、データファイルをスキャンして変更されたブロックを特定する必要がありません。Oracle 10g は、特別の変更追跡ファイル（1 TB データベースの場合、このファイルのサイズは約 4 MB）内の各データファイルの変更されたブロックを追跡して、増分バックアップ実行時に、データベースによって、自動的に変更追跡ファイルが使用されデータファイル・スキャンのオーバーヘッドなしに、変更されたブロックのみがバックアップされるようにします。これで、増分バックアップの時間が短縮され領域の効率が上がります。デフォルトの場合、Oracle で、変更されたブロックは追跡されません。このため、この機能を有効にして、増分バックアップを強化する必要があります。変更ブロックの追跡を有効にするコマンドは、次のとおりです。

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

これ以外に Oracle 10g で新たに強化されたのは、増分バックアップをデータベースのイメージ・コピーにマージする機能です。この強力な機能がイメージ・バックアップ・コピーを進めます。このため、メディア・リカバリに要する時間を短縮でき、さらに、マージされた増分バックアップも削除できるため領域の保存も可能です。これは、データベースが寿命となるまでに完全データベース・バックアップを 1 回だけ行えばよく、フラッシュ・リカバリ領域は安定し、実際のデータベース・サイズ以上にはならないという意味です。毎週増分バックアップを行って、データベース・バックアップのイメージ・コピーとマージする必要があります。増分バックアップを進めてイメージ・コピーにマージするコマンドは、次のとおりです。

```
RECOVER COPY OF DATAFILE <datafile# or name>
```

Oracle 10g から、Database Configuration Assistant(DBCA)を使用してデータベースを作成するときに、自動的にバックアップ・ジョブが作成されフラッシュ・リカバリ領域に夜間増分バックアップされるようになっています。デフォルトでは、このジョブは Enterprise Manager (EM) スケジューラ・ジョブとして作成されます。ユーザーは、この事前に定義されたジョブを使用するか、手作業でサーバー・ベースの Unified Scheduler を使用して同じジョブを定義することができます。Unified Scheduler の場合、ユーザーは、後述で推奨する設定手順によってバックアップ・スクリプトを作成する必要があります。

2. 領域管理

バックアップ管理はルーチン管理機能の中で最も重要であるのに対して、領域管理は最も一般的といえます。DBA がない環境の場合、業務関連の領域管理をまったく排除できないのであればせめて最小化できるよう、データベースを構成する必要があります。Oracle8i からの機能強化により、オラクル社では、着実に領域管理機能を自動化し、ついに Oracle 10g でこのソリューションが完了しました。組込みデータベースで使用する 5 つの重要な領域管理ソリューションは、次のとおりです。

1 Oracle 管理ファイル (OMF)

OMF は、Oracle9i に導入されました。OMF によって、データベース・ファイルを手作業で管理する必要がなくなりました。この機能では、デフォルトの記憶場所が、データファイル、コントロール・ファイルおよびオンライン・ログ・ファイル用に指定されています。新しいファイルを追加または作成するとき、ユーザーは記憶場所やファイルのサイズを指定する必要がありません。Oracle では、自動的にファイル固有の名前が生成され、ファイルが作成されます。また、対象のオブジェクトがデータベースから削除されると、これらのファイルは OS から削除されます。この機能を有効にする場合は、次の初期化パラメータを指定します。

DB_CREATE_FILE_DEST : データファイルのデフォルトの記憶場所を指定します。
 DB_CREATE_ONLINE_LOG_DEST_<n> : オンライン・ログとコントロール・ファイルのコピーのデフォルトの記憶場所を指定します。

デフォルトで作成された OMF データファイルのサイズは 100 MB で、最大サイズは無限で自動的に拡張できます。OMF オンライン・ログのデフォルトのサイズも 100 MB です。

OMF には、多くの利点があります。OS 固有のファイル名とサイズをスクリプトで入力する必要性を排除したことで、データベース・ファイルの作成と削除が簡単になりました。正しくないファイル名の指定という人的エラーの可能性を減らし、古いファイルが占めるディスク領域を減らします。これらの機能すべてを合わせると、表領域が領域外の条件と合致した場合にデータファイルを追加する機能など、ファイルの作成と管理に関連した多くの業務を自動化する機能として非常に適したものになっています。

2 自動取消管理 (AUM)

AUM は、Oracle9i に導入されました。AUM によって、データベース・サーバーは、ユーザーがロールバック・セグメントの作成と管理を気にかける必要性を排除する取消セグメントを自己管理することができます。ユーザーは、もはやトランザクションを特定のロールバック・セグメントに割り当てる必要はなく、ロールバック・セグメントの正しいサイズや数を決定することを気にかける必要もなくなりました。ユーザーの役割は、取消表領域の作成のみで、残りはデータベースがやってくれます。AUM には、たくさんの利点が追加されています。AUM では、サーバーが取消セグメントに割り当てられた領域を動的に調整し、現在のワークロード要件に適合させるため、取消ブロックと読み込み一環の競合が行われる可能性はほぼ完全に排除されます。さらに、取消表領域を正しくサイジングすることで、ユーザーは、懸念される ORA-1555 (スナップショットが古すぎる) エラーの発生を完全に回避できます。構成を決める必要があるのは、取消表領域のサイズのみです。Oracle 10g には、表領域のサイジングをガイドする Undo Advisor が実装されています。しかし、組込みデータベースの場合は、取消表領域を自動的に拡張することをお勧めします。これは表領域の属性で、この機能を利用する場合は「はい」に設定してください。これで、取消表領域は、アプリケーション・ワークロードに基づき必要に応じて自動的に拡張できます。

3 ローカル管理表領域

ローカル管理表領域は、Oracle8i に導入されました。この機能は、辞書管理表領域よりも性能が高く、管理が簡単で、問題に関連した領域断片化をなくします。領域割当および割当解除などの操作の場合、データファイル・ヘッダーに格納されているビットマップが使用され、辞書管理表領域とは違って、集中管理リソースの奪い合いは起こりません。新しいエクステントの割当など、時々辞書管理領域割当に必要な反復操作を繰り返す必要がなくなります。ローカル管理表領域には、自動割当と均一割当の2つのエクステント管理オプションがあります。自動割当の場合は、Oracle サーバー自身がそれぞれに割当したエクステントのサイズを決定します。均一割当の場合は、その表領域のすべてのエクステントが、CREATE TABLESPACE 文の UNIFORM 条項に指定されたサイズとなります。自動割当は、複数エクステントのオブジェクトになる可能性があります。ローカル管理表領域は、パフォーマンスに気づくような影響を与えることなしに多数の(オブジェクト当たり 1000 より多い)エクステントを処理できるため安心です。ユーザーは自動割当を使用されることをお勧めします。したがって、データベース管理者は、エクステント・サイズを気にせずに自動割当オプションを使用してください。

ローカル管理表領域で、組込み環境に危険な2つの重要な問題が解決できます。1番目は、表領域の外部断片化を排除できることです。2番目は、エクステント・サイジングと内部管理を処理できることです。これは、表領域の場合、デフラグによって、失われた領域をリカバーしたりパフォーマンスを強化する必要がありません。非常にコストのかかるこのメンテナンス操作は、ローカル管理表領域を使用すれば不要になります。

4 自動セグメント領域管理 (ASSM)

ASSM は、Oracle9i に導入されました。ASSM は、セグメント内で自動的に領域管理を行います。Oracle9i より前の製品では、FREELISTS、FREELIST GROUPS および PCTUSED などのセグメント属性は、手作業で構成する必要がありました。データの成長またはそれと同時にデータが増加するなど、データベース環境が次第に変化するにつれて、これらの属性はセグメントを作成し、手作業で再構成する必要がある可能性があります。ASSM では、手作業による構成または再構成を必要としません。ASSM では、各データ・ブロックの領域使用管理に FREELISTS ではなくビットマップを使用して、セグメント内の領域管理を完全に透明化しています。また、ASSM では、データ・ブロックの領域使用が改善されます。これは、ビットマップのデータ・ブロック・レベルの空領域の追跡と管理を行う機能が FREELISTS よりも優れているためです。この特長によって、特に行が変化するセグメントの場合、利用可能な空領域をより効率よく再使用できるようになります。さらに、ASSM では、ビットマップの異なる部分を同時に使用して空領域参照用の連続番号を排除しているため、同時進行の DML 操作パフォーマンスが著しく改善されます。

5 オンライン・セグメント圧縮

オンライン・セグメント圧縮は、Oracle 10g の新しい機能です。DML アクティビティが頻繁に挿入、削除されるセグメントは、内部が断片化され、データ・ブロックの中に使用できない空領域を作ります。その結果、セグメント内に無駄な領域ができ、実際に必要な領域よりずっと多くの領域を占拠します。さらに、データ・ブロックのこの無駄な領域は不必要な行のチェーニングを引き起こしパフォーマンスにも悪影響を及ぼします。オンライン・セグメント圧縮を使用すると、このような無駄な領域があるセグメントを圧縮して再生することができます。セグメント圧縮によって、セグメント内を行の断片が動き回り無駄な領域を埋める処理が行われます。また、行のチェーニングができるかぎり取り除かれるため、そのセグメントの読出し操作パフォーマンスが改善されます。この操作は、オンラインで、かつインプレースで行われます。これは、セグメントのアクティビティの圧縮操作が行われている間続行でき、オンライン・データ、再定義および再編成などその他のオンライン操作に必要なものの、つまり一時データを格納する追加領域が不要だという意味でもあります。

Oracle 10g には、無駄な領域の量に基づいて圧縮が必要なセグメントの候補を知らせる Segment Advisor を実装しています。表領域の名前を入力すると、その表領域内のすべてのセグメントの圧縮候補を知らせてきます。毎週実行する Unified Scheduler を使用してジョブを作成し、DML アクティビティを行う Segment Advisor をアプリケーション表領域に呼び出し、アドバイザが薦める圧縮を実行してください。これにより、セグメント内でより効率的に領域を使用できるようになり、領域外の条件に適合する可能性も減少します。次の SQL*Plus スクリプトは、どのように自動的に PL/SQL インタフェースを使用し、Segment Advisor をコールして圧縮候補を推薦するか、その一例を示します。このスクリプトによって、表領域 USERS の Segment Advisor が実行され、その表領域内にある圧縮候補に適したセグメントのすべてが圧縮されます。

```
variable id number;
begin declare
name varchar2(100) ; descr varchar2(500) ; objid number;
begin
name := ' ' ;
descr := 'Segment Advisor Demo';
dbms_advisor.create_task('Segment Advisor', :id, name, descr, NULL) ;
dbms_advisor.create_object(name, 'TABLESPACE','USERS',NULL, NULL, NULL,
objid);
dbms_advisor.set_task_parameter(name, 'RECOMMEND_ALL', 'TRUE') ;
dbms_advisor.execute_task(name) ;
end ;
end ;
/

set echo off
spool shrink.sql
select attr1 || ';' from dba_advisor_actions where task_id=:id ;
spool off
@shrink
```

ローカル管理表領域によって、表領域の領域断片化または外部断片化の問題が処理され、これで、オンライン・セグメント圧縮を実行する通常のジョブを作成し、セグメント領域断片化または内部断片化の問題を排除します。このように、ローカル管理表領域とオンライン・セグメント圧縮の両方を使用することで、断片化が問題となる領域はすべて総合的に解決されます。

前述のソリューションを合わせると、領域管理機能すべてを自動的に実行する総合ソリューションとなります。組み込みデータベース環境でオンサイト・データベース・メンテナンスの必要性をまったく排除するには、これらのソリューションが絶対が必要です。

3. パフォーマンス管理: オプティマイザ統計データ回収

アプリケーション・パフォーマンスは、全体的に、最適な実行計画を作成する Oracle のコスト・ベース・オプティマイザ (CBO) の機能に依存します。CBO は、オプティマイザ統計データとして知られる、オブジェクトとそのデータに関する有効な統計データの可用性に依存し、高速で効率的な実行計画を作成します。無効または古くなったオプティマイザ統計データは、最適とはいえない、または役に立たない実行計画さえも作成して、アプリケーション・パフォーマンスを低下させる可能性があります。オブジェクトによって DML アクティビティの実行につれて、オプティマイザ統計データは典型的なデータではなくなるか、または古くなる可能性があります。これは、オプティマイザ統計データを定期的に取りフレッシュして、CBO が常に利用できる有効な統計データにしておく必要があるということです。特に多量のデータの場合は、このタスクを効率的に行うのは簡単なことです。重要な変更を行う必要があるデータを持つオブジェクトを追跡して統計データを回収しなおし、統計データ回収に適切なサンプル・サイズを決定し、ヒストグラムを必要とする表の列を確認し、適切な平行度を決定して統計データを回収します。この非常に重要なルーチン機能を実施する DBA がいない組み込みデータベース環境では、夜間に、効率的に必要なオプティマイザ統計データを回収するかリフレッシュする、あるいはその両方を行うジョブを組み込む必要があります。

Oracle 10g では、このタスクは完全自動化されています。どの Oracle 10g データベースでも、デフォルトで、古くなったオブジェクトの統計データまたは統計データがない状態をリフレッシュするジョブが作成されます。このジョブは次の処理を行います。

- どのオブジェクトがどの程度修正されているかを追跡する修正監視機能を使用して、統計データ回収候補を特定します。
- 修正の順序にしたがってオブジェクトを分析します。まず、一番修正の多い行があるオブジェクト、次に 2 番目に修正の多い行があるオブジェクトというように分析されていきます。
- 次のことを自動的に決定します。
 - 統計データ回収のサンプル・サイズ。
 - ヒストグラムを必要とする表の列。
 - 統計データ回収ジョブの平行度。
 - 分割された表の細分性、つまり全体レベル、パーティション、またはサブパーティションレベルで統計データを回収するかどうか。
- 辞書オブジェクトの統計データを収集します。これは、Oracle 10g から導入されて CBO がデータ辞書文に使用されているため、非常に重要です。

Oracle 10g では、オプティマイザ統計データの自動回収を有効にするのに特別な構成は必要ありません。そのジョブは、デフォルトで事前に定義されたメンテナンス・ウィンドウで夜間に実行されるよう Unified Scheduler によって組み込まれています。時間や頻度など、このウィンドウのパラメータはすべて変更できます。さらに、Resource Manager を使用して、なんらかの重要なバッチ・ジョブまたはトランザクションがシステム上で実行されているときにそのジョブが CPU のリソースを使いすぎないように、メンテナンス・ウィンドウで CPU 消費を管理できます。Unified Scheduler の他のジョブと同様、オプティマイザ統計データ回収ジョブは無効化できるので注意してください。この理由から、組み込みデータベース・インストールの必須要件として、ジョブは常に有効にしておく必要があります。

適応システムの構成

組込みデータベースは、システム・ワークロードの基本的変動に適応できるよう構成する必要があります。適応できない場合はシステム・パフォーマンスが不安定になり、例外条件に合致して、ORA-4031「共有メモリー・エラー以外」、またはORA-1555「スナップショットが古すぎる」エラーなどが発生する可能性が増加します。この項では、データベースの適合可能性をより高めてワークロードの変化をシームレスに処理できるようにする2つの重要な領域の構成について説明します。

1. 自動メモリー調整

1 自動SQL実行メモリー調整

Oracle9iに、自動SQL実行メモリー機能が導入されました。この機能により、データベース・サーバーは、自動的に、インテリジェントな方法で様々なSQL実行メモリーをアクティブ・プロセス間に配布して、最大のパフォーマンスを発揮し最も効率的なメモリー使用を実現します。これは、システム上のワークロードが変化すると、サーバーは必要に応じてメモリーを様々なプロセス間に割り当てなおし、自動的に変化に適合することを意味します。この動作は、SQL実行メモリー割当てが静的に行われた旧動作とは大きく異なります。SORT_AREA_SIZE、HASH_AREA_SIZE、BITMAP_MERGE_AREA_SIZE、およびCREATE_BITMAP_AREA_SIZEなどの初期化パラメータが、ソート、ハッシュ・ジョインなどの操作を行うSQL作業領域に割り当てられた最大メモリー量をコントロールするように指定されました。各プロセスに割り当てられたメモリーは静的で、ワークロードが変化しても他のSQL作業領域に配布しなおすことはできません。この種のSQL実行メモリー管理では、システムがワークロード1つのみを対象として構成されていたため、ワークロードが変化するシステムでは低いパフォーマンスを示す結果となりました。このことはオンサイトDBAがいるシステムでは、この問題は、違うワークロードには違う構成を設定して回避されました。ワークロードが、たとえばOLTPからバッチに変化すると、それに合わせて初期化パラメータの設定を変更します。オンサイトDBAがいない組込み環境ではできません。このため、自動SQL実行メモリー管理機能はこのような環境には非常に重要です。この機能は、1初期化パラメータPGA_AGGREGATE_TARGETをデータベースが利用できる最大PGAメモリーに設定するだけで利用できます。これは、Oracleサーバーに引き継がれて、SQL実行メモリーは自動的に管理されます。

2 自動共有メモリー調整

自動共有メモリー調整は、Oracle 10gの新しい機能です。自動SQL実行メモリー調整では、プライベートまたはSQL実行メモリーの問題が解決できますが、この自動共有メモリー調整によれば、これと同じ共有メモリーの問題が正確に解決できます。この機能により、バッファ・キャッシュ、共有プール、ラージ・プールおよびジャバ・プールなど、ワークロードの変化に基づいて、様々なメモリー・プール間の共有メモリーを動的に配布しなおすことができます。たとえば、バッチ・ワークロードは、平行な問合せ用のより大きなラージ・プールを必要とする可能性があります。また、OLTPワークロードは、より大きなバッファ・キャッシュを使用してより優れた処理を行うことができます。Oracleでは、自動共有メモリー調整機能によって、バッチ・ワークロード実行中により大きなラージ・プールを割り当ててメモリー・プールのサイズを調整し、ワークロードがOLTPに切り替わる際には、ラージ・プールからのメモリー割当てを解除し、それをバッファ・キャッシュに割り当てて外部の介入なしに最適なOLTPパフォーマンスを発揮します。Oracle 10gより前の製品では、各共有メモリー・プールは、ワークロードが変化すると別個に調整しなおす必要がありました。つまり、DBAのいない環境では、システムを1つのワークロードに最適となるように調整することは可能ですが、ワークロードが変化すると、データベースは標準以下で処理しかできない可能性があり、ORA-4031: xxx バイトの共有メモリー割当て不能などの例外条件に合致する可能性すらあります。自動共有メモリー調整では、異なるワークロードに動的に、かつ外部の介入を受けることなく適合するばかりか、共有メモリーをより効果的に使用し、システムの全体的パフォーマンスを強化

します。この機能は、組込み環境には不可欠です。この機能は、初期化パラメータ `SGA_TARGET` を、データベース・インスタンスが利用可能な総共有メモリー量に設定するだけで利用できます。

2. 自動取消保持調整

自動取消保持調整は、Oracle 10g の新しい機能です。この機能は、自動取消管理 (AUM) が有効の場合に利用できます。この機能を使用すると、取消表領域の指定サイズおよびシステムのワークロード用の取消保持を動的に調整できます。手動取消保持調整によって、ユーザーは、手作業で、初期化パラメータ `UNDO_RETENTION` を任意の最長実行問合せ時間に設定できます。このパラメータが最初に正しく設定されても、データ配布が次第に変化すると、システムの最長実行問合せの時間が変化する可能性があります。この時間が増加した場合は、ORA-1555「スナップショットが古すぎる」エラーが発生する可能性があります。減少した場合は、取消表領域の領域が使用されています。最長の問合せにかかる時間が増加するという前者のシナリオがより一般的に発生し、結果はより深刻となります。例外的に、問合せまたはトランザクション、あるいはその両方が失敗する結果となります。手動取消保持モードで問題を解決するのに利用できる唯一の方法は、手作業による `UNDO_RETENTION` パラメータの変更です。繰り返しますが、組込みデータベースの場合手作業によるパラメータの変更はできません。このため、組込みデータベースは、自動取消保持調整を使用できる構成にする必要があります。そのために、動的に取消保持を変更し、最長の問合せに先んじてシステム上に変更値を維持します。この機能を有効にし、かつ自動拡張可能取消表領域を使用すると (前に推奨したとおり)、ORA-1555 などのエラーが起こる可能性はほとんどゼロになります。

自己依存システムの設計

ルーチン管理業務およびデータベース適合理化構成を自動化すると、オンサイト DBA のいないシステムが必要とする主要要件は達成されます。しかし、完全にセットアップしても、それでもエラーまたは例外が起こります。このため、このような例外を適切に処理し、リカバーして自己修正できる設計にすることが不可欠です。Oracle には、このような自己依存システムを簡単に作成できるソリューションがたくさん実装されています。

自己依存システムには、次の 2 つの処理を行う機能が必要です。第 1 に、例外をトラップまたは特定できること、第 2 に、問題を解決する修正動作が可能なことです。Oracle 10g には、いくつかの最も一般的な種類の例外が発生するとアラートを発生する、新しいインフラストラクチャ Server Generated Alerts が実装されています。これには、表領域が領域外で実行されている場合の領域アラート、ORA-155「スナップショットが古すぎる」メッセージ・アラート、リカバリ領域の領域問題のアラートおよび多数のパフォーマンス・メトリック・アラートがあります。自己依存システムの第 1 要件 - 例外条件をトラップまたは特定する機能を処理します。第 2 の要件である修正動作を提供する機能も、Oracle には実装されています。生成される各種のアラートの Enterprise Manager (EM) では、応答動作は、OS コマンドまたはスクリプトの形で定義できます。この基本的フレームワークは、自己依存組込みデータベースのセットアップに使用できます。次の図 2 に、このセットアップが行える EM インタフェースを示します。

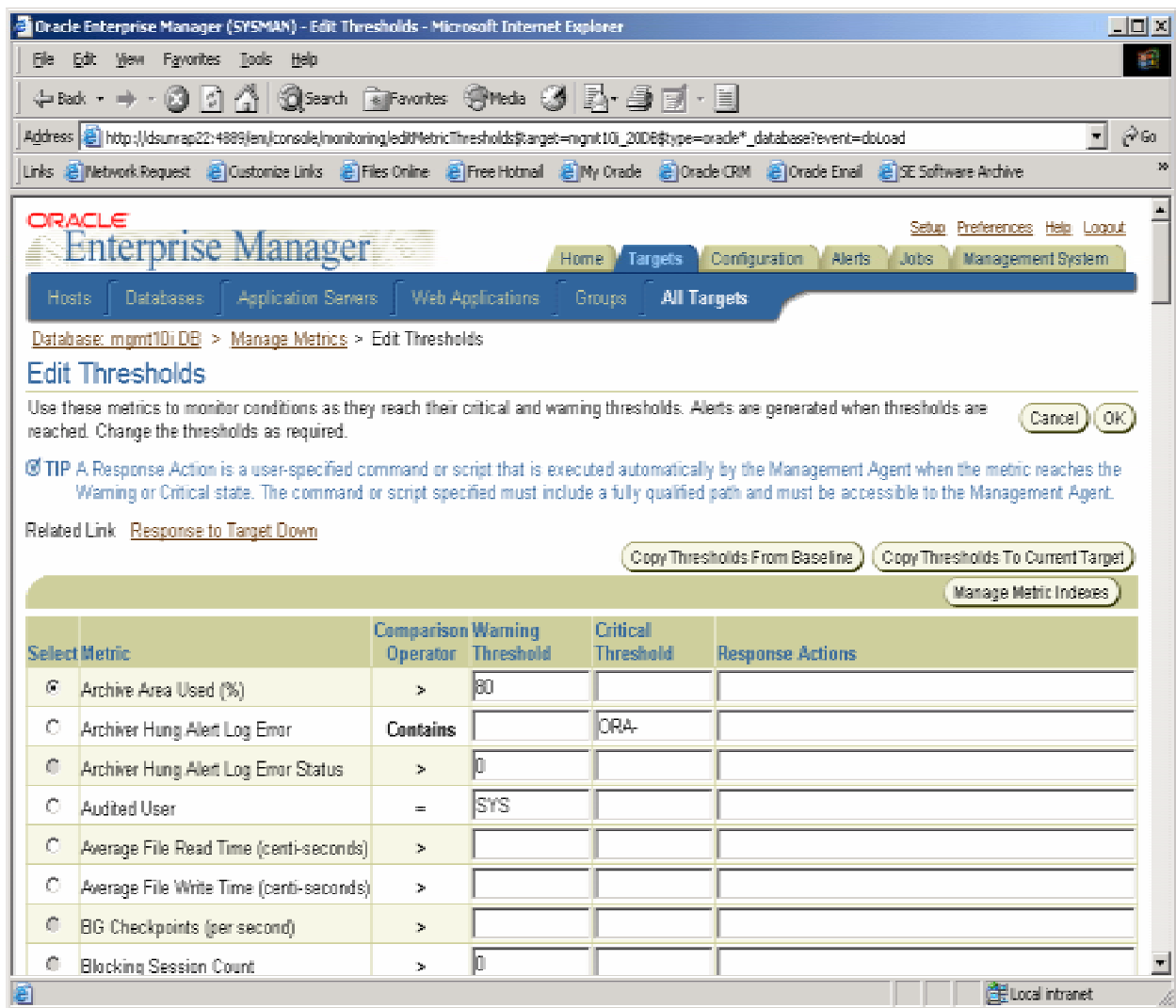


図 2: 応答動作設定用の Enterprise Manager インタフェース

1 領域関連の例外処理

最も一般的な領域問題は、領域外で実行する表領域で発生します。Oracle9i に、再開可能領域割当機能が導入されました。この機能によって、領域外条件に合致する操作をすべて中断するようデータベースに指示して、領域関連の問題がトランザクションを中断しないようにできます。これで、操作が中断されている間に修正動作を行い、問題が解決されてから自動的に実行を再開します。Oracle 10g では、表領域の使用率が 85% になると警告アラートが生成され、97% になるとクリティカル・アラートが生成されます（これらの値はしきい値のデフォルト値で、ユーザーによって変更できます）。このように、再開可能と記されたセッション（セッション内で ALTER SESSION ENABLE RESUMABLE コマンドを完了します）が領域外条件に合致すると、必ず、このセッションが中断モードに入り、自動的に領域アラートが生成されます。応答動作は、問題を修正するスクリプトを実行する EM の特定の表領域に定義できます。修正には、次の 1 つ以上の動作を含みます。

- 懸案の表領域から無駄な領域をリカバーするために、オンライン・セグメント圧縮を使用します。これで、新しいデータファイルを追加することなく問題を修正できます。次の SQL コマンドで、指定のセグメントが圧縮されます。

```
ALTER TABLE <table_name> SHRINK SPACE;
```

- 新しいデータファイルを表領域へ追加します。このスクリプトは、すでに説明した Oracle 管理ファイル(OMF)機能を使って、データファイルの実際の名前、記憶場所またはサイズをハードコードせずに、完全に一般化することができます。

2 ORA-1555 「スナップショットが古すぎる」エラー処理

Oracle 10g では、ORA-1555 が発生した場合にもアラートが生成されます。このアラートの場合、修正動作には、自動取消保持調整を有効にし、取消表領域サイズを増加する動作が含まれます。より高度な処理を行う場合は、PL/SQL パッケージ、DBMS_UNDO_ADV およびその出力を使用して Undo Advisor をコールし、正確な必要な領域を決定できます。この方法で、ORA-1555 の修正に必要な取消表領域の実際のサイズを決定するスクリプトが作成でき、新しいデータファイルを追加して取消表領域を新しいサイズに変更できます。この方法では、例外条件を修正してから、トランザクションまたは問合せを再発行して完了できます。そのため、外部からの介入の必要性を未然に防ぎます。

3 リカバリ領域アラート処理

このアラートは、削除できるファイルがなくなってフラッシュ・リカバリ領域に領域圧力がかかった場合または RMAN で書き込みを行う必要がある場合に、Oracle 10g に生成されます。このアラートを生成する適切な応答動作に、次の 1 つ以上の動作を含めることができます。

- フラッシュ・リカバリ領域に、より多くの領域を割り当てます。
- テープまたはその他の第 3 デバイスにファイルをバックアップします。
- 保持ポリシーを削減します。これは、次の RMAN 構成コマンドで実施できます。

```
RMAN CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF <x> DAYS;
```

- 次のコマンドを使用して、増分バックアップをイメージ・コピーにマージします。

```
RECOVER COPY OF DATAFILE <datafile# or name>
```

これで、増分バックアップ・ファイルを削除できます。

これらの動作で、REDO ログがアーカイブできずデータベースがハングしたり、リカバリ領域の領域不足のため通常のバックアップができないなどの問題を修正できます。

Oracle 10g には、自己依存システム設計用の基本的インフラストラクチャが実装されています。わずかの作業で、より一般的な問題のいくつかを簡単に処理できます。そのいくつかの問題とは、前述した問題のことです。このインフラストラクチャは、アプリケーション・ユーザーにほとんどみえない本当の自己依存データベースを作成するのに重要です。

問題の簡単な診断と解決

組込みデータベースには、問題が発生したとき、診断と解決の簡単な方法を実装する必要があります。完璧な組込みデータベース・インストールであっても、管理上の介入が必要となる問題が発生します。このようなケースは組込みデータベースではごくまれですが、データベースに問題が発生したとき、問題を検出し、診断および解決する簡単な方法が実装されている必要があります。組込みインストールの大多数は、データベース・システム内部に大きな技術的ノウハウが実装されていないため、初心者ユーザーでも直接かつ簡単に問題の診断と修正が行えるプロセスがさらに重要となります。この問題の解決に、Oracle 10g では、データベースでの問題の発生を予見し、診断できる新しい自己診断機能が導入されました。

自動データベース診断モニター (ADDM)

ADDM は、データベース全体の包括的なリアルタイムのパフォーマンスを画像で提供する新しいデータベース・ソリューションです。このソリューションは、30 分ごとにパフォーマンスに問題がないか調べます。また、必要なときはいつでも実行できます。1 回実行されるたびに、結果を記したレポートを作成します。結果は、問題のない領域だけでなく問題の出所も強調して表示します。問題が検出されると、根本原因を特定して、問題を解決する適切な方法を薦めてきます。ユーザーの介入を必要とするデータベースにパフォーマンスの問題が発生した場合、ユーザーが最初に閲覧する必要があるのは ADDM レポートのみです。このレポートには問題が列挙され、推薦する適切なソリューションにパフォーマンスの利点を書き添えてあります。ADDM には、PL/SQL と Oracle EM インタフェースが実装されています。ISV は、PL/SQL インタフェースを使用して、ADDM 機能を自分のアプリケーションに組み込んだり、EM インタフェースを使って ISV の顧客データベースを遠隔診断できます。また、ISV は、ADDM PL/SQL インタフェースを使用して、自動的に ADDM レポートを配布して事前にサポートができます。PL/SQL インタフェースを使用できるように、Oracle 10g には、新パッケージ DBMS_ADVISOR が実装されています。次に、コマンド・ラインから DBMS_ADVISOR パッケージを使用する方法を 3 つ紹介します。例 1 に、手作業で SQL*Plus から ADDM の実行方法を示します。

例 1: 手作業による ADDM の実行

```

set serveroutput on;

begin
  declare
    id number; name varchar2(100); descr varchar2(500);
  BEGIN
    name := ' ' ; descr := 'manual addm run';
    dbms_advisor.create_task('ADDM',id,name,descr,null);
    dbms_advisor.set_task_parameter(name, 'START_SNAPSHOT', 420);
    dbms_advisor.set_task_parameter(name, 'END_SNAPSHOT', 421);
    dbms_advisor.set_task_parameter(name, 'INSTANCE', 1);
    -- execute task
    dbms_advisor.execute_task(name);
  end;
end;
/

```

例 2 には、特定の ADDM タスクの ADDM レポートを閲覧するための SQL*Plus スクリプトを示します。このスクリプトは、@<Script_Name> <TASK_NAME>として SQL*Plus から実行されます。

例 2: 特定のタスクの ADDM レポート

```

SET LONG 1000000
SET PAGESIZE 0
SET LONGCHUNKSIZE 1000
SET VERIFY OFF
COLUMN get_clob format a80

select dbms_advisor.get_task_report('&1', 'TEXT', 'ALL') from dual;

```

初心者ユーザーであっても、ADDM には、非常に簡単にかつ直接、パフォーマンスの問題すべてを診断して解決できる方法が実装されています。ADDM は、その強力なパワーと簡単な使用方法ゆえに、ユーザーの介入が必要なパフォーマンス問題がまれに発生したときの組込みデータベース・インストールに絶対に不可欠なものになっています。

サーバー生成アラート

ADDM で、パフォーマンス領域の問題が解決できます。パフォーマンス以外のすべての問題には、新しいサーバー生成アラート・フレームワークからソリューションが提供されます。これらのアラート機能には、即時であれ潜在的であれデータベースに発生する多くの深刻な問題をキャプチャする機構が搭載されています。デフォルトでは、次のアラートが有効です。

- 1 表領域使用率: 表領域の使用率が 85%に達すると警告アラートを、97%に達するとクリティカル・アラートを送出します。これらの数値はデフォルト値で変更できます。
- 2 ORA-1555「スナップショットが古すぎる」: このエラーが発生したときは必ずアラートが生成されます。

これらのアラートによって、データベースに発生する可能性のある一般的な非パフォーマンス関連の例外のほとんどがキャプチャされます。これらのアラートを生成、管理、分析するフレームワークを1つ実装したことで、非パフォーマンス関連の問題を検出するタスクが非常に簡単になりました。ユーザーは、DBA_ALERT_HISTORY と DBA_OUTSTANDING_ALERTS を問合せただけで、非パフォーマンス関連の問題が特定できます。

ADDM とサーバー生成アラート間に、データベース問題の検出、診断、解決を行う包括的で簡単な方法が存在します。この2つの機能のインタフェースと使用モデルは、単純で複雑なものではありません。データベースのノウハウがほとんど、またはまったくない組込み環境では、これらの機能を、全包括的な問題の診断と解決に使用する必要があります。

ソフトウェア・メンテナンスとサポート

組込み環境で解決の必要がある3つ目の問題は、ソフトウェア・メンテナンスとサポートの問題です。ソフトウェア・パッチを適宜適用して、クリティカルな問題の修正が必要です。Oracle では、主に次の3種類のソフトウェア更新が利用できます。

- パッチ
- パッチセット、および
- リリース更新

これらのいずれかで自分の組込みアプリケーションを更新して、自分の顧客に対応する必要な場合があります。Oracle には、それぞれの顧客に合った異なるツールとアプローチが実装されています。

パッチおよびパッチセット

パッチおよびパッチセットは、既存のバイナリを新しいバイナリに置き換えるだけのものです。パッチは、OPATCH という Oracle ユーティリティを使用して適用し、パッチセットは、Oracle Universal Installer (OUI) を使用して適用します。これらのツールはいずれも、サイレント・モードで Oracle バイナリの更新を行えます。いくつかのクリティカル問題を修正するソフトウェア・パッチを適用するか、または同じリリースの最新バイナリを使用して最新状態に維持するパッチセットを適用する必要がある可能性があります。さらに、パッチセットを適用して、パッチセットの中の既知の不具合修正機能を使用する必要があることもあります。この操作を簡単にするには、適用する必要がある関連パッチとパッチセットを簡単に特定する方法が必要です。Oracle 10g には、まったく新しい Oracle Enterprise Manager (EM) が実装されています。新しい EM は、どのレベルでもファイアウォール全体で動作する設計となっており、軽量の HTML インタフェースが搭載されています。EM には、Oracle プラットフォームの各バージョンのシステムのハードウェア、ソフトウェアおよびネットワーク、

OS の種類およびそのバージョン、パッチ・レベル、依存性などに関するすべての構成情報を追跡する Enterprise Configuration Management (ECM) というコンポーネントが搭載されています。また、直接 Oracle サポート (Metalink) とも統合されて、パッチまたはパッチセットの自動的な更新を行います。ECM には、次の処理を行う Patch Wizard があります。

- 自動的に、特定のパッチが適用できる顧客を特定するウィザード
- 自動的に、アプリケーション用パッチをダウンロードして設定するウィザード

ISV は、特定のパッチが利用可能となったとき、そのパッチを必要としている顧客がわかるよう、顧客をサポートする異なる種類のデータベース構成を使用して ECM をセットアップできます。ISV は、パッチをダウンロードして適切な顧客に配布するか、自分の Web サイトにのせて顧客がそこからダウンロードできます。さらに、Oracle 10g では、サイレン

ト・モードのパッチセット/パッチ・インストールもサポートしています。この機能で、ISV は、アプリケーション・パッチ適用機構の一部としてパッチセットとパッチをパッケージ化し、インストール用として顧客に出荷できます。

リリースの更新

主要データベース・リリースを更新するには、新しいリリースを別の Oracle ホームにインストールする必要があります。しかし、メンテナンス・リリースの更新は、同じ Oracle ホーム上で行えます。データベース・リリースの更新は、Oracle バイナリをサイレント・モードで簡単にインストールできます。既存の Oracle データベースを Oracle データベースの新しいリリースに更新するツールとして、Oracle には、Database Upgrade Assistant (DBUA) というツールが搭載されています。このツールには、サイレント・モードでデータベースを完全更新するためのコマンド・ライン・インタフェースが搭載されています。DBUA は、更新前にデータベース・バックアップを行うなどのデータベース更新タスクをすべて自動で行って、古くなったパラメータを置き換え、更新が正しく完了したことを確認します。

組込みに利用可能なツール

Oracle には、Oracle9iR2 と Oracle 10g のデータベースの詳細な組込み手順を示す Oracle 組込みインストール (OEI) キットが実装されています。このキットのサポート対象は、Oracle パートナのみです。Oracle パートナは、<http://opn.oracle.com> の Oracle Partner Network からキットをダウンロードできます。

結論

Oracle Database 10g は、組込みデータベースのあらゆる問題を総合的に解決する初めてのデータベースです。このデータベースは、エンド・ユーザーにまったくみえない方法で配置し、実行します。このデータベースのサイレント配置、柔軟性のあるパッケージ化、自己管理機能、拡張性および信頼性は、従来のパフォーマンスの特長と組み合わせることで、組込み環境に理想的なデータベースを作りあげています。

付録 A:

DBCA を使用したデータベースのパッケージ化 (データファイルあり、またはなし)

次に、既存のデータベースを DBCA テンプレートにパッケージ化する詳細手順を説明します。

1. DBCA を Program Manager (Windows の場合のみ) またはデータベースが存在するマシン上で dbca としてコマンド・ラインを実行します。
2. 「次へ」ボタンをクリックして次に進みます。
3. 「テンプレートを管理する」を選択します。
4. この画面で、データベース・テンプレートを作成する方法を、次の 3 つのオプションから選択できます。
 - a. 既存のテンプレートから作成する: 既存のテンプレートに変更を加えて別のデータベース・テンプレートを作成します。このオプションは無視してください。
 - b. 既存のデータベースから作成する (構成のみ): テンプレートを作成しますが、データベースの内容 (データファイル) はパッケージしません。同じデータベースを作成するのにデータベース作成スクリプトのみをパッケージ化する場合に選択してください。このオプションではデータファイルはパッケージされないため、領域が節約できます。このアプローチではインストール時にデータベースを作成するのにかなりの時間を要する場合があります。ISV は、インストール後にスキーマ・オブジェクトをロードする必要もあります。
 - c. 既存のデータベースから作成する (データと構成の両方): このオプションをお勧めします。このオプションは、データベース構成だけでなく、データベースの内容もパッケージします。このオプションを使用すると、ISV は、アプリケーション・スキーマをデータベースにプリインストールして、インストール時間を大幅に短縮できます。
5. ソース・データベースを選択して、ユーザーSYS として接続します。
6. テンプレート名と内容を入力します。データファイルをデフォルトに保持する圧縮ファイル用の記憶場所を残しません。このテンプレート名は、データベースを作成するインストール時に引数として使用されます。このため、テンプレート名の領域を使用せずに短いままにしておきます。
7. この画面で、「ファイルの記憶場所を変換して OFA を使用する」を選択し、「終了」をクリックします。このオプションを選択すると、テンプレートの定義が有効になり、現在のデータベース・ファイルのハードコード・パスのかわりにデータベース・ファイルの相対的な記憶場所を保持することができます。
8. この後に、「サマリー」画面が現れます。この画面には、実行される操作と作成されるテンプレート・ファイルが表示されます。「OK」ボタンをクリックすると、テンプレートの作成を開始します。
9. 「テンプレートの作成を完了しました」というメッセージが表示されます。「いいえ」を選択して終了すると、「他の操作を実行しますか」と表示されます。
10. この時点で、テンプレート・ファイルが 1 つか 2 つ、ORACLE_HOME/assistants/dbca/template ディレクトリに作成されます。これらのファイルは、<テンプレート名>.dbc (テンプレート定義) および <テンプレート名>.dfb (圧縮データファイル) となります。しかし、手順 4 でオプション b. を選択すると、データベース構成の定義を含む <テンプレート名>.dbt ファイルが 1 つだけ作成されます。

付録 B:**カスタマイズ・インストールのデータベースのパッケージ化**

Oracle シード (汎用) データベースを置き換えないで自分のデータベースをパッケージ化する場合は、Oracle に実装されているシード (汎用) データベースをパッケージから削除して、領域を節約すると便利です。次に、Oracle に実装されているシード (汎用) データベースをパッケージから削除する詳細手順を説明します。

- 1 ORACLE CD 全体を自分のハードディスクにコピーします。これを、ShipHome と呼びます。
- 2 ShipHome で、<ShipHome>\Disk1\stage\Components\oracle.rdbms.seeddb\10.1.0.1.0\1\DataFiles\Expanded\seed\templates にある Data_Warehouse.dfb データファイルを 0 バイトのファイルに置き換えます。

前述のプロセスで、自分のパッケージ化されたデータベースにインストール用としてパッケージ化して出荷できる新しい Oracle ShipHome ができあがります。絶対に、この ShipHome から Oracle のシード (汎用) データベースを作成しないでください。エラーが発生します。

Oracle のシード (汎用) データベースをパッケージから削除した後は、簡単に自分のデータベース・テンプレート・ファイルを自分のパッケージの任意の場所にコピーできます。しかし、インストール時に、これらのファイルを自分のターゲット・マシンの ORACLE_HOME/assistants/dbca/template ディレクトリにコピーする必要があります。

付録 C:**Oracle シード (汎用) データベースをカスタム・データベースに置換**

次に、Oracle に実装されているシード (汎用) データベースをインストールする自分のパッケージ化されたデータベースに置き換える詳細手順を説明します。

- 1 自分のデータベースを含むデータベース・テンプレートを取得します。(MySeedDBTpl.dbc および MySeedDBTpl.dfb です。)
- 2 ORACLE CD 全体を自分のハードディスクにコピーします。これを、ShipHome と呼びます。
- 3 DataFiles ディレクトリを<ShipHOME>\Disk1\stage\Components\oracle.rdbms.seeddb\10.1.0.1.0\1から一時記憶場所にバックアップします。(component temp と呼びます。)
- 4 次のコマンドを使用して、dbc.1.1.jar のファイルを他の一時記憶場所 (template temp) に抽出します。

```
jar -xvf dbc.1.1.jar
```

注: jar は、JavaSoft から JDK の一部として含められたユーティリティです。

- 5 この template temp ディレクトリ構成の中から、General_Purpose.dbc0 ファイルまたは General_Purpose テンプレートと呼ばれるものはすべて削除します。
- 6 以前作成した自分のデータベース・テンプレート定義ファイル (MySeedDBTpl.dbc) をこの記憶場所にコピーして、General_Purpose.dbc0 または以前の名前に変更します。
- 7 このファイルを編集して、次のように<Location>エレメントを変更します。

次の状態から、

```
<StorageAttributes>
<DataFiles>
<Location>{ORACLE_HOME}\assistants\dbca\templates\MySeedDBTmp1.dfb
...
```

次のように変更します。

```
<StorageAttributes>
<DataFiles>
<Location>{ORACLE_HOME}\assistants\dbca\templates\Data_Warehouse.dfb
...
```

- 8 次のコマンドを使用して、template temp のファイルをすべて jar にパッケージします。

```
jar -cvf dbc.1.1.jar *
```

注: 必ず、抽出したとおりにまったく同じファイルのみをパッケージ化してください。

- 9 この dbc.1.1.jar を component temp 記憶場所にコピーして既存のファイルを置き換えます。
- 10 DataFiles \ Expanded \ seed \ templates の元の Data_Warehouse.dfb データファイルを component temp ディレクトリから削除します。
- 11 ここで、自分のテンプレート圧縮データファイル (MySeedDBTmp1.dfb) をこの記憶場所にコピーして、名前を Data_Warehouse.dfb に変更します。
- 12 自分の component temp ディレクトリの内容をすべて <ShipHome> \ Disk1 \ stage \ Components \ oracle.rdbms.seeddb \ 10.1.0.1.0 \ 1 \ にコピーします。

前述のプロセスで、Oracle のシード (汎用) データベースとして自分のパッケージ化されたデータベースへのインストール用として、パッケージ化して出荷できる新しい Oracle ShipHome ができあがります。