



Oracle Database 12c Release 2 (12.2.0.1)へのアップグレードと移行

Oracle ホワイト・ペーパー | 2016年12月



ORACLE



免責事項

下記事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないで下さい。オラクルの製品に関して記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

目次

免責事項.....	ii
はじめに.....	1
アップグレードと移行の比較.....	2
データベースのアップグレード.....	2
データベースの移行.....	2
マルチテナント・アーキテクチャ.....	3
データベースのアップグレードまたは移行方法の選択.....	3
Oracle Database 12c Release 2 への直接アップグレード.....	4
アップグレードおよび移行方法の詳細.....	5
方法 1：DBUA またはコマンドライン・アップグレード.....	5
方法 2：フル・トランスポータブル・エクスポート/インポート またはトランスポータブル表領域.....	10
方法 3：Oracle Data Pump エクスポート/インポート.....	14
方法 4：オリジナルのエクスポート/インポート.....	16
停止時間を最小限に抑えたデータベース・アップグレード.....	17
Oracle Data Guard によるローリング・アップグレード.....	17
Oracle RMAN のクロス・プラットフォーム増分バックアップ.....	18
Oracle GoldenGate.....	19
まとめ.....	20
付録 A：アップグレードと移行に関する Oracle Database 12c (12.1) の新機能.....	21
付録 B：アップグレードと移行に関する Oracle Database 12c Release 2 (12.2) の新機能.....	22



はじめに

Oracle Database 12c Release 2 (12.2.0.1) では、新機能の追加とともに一部の機能が強化されており、既存の Oracle データベースをアップグレードする上での大きな魅力となっています。Oracle Database 12c Release 2 への移行は、新しく購入したサーバー・ハードウェアへの移行、Oracle Automatic Storage Management などの異なるストレージ・アーキテクチャへの移行、データベース・キャラクタ・セットの変更のほか、完全に異なるオペレーティング・システムや Oracle Cloud への移行などと合わせて行うこともあるでしょう。

アップグレードおよび移行のシナリオは置かれた状況がさまざまであるため、オラクルでは、データベースを Oracle Database 12c Release 2 にアップグレードおよび移行するための複数の方法を提供しています。このホワイト・ペーパーでは、これらのアップグレードおよび移行方法の概要を説明します。さまざまなユースケースを紹介し、個々の要件に最適な方法を選択する際に考慮するおもな要因について説明します。

「DVAG 社では、3 ステップからなるアプローチで重要な環境を徹底的にテストします。テスト・サイクル完遂一回分と、それに伴うコストも同時に削減できるよう、前データベースリリースのターミナル・パッチ・セットをスキップし、Oracle Database 12c に直接アップグレードするという決断を下しました。」

Deutsche Vermögensberatung AG, Michael Kuhn 氏

アップグレードと移行の比較

「アップグレード」と「移行」という用語は、他の状況では同義語として使用されることが多いものの、「データベースのアップグレード」と「データベースの移行」には違いがあります。この違いを把握することが、プロジェクトに最適なアップグレードまたは移行方法を選択する際の最初のステップとなります。

注：「移行」という用語は、Oracle 以外のデータベースから Oracle にデータを移動する場合にも使用されます。このホワイト・ペーパーでは、ソースとターゲットの両方が Oracle データベースである場合の移行についてのみ説明します。

データベースのアップグレード

Oracle データベースのアップグレード操作には、新しいバージョンの Oracle データベース・ソフトウェアとの互換性を保つため、データ・ディクショナリを変更する操作が含まれます。データベースのアップグレードに含まれる代表的な操作には、次のようなものがあります。

- » システム表とシステム・ビューの列の追加、削除、または変更
- » システムの新しいパッケージまたはプロシージャの作成
- » システムの既存のパッケージまたはプロシージャの変更
- » データベース・ユーザー、ロール、および権限の作成、変更、または削除
- » Oracle データベースのコンポーネントで使用されるシード・データの変更

これらのすべての操作がデータベースのデータ・ディクショナリに影響します。ユーザーやアプリケーションの表領域に保存されているデータには影響しません。そのため、Oracle データベースに保存されている膨大な量のデータがデータベースのアップグレードに影響することは、ほとんどないか、まったくありません。

データベースの移行

「移行」とは、Oracle データベースがサポートする、設定変更を伴う作業を指します。このような変更には、データベース・バージョンに加えて、以下のいずれかまたはすべてに対する変更が含まれます。

- » コンピュータ・サーバー
- » ストレージ・アーキテクチャ
- » キャラクタ・セット
- » オペレーティング・システム
- » スキーマ・トポロジ（パーティション化スキームの変更）
- » 暗号化
- » 圧縮
- » データベース・アーキテクチャ（非 CDB から PDB への移動）

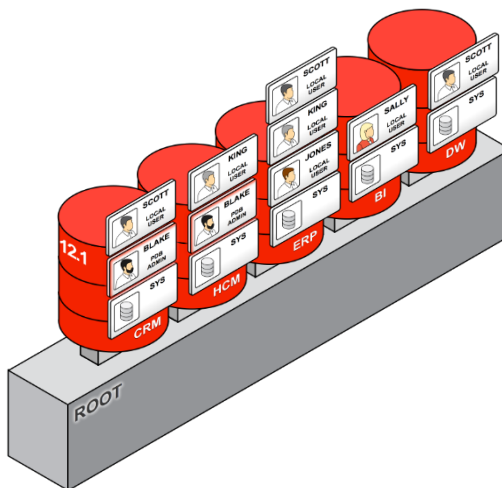
データベースの移行は、2つの重要な点でデータベースのアップグレードと異なります。1つ目は、データベースの移行には通常、データベース内のユーザー・データとアプリケーション・データの

移動または変更が伴う点です。つまり、データベースのサイズがデータベース移行プロジェクトに大きく影響します。2 つ目は、上記のいずれの移行も、データベース上で実行でき、新しいバージョンにアップグレードする必要がない点です。このホワイト・ペーパーで説明する方法を使用すると、Oracle Database 12c へのアップグレードおよび移行の両方を同時に実行できます。

マルチテナント・アーキテクチャ

Oracle Database 12c には、新しいマルチテナント・アーキテクチャが導入されました。このアーキテクチャにより、Oracle Net クライアントに個別のデータベースとして表示される、スキーマ、スキーマ・オブジェクト、および非スキーマ・オブジェクトのポータブル・コレクションを Oracle データベースに含めることができます。この自己完結型のコレクションはプラグブル・データベース (PDB) と呼ばれています。マルチテナントのコンテナ・データベース (CDB) は、ユーザーが作成した PDB がまったく含まれないか、1 つ、または多数の PDB が含まれる、Oracle データベースです。この新しいアーキテクチャにより、複数のデータベースの統合が容易になっただけでなく、PDB への移行という、データベース移行に非常に有用でお客様が検討すべき方法をとることが可能となりました。

従来のデータベース・アーキテクチャ (Oracle Database 11g など) から PDB への移行を、他のデータベース・アップグレード・タスクやデータベース移行タスクと組み合わせて 1 回の操作にできる場合もあれば、PDB への移行に追加の手順が伴う場合もあります。既存のデータベースや非 CDB データベースを PDB に移行する手順については、このホワイト・ペーパーの各アップグレードまたは移行方法のセクションで説明します。



データベースのアップグレードまたは移行方法の選択

アップグレードおよび移行にはいくつかの方法がありますが、個々のプロジェクトに最適なアップグレードまたは移行方法を選択するには、プロジェクトで重要となる特性を分析する必要があります。Oracle Database 12c にアップグレードまたは移行するときに、これらの各特性に最も適した方法を選択することになります。

- » パッチ・セット・レベルにいたるまでの、アップグレードまたは移行するソースのバージョン
- » ソースとターゲットのオペレーティング・システムとバージョン
- » ソースとターゲットのハードウェア・プラットフォームとこれらのエンディアン特性
- » 実際のデータ・レイアウトやデータ形式の変更計画 (キャラクタ・セット、パーティション化、暗号化、圧縮の変更など)
- » 可用性要件 - アップグレード・プロジェクトまたは移行プロジェクトで許容される停止時間の長さ
- » 移行するデータベースのサイズ
- » アップグレードのターゲットが Oracle Database 12c Release 2 PDB かどうか

考えられるすべてのアップグレード・シナリオと移行シナリオに最適な、単一のアップグレード方法や移行方法はありません。逆に、上記の特性に基づくそれぞれのシナリオごとに、最適な方法が見つかるはずで

Oracle Database 12c Release 2への直接アップグレード

直接アップグレードでは、Database Upgrade Assistant (DBUA) またはコマンドライン・アップグレード・スクリプトを使用して Oracle Database 12c にアップグレードします。直接アップグレードは、次の表に示すいずれかのリリースがソース・データベースで実行されている場合にサポートされます。

表1: Oracle Database 12c Release 2の直接アップグレード・パス

ソースのリリース	ソースのパッチ・セット	直接アップグレードのサポート
Oracle Database 12c	12.1.0.1、12.1.0.2	あり
Oracle Database 11g Release 2	11.2.0.3、11.2.0.4	あり
	11.2.0.1、11.2.0.2	なし (別の方法を使用する必要があります)
Oracle Database 11g Release 1 以前	すべてのパッチ・セット	なし (別の方法を使用する必要があります)

直接アップグレードがサポートされない場合、Oracle Database 12c Release 2 への他の移動方法が適用されます。これらの方法については、次のセクションで説明します。

アップグレード方法と移行方法

このホワイト・ペーパーでは、4 つの異なるアップグレードおよび移行方法を説明しており、これらの方法のうち3 つには、それぞれの状況に合わせたいくつかの選択肢があります。

これらの方法は次のとおりです。

このホワイト・ペーパーで説明する4つのアップグレードおよび移行方法はすべて、PDB への移行に使用できます。

1. DBUA またはコマンドライン・アップグレード・スクリプトを使用した、データベース・アップグレード
2. Oracle Database 12c の機能であるフル・トランスポートブル・エクスポート/インポートまたは従来のトランスポートブル表領域 (TTS) モードを使用した、TTS のエクスポートとインポート
3. ダンプ・ファイルまたはネットワーク・モードを使用した、Oracle Data Pump エクスポート/インポート
4. オリジナルのエクスポート/インポート・ユーティリティ

次の表に、前のセクションに示したプロジェクト特性に基づいて、個々のシナリオでのこれらのアップグレードおよび移行方法の適用性について概要を示します。

表2：データベースのアップグレードおよび移行方法

方法	複雑さ	時間	ソースの最低バージョン	新規サーバーへの移行	0/Sの変更	データ・レイアウト、キャラクタ・セット、暗号化、圧縮の変更
アンプラグ、プラグイン、アップグレード	中	最短	12.1.0.1	あり	なし	なし
Database Upgrade Assistant	低	最短	11.2.0.3	なし	なし	なし
コマンドライン・アップグレード	中	最短	11.2.0.3	あり	なし	なし
フル・トランスポート フル・エクスポート/ インポート	中	より短い	11.2.0.3	あり	あり	なし
トランスポート 表領域	高	より短い	8.1.5	あり	あり (10.1以降)	なし
Data Pump expdp/impdp	中	短い	10.1	あり	あり	あり
オリジナルの エクスポート/ インポート	中	長い	5	あり	あり	あり

アップグレードおよび移行方法の詳細

このセクションでは、各アップグレードおよび移行方法について、機能する仕組み、利点、および各方法を選択・決定する際の考慮事項について説明します。

方法1：DBUAまたはコマンドライン・アップグレード

Oracle Database 12c Release 2 へもっとも容易にバージョンアップする方法は、DBUA またはコマンドライン・アップグレード・スクリプトを使用して既存のデータベースをアップグレードすることです。この方法には、おもにユーザー・インターフェースが異なる 2 つの進め方があります。いずれも、コピーや新しいインスタンスを作成しなくてもデータベース上で機能するため、「インプレース」のアップグレードとも呼ばれます。

Database Upgrade Assistant

Database Upgrade Assistant は、データベース・アップグレード・プロセスを、順を追って示すグラフィカル・ユーザー・インターフェース (GUI) を備えており、表示される一連の画面でデータベース・アップグレードのオプションを指定できます。アップグレード・プロセス中に、コマンドライン・アップグレードに使用すると同じスクリプトが DBUA によって起動されます。アップグレード前の検証ステップも実行され、アップグレード後のタスクを自動化できます。DBUA を使用すると、データベース・アップグレードで必要となる手動の操作を大幅に低減できますが、コマンドライン・アップグレードと比較して柔軟性は低くなります。DBUA の詳細については、[Oracle Database アップグレード・ガイド](#)を参照してください。

Database Upgrade Assistant の使用例

この DBUA ワークフロー例のように、ダイアログ・ボックスを使用してデータベース・アップグレードのオプションを選択できるため、GUI ベースのユーティリティとして DBUA は非常に簡単に使用できます。

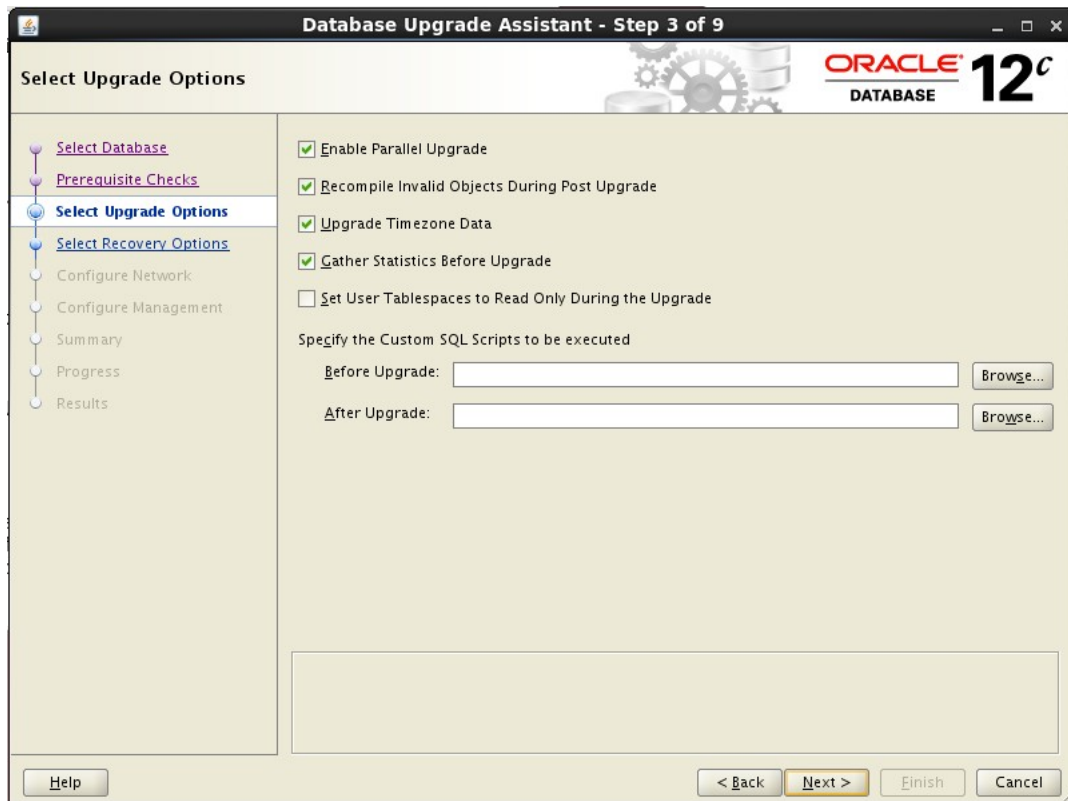


図1 : DBUAのUpgrade Optionsページのスクリーンショット

このスクリーンショットでは、ワークフローの手順がウィンドウの左側のペインに表示されており、現在の手順（強調表示）に使用できるオプションが右側のペインに表示されています。オプションはタブに分けて表示され、現在のタブはこの操作の Upgrade Options です。オプションには次のようなものがあります。

- » Enable Parallel Upgrade。Oracle Database 12c で導入されたパラレル・アップグレードを使用してアップグレードを加速させることで、停止時間を短縮できます。
- » Recompile Invalid Objects After The Upgrade。このオプションを選択しない場合は、アップグレードの終了後、`utlrip.sql` を実行してオブジェクトを再コンパイルする必要があります。
- » Upgrade Timezone Data。ソース・データベースに最新バージョンのタイムゾーン定義がインストールされていない場合、アップグレード中にタイムゾーンのバージョンをアップグレードすることを推奨します。タイムゾーン・データのアップグレードはデータベースのアップグレード自体とは異なり、アップグレード後のタスクとして実行されます。
- » Gather Statistics Before Upgrade。データベースのアップグレード前にディクショナリ統計を収集すると、アップグレードのパフォーマンスを向上させることができます。
- » Set User Tablespaces Read Only During Upgrade。このオプションは、アップグレードが中断された場合を想定したフォールバック戦略の一つとして役立ちます。
- » Specify Custom SQL Scripts to run Before or After the Upgrade。このオプションは、アップグレード・プロセス中に環境全体で管理スクリプトや監視スクリプトを実行したい場合に役立ちます。

これらすべてのオプションや操作は、コマンドライン・アップグレード中のさまざまな時点で指定できますが、DBUA の非常に便利なインタフェースによりアップグレード・プロセスが自動化され

ます。そのため、アップグレード中の問題を引き起こす、誤字などの間違いが発生する可能性を低減できます。

「新しいパラレル・アップグレード・スクリプトを使用すると、計画的なメンテナンスで生じる停止時間を大幅に短縮できます。当社の環境では、以前のアップグレード・プロセスと比較して、停止時間が 37%短縮されました」

Payback GmbH, Leiter Datenbanken, Harald Stefan 氏

Database Upgrade Assistant の使用に関する考慮事項

一般に、現在のハードウェア・システム上で Oracle データベースをアップグレードする場合にのみ DBUA を使用できます。これは、DBUA はソースの Oracle ホームを使用してアップグレード前の検証ステップを実行し、その後、ターゲットの Oracle ホームへ切り替えてアップグレードとアップグレード後のステップを実行するためです。

DBUA を使用すると、ユーザーはデータベースのアップグレードに使用される並列度を制御できません。

DBUA では、データベース・アップグレード前のスクリプトに記述される多数のアップグレード前タスクが自動化されますが、一部のアクションには引き続き手作業が必要な場合があります。

コマンドライン・アップグレード

Oracle Database 12c では、新しいコマンドライン・アップグレード・ユーティリティ (`catct1.pl`) が導入されました。このユーティリティは、以前のリリースの Oracle Database にアップグレードするために使用されていた、`catupgrd.sql` スクリプトに代わるものです。この新しいコマンドライン・アップグレード・ユーティリティを使用すると、データベース・アップグレード中のパラレル処理が可能になるため、アップグレードのパフォーマンスが向上し、データベースの停止時間が短縮されます。

コマンドライン・アップグレードでは、DBUA でのアップグレードと同じステップが実行され、同じ時間がかかります。コマンドライン・アップグレードは、より直接的に制御することを望んでいるデータベース管理者 (DBA) や、アップグレードと同時に新しいハードウェア・サーバーに移動する状況で、もっともよく使用されています。

Oracle Database 12c 以降では、アップグレードの成功を妨げる一般的な問題に対処するように、アップグレード前情報ツール (`preupgrd.sql`) によって修正スクリプトが自動的に生成されます。アップグレード後のフェーズも、数多くのアップグレード後ステップの実行を自動化するように強化されています。

コマンドライン・アップグレードの使用例

コマンドライン・アップグレードを使用して Oracle データベースをアップグレードするプロセスは、3 つのフェーズに分けられ、適切にアップグレードするために実行する少数のステップが各フェーズに含まれます。Oracle Database 12c 以降では、その後で CDB にデータベースをプラグインできます。

» アップグレード前のフェーズ

1. 新しいアップグレード前の情報ツール (`preupgrd.sql`) を実行して、アップグレードするデータベースの準備状況を確認する

2. `preupgrade_fixups.sql` スクリプトを実行して、アップグレード前の情報ツールで検出された問題に自動的に対処する
3. アップグレード前の情報ツールで特定された、手動の修正ステップを実行する

» アップグレード・フェーズ

1. パラレル・アップグレード・ユーティリティ (`catct1.pl`) を実行する

» アップグレード後のフェーズ

1. `postupgrade_fixups.sql` スクリプトを実行して、アップグレード前の情報ツールで特定された、アップグレード後に対処する必要がある問題を自動的に修正する
2. パラレル・アップグレード・ユーティリティで生成されたログ・ファイルを確認する
3. `utlrlp.sql` を実行して、無効なオブジェクトを再コンパイルする

» CDB へのプラグイン

1. `DBMS_PDB.DESCRIBE()` プロシージャを使用して、データベースに XML ディスクリプタ・ファイルを生成する
2. CDB に接続して `CREATE PLUGGABLE DATABASE` コマンドを実行し、非 CDB データベースを PDB に移行する
3. `noncdb_to_pdb.sql` スクリプトを実行し、非 CDB を使用可能な PDB に変換する

ソース・データベースに暗号化された表領域が含まれている場合や暗号化された列を持つ表が含まれている場合は、Oracle Database 12c の新しい SQL 文である `ADMINISTER KEY MANAGEMENT` の `export` ファンクションと `import` ファンクションを使用して、これらの表領域または表に関連付けられたキーをソース・データベースから新しい PDB に移行する必要があります。

これらのステップが、コマンドライン・アップグレード・プロセスの概要です。コマンドライン・アップグレードの詳細については、[Oracle Database アップグレード・ガイド](#)を参照してください。

コマンドライン・アップグレードの使用に関する考慮事項

多くの DBA は、DBUA などの GUI を使用するよりも、コマンドラインでコマンドを入力して得られる制御レベルを好みます。個人の好みを除外すると、コマンドライン・アップグレードの一般的なユースケースは、同じオペレーティング・システム・アーキテクチャに留まりながら、新しいハードウェアへと移行する場合です。

コマンドライン・アップグレードを使用すると、ユーザーはデータベース・アップグレードに使用されるパラレル処理の数を指定できるため、アップグレード速度とシステム・リソース使用率のバランスを取ることができます。

Oracle Database 12c Release 2 のコマンドライン・アップグレードからは、'-R'パラメータを使用して、中断されたポイントを自動的に検出し、アップグレードを再開できるようになりました。何らかの理由でコマンドライン・アップグレードが中断された場合は、[Oracle Database アップグレード・ガイド](#)の手順に従ってパラレル・アップグレード・ユーティリティを再実行または再開できます。

非 CDB データベースの CDB へのプラグイン

Oracle Database 12c にアップグレードしたら、データベースを CDB にプラグインできます。ソース・データベースとターゲットの CDB は次の要件を満たしている必要があります。

- » ハードウェアとオペレーティング・システム・プラットフォームでエンディアンが同じである
- » 互換性のあるデータベース・オプションがインストールされている
- » 互換性のあるキャラクタ・セットと各国語キャラクタ・セットがインストールされている。つまり、次のいずれかに該当する
 - » CDB のキャラクタ・セットが AL32UTF8 である
 - » 非 CDB のキャラクタ・セットが CDB のキャラクタ・セットと同じである
 - » 非 CDB のキャラクタ・セットが CDB のキャラクタ・セットのバイナリ・サブセットである

非 CDB から PDB を作成する手順については、[Oracle Database 管理者ガイド](#)を参照してください。オラクルでは、CDB に AL32UTF8 キャラクタ・セットを使用することを推奨しています。

複数の PDB を伴う CDB のアップグレード

Oracle Database 12c のコンテナ・データベースをアップグレードする場合、柔軟な CDB のアーキテクチャを活用し、次の 2 つのうちのいずれかの方法でアップグレードを実行できます。

- » CDB とすべての PDB を単一コマンドでアップグレードする
- » PDB または PDB のサブセットを一つずつアップグレードする
 1. 新規バージョンの Oracle Database ソフトウェアを使用して、新たな CDB を作成する
 2. ソース CDB から 1 つまたは複数の PDB をアンプラグし、新しい CDB にプラグインする
 3. DBUA またはコマンドライン・アップグレードのいずれかを使用して、古い PDB をアップグレードする

それぞれの手法には、トレードオフがあります。CDB のアップグレード方法を選択する際のいくつかの考慮事項は以下のとおりです。

“同時”アップグレード

- » 少ない労力：最大で 252 の PDB を単一コマンドでアップグレードできる
- » 低い柔軟性：すべてのアプリケーション PDB で共通の停止時間が必要となる
- » アップグレードの自動化により、アップグレード・プロセスの全体的な停止時間が削減される
- » 個々の PDB の停止時間が長くなる
- » CDB で元のデータベース SID が維持される

アンプラグ/プラグイン/アップグレード

- » 多くの労力：個々の PDB または PDB のセットをアンプラグ、プラグイン、およびアップグレードする必要がある
- » 高い柔軟性：ユーザーのニーズに応じて移行期間を計画できる
- » 新規サーバーへの移行が容易である
- » 個々の PDB の停止時間が短くなる
- » 2 つ目の CDB を使用する（同一サーバー上にある場合はリソースの割当てが必要な場合がある）

CDB のアップグレード方法は、自身のユースケースと、アップグレードまたは移行シナリオに基づき選択します。また、複数の方法を組み合わせることもできます。たとえば、残りの PDB をソース・バージョンに残しながら、PDB を一つアンプラグし、新しい CDB にプラグインしてアップグレードできます。次に、元の CDB を同時アップグレードし、最初の（アップグレード済みの）PDB を、アンプラグ/プラグインを使用して元の CDB に再び戻すことができます。

方法2：フル・トランスポートابل・エクスポート/インポート またはトランスポートابل表領域

トランスポートابل表領域を使用すると、データベースから別のデータベースに一連の表領域をコピーできます。この方法では、これらの表領域からデータをエクスポートしてインポートするよりも、かなり短時間に行うことができます。表領域は物理ファイルとしてコピーされ、これらのファイルに含まれている行や索引などの論理エンティティを解釈する必要がないためです。表領域のコピーに加えて、Data Pump エクスポート/インポートにより、ソース・データベース内のオブジェクトを記述したメタデータを新しいデータベースに移動する必要があります。

トランスポートした表領域を、異なるオペレーティング・システム・プラットフォームで実行されている別のデータベースや、異なるバージョンの Oracle Database ソフトウェアが実行されている別のデータベースにコピーできます。このため、トランスポートابل表領域を使用すると、1 回の操作で、データベースを比較的短時間で移行およびアップグレードできます。このスピードのトレードオフとして、トランスポートابل表領域では、プロシージャ、パッケージ、制約などのメタデータをユーザー自身で移動する必要があるため、かなり複雑になる可能性があります。Oracle Database 12c 以降では、新しいフル・トランスポートابل・エクスポート/インポート機能により、トランスポートابل表領域のスピードの利点が得られるだけでなく、インストールされているデータベース・オプションのメタデータとデータをトランスポートするプロセスが大幅に簡単になっています。

フル・トランスポートابل・エクスポート/インポート

フル・トランスポートابل・エクスポート/インポートは Oracle Database 12c の新機能であり、この機能により、トランスポートابل表領域機能を使用してデータベース全体を容易に移動できます。メタデータを移動するプロセスを自動化するほか、SYSAUX や SYSTEM などの非トランスポートابلの表領域に配置されているデータを移動できます。さらに、暗号化された表領域もトランスポートできます。

フル・トランスポートابل・エクスポート/インポートでは、ダンプ・ファイルを使用するか、またはデータベース・リンク経由で、メタデータを移動します。このスピードと簡易性の組合せにより、フル・トランスポートابل・エクスポート/インポートは多くの移行シナリオに適した選択肢となっています。フル・トランスポートابل・エクスポート/インポートを使用する上での詳細なステップについては、Oracle ホワイト・ペーパー、[Oracle Database 12c: フル・トランスポートابل・エクスポート/インポート](#)を参照してください。

「Oracle Database 12c は、フル・トランスポートابل・エクスポート/インポートを使用してデータベース全体をインスタンスから別のインスタンスにコピーできるようになりました。これにより、顧客データベースの移行時に、停止時間が最小限に抑えられています」

Accenture、データベース管理部門責任者、Julian Dontcheff 氏

フル・トランスポート/インポートを使用した、プラグブル・データベースへの移行
フル・トランスポート/インポートを使用すると、新しいオペレーティング・システム・プラットフォームおよび Oracle Database の新しいリリースに 1 回の操作でデータベース全体を移行できるため、この方法は PDB への移行に非常に役立ちます。フル・トランスポート/インポートを使用して PDB に移行する手順は、非 CDB に移行する手順と同じです。

フル・トランスポート/インポートの使用例

フル・トランスポート/インポートの利点として、暗号化された表領域や暗号化された列が含まれた表を新しいデータベースにトランスポートできることが挙げられます。

フル・トランスポート/インポート機能は、Oracle Database 12c に実装済みです。また、この機能のエクスポート側のサポートは、Oracle Database 11g Release 2 (11.2.0.3) 以降で提供されています。

これらの例では、次のユーザー表領域が含まれた 11.2.0.3 のソース・データベースを使用していると想定しています。

表3：ソース・データベースの表領域

表領域名	暗号化の有無	データファイル名
HR	あり	/data3/oracle/dbs/hr_1.f
ENGTABLES	なし	/data4/oracle/dbs/eng_1.f
ENGINEXES	なし	/data4/oracle/dbs/eng_2.f

これらの 3 つの表領域すべてと、プロシージャ、パッケージ、制約、トリガーなどのすべてのメタデータが、1 回の操作でソースからターゲットに移動されます。

最初の例は、ダンプ・ファイルを使用したフル・トランスポート/インポート操作です。この場合、ソース・データベースのメタデータがダンプ・ファイルにエクスポートされ、ダンプ・ファイルと表領域のデータファイルの両方が新しいシステムに転送されます。手順は次のとおりです。

1. ソース・データベースのユーザー表領域を読み取り専用を設定します。
2. Oracle Database 11g Release 2 (11.2.0.3) 環境から、FULL=Y パラメータと TRANSPORTABLE=ALWAYS パラメータを使用して、ソース・データベースの管理表領域内に配置されているメタデータとすべてのデータをエクスポートします。VERSION=12 パラメータは、Oracle Database 11g Release 2 データベースからエクスポートする場合のみ必要です。

```
expdp src112admin/<passwd>@src112 DIRECTORY=src112_dir
DUMPFIL=src112fulltts.dmp VERSION=12 FULL=Y
TRANSPORTABLE=ALWAYS
EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
ENCRYPTION_PASSWORD=<enc_passwd> METRICS=Y
LOGFILE=src112fulltsexp.log
```

3. ソース・システムの表領域のデータファイルをターゲット・システムにコピーします。エクスポート操作のログ・ファイルに、移動する必要があるデータファイルが一覧表示されます。

4. ソース・データベースのインポート先となる PDB を含め、ターゲット・システムに CDB を作成します。PDB の管理については、[Oracle Database 管理者ガイド](#)を参照してください。
5. Oracle Database 12c 環境で、事前に作成した PDB に接続し、ダンプ・ファイルをインポートします。ダンプ・ファイルのインポート操作により、表領域のデータファイルがターゲットの PDB にプラグインされます。

```
impdp pdbadmin/<passwd>@tgtpdb DIRECTORY=src112_dir
      DUMPFILE=src112fulltts.dmp
      ENCRYPTION_PASSWORD=<enc_passwd>
      METRICS=Y LOGFILE=src112fullttsimp.log
      TRANSPORT_DATAFILES='/recovery1/data/hr_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_2.f '
```

フル・トランスポートابل・エクスポート/インポートの2つ目の使用例では、ネットワーク・モードの Data Pump を使用し、ダンプ・ファイルは使用しません。この場合、表領域のデータファイルが、ソース・システムとターゲット・システムの両方からアクセスできる Storage Area Network (SAN) デバイスなどの場所に配置されていると想定しています。これにより、1 回の Data Pump コマンドで非 CDB から PDB に移行できます。

1. ソース・データベースのインポート先となる PDB を含め、ターゲット・システムに CDB を作成します。
2. ターゲットの PDB に、ソース・データベースを指すデータベース・リンクを作成します。
3. ソース・データベースのユーザー表領域を読み取り専用を設定します。
4. Oracle Database 12c 環境で、フル・トランスポートابل・エクスポート/インポートをネットワーク・モードで使用して、ソース・データベースからターゲットの PDB に直接インポートします。

```
impdp pdbadmin/<passwd>@tgtpdb NETWORK_LINK=src112 VERSION=12
      FULL=Y TRANSPORTABLE=ALWAYS
      EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
      ENCRYPTION_PASSWORD=<enc_passwd>
      METRICS=Y LOGFILE=tgtpdb_dir:src112fullimp.log
      TRANSPORT_DATAFILES='/recovery1/data/hr_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_1.f '
      TRANSPORT_DATAFILES='/recovery1/data/eng_2.f '
```

従来のダンプ・ファイルまたはネットワーク・モードのいずれを使用する場合でも、フル・トランスポートابل・エクスポート/インポートでは、新しいバージョンへのデータベースのアップグレード、異なるオペレーティング・システムやハードウェア・プラットフォームへの移行、PDB への移行を 1 回の操作で実行でき、これら 3 つのアップグレードと移行すべても 1 回の操作で実行できるため、便利です。

フル・トランスポートابل・エクスポート/インポートの使用に関する考慮事項

フル・トランスポートابل・エクスポート/インポートには、『Oracle Database 管理者ガイド』に示されている制限事項が適用されます。

フル・トランスポートابل・エクスポート/インポートは、Oracle Database 11g Release 2 (11.2.0.3) 以降のソース・データベースの移行に使用できます。

フル・トランスポートابل・エクスポート/インポート・ジョブは再開できません。操作が中断された場合、ジョブ全体を最初から開始する必要があります。

PDB へのネットワーク・モードの移行の詳細を含めた、フル・トランスポートابل・エクスポート/インポートの詳細については、『Oracle Database ユーティリティ』のガイドを参照してください。

ソース・データベースとターゲット・データベースで、ハードウェア・プラットフォームとオペレーティング・システム・プラットフォームのエンディアン特性が異なる場合、Oracle RMAN の CONVERT コマンドを使用して各表領域を新しいプラットフォーム向けに変換する必要があります。Oracle RMAN の CONVERT コマンドの詳細については、『Oracle Database バックアップおよびリカバリ・リファレンス』を参照してください。

トランスポートابل表領域

トランスポートابل表領域機能を使用すると、1 つの表領域または一連の表領域を新しいデータベースにコピーできます。このデータ移行方法は高速で信頼性が高いですが、一部の DBA にとっては好ましくない、数多くの複雑な手順が必要になります。

トランスポートابل表領域を使用した、プラグブル・データベースへの移行

フル・トランスポートابل・エクスポート/インポートと同様に、トランスポートابل表領域は、PDB に移行するために使用できます。トランスポートابل表領域を使用してデータベースを PDB に移行する手順は、非 CDB へのトランスポートابل表領域の移行と同じです。

トランスポートابل表領域の使用例

Oracle Database 11g Release 2 (11.2.0.3) 以降のデータベースを移行するには、フル・トランスポートابل・エクスポート/インポートを使用することを推奨します。引き続き、トランスポートابل表領域機能を使用して以前のリリースのデータベースを移行することもできます。トランスポートابل表領域機能の使用の詳細については、次のホワイト・ペーパーを参照してください。

- » [トランスポートابل表領域を使用したデータベースのアップグレード](#)
- » [トランスポートابل表領域を使用したプラットフォームの移行](#)

この機能の使用手順は、Oracle Database 12c でも変更ありません。

トランスポートابل表領域の使用に関する考慮事項

データベース間での表領域のトランスポートには、[Oracle Database 管理者ガイド](#)に示されている制限事項が適用されます。

トランスポートابل表領域のエクスポート/インポート・ジョブは再開できません。操作が中断された場合、ジョブ全体を最初から開始する必要があります。

トランスポートابل表領域は、Oracle8i Database 以降のデータベースで使用できます。この機能で、Oracle Database 10g (10.1.0.3) 以降のクロス・プラットフォームの移行を実行できます。

ソース・データベースとターゲット・データベースで、ハードウェア・プラットフォームとオペレーティング・システム・プラットフォームのエンディアン特性が異なる場合、Oracle RMAN の CONVERT コマンドを使用して各表領域を新しいプラットフォーム向けに変換する必要があります。Oracle RMAN の CONVERT コマンドの詳細については、[Oracle Database バックアップおよびリカバリ・リファレンス](#)を参照してください。

方法3：Oracle Data Pumpエクスポート/インポート

Oracle Data Pump では、Oracle データベース内および Oracle データベース間でデータとメタデータを高速移動できます。Oracle Data Pump のエクスポート・ユーティリティ (expdp) とインポート・ユーティリティ (impdp) は、柔軟性と操作性が非常に優れていることから、新しいハードウェア・サーバー、異なるオペレーティング・システム・プラットフォーム、および新しいリリースの Oracle Database ソフトウェアに表、スキーマ、データベースを移行するために広く使用されています。

Oracle Data Pump では、ディスク上のダンプ・ファイルにデータを書き込むことも、ソース・データベースからターゲットにネットワーク経由でデータを直接転送することも可能です。データをインポートするときに、ターゲット・データベースの特性に合わせてデータを変換できます。インポート時にデータベースを変換する方法には、新しいキャラクタ・セットへの移行、暗号化や圧縮の実装、SecureFiles LOB への BasicFiles LOB の変更、データベース内の表のパーティション化の変更など、役立つ選択肢があります。

Oracle Data Pump は、コマンドライン・インタフェースのフル・トランスポートブル・エクスポート/インポート機能でもあります。フル・トランスポートブル・エクスポート/インポートでは、Data Pump エクスポートよりもデータを高速に移動できますが、Data Pump エクスポートは制限事項が少なく、データ・レイアウトとデータ形式をさらに柔軟に処理できます。

ダンプ・ファイルを使用した、Data Pumpエクスポート/インポート

長年の間、ダンプ・ファイルにデータをエクスポートするのが、データベース間でデータを移動するもっとも一般的な方法でした。Data Pump エクスポートでは、ソース・データベースからフラット・ファイルにデータを抽出するのに最善の方法が選択され、Data Pump インポートでは、ダンプ・ファイルからデータを読み取ってターゲット・データベースに挿入するのに、同じように最善の方法が選択されます。

ダンプ・ファイルにエクスポートする最大の利点は、ディスク上にデータの永続的なコピーが保持されることです。これにより、ダンプ・ファイルを複数のインポートで再利用できるため、インポートのテストやソース・データベースのチューニングが必要な場合に特に役立ちます。また、ダンプ・ファイルにエクスポートすると有効なケースとして、ソースとターゲット間のネットワークが比較的低速な場合があります。このような場合、ディスクをソース・システムから物理的に切断してターゲットに接続すると、ネットワーク経由でデータをコピーする場合と比較し、さらに高速になる可能性があります。

Oracle Data Pumpネットワーク・モード

ソース・データベースをダンプ・ファイルにエクスポートしてこの同じダンプ・ファイルからターゲット・データベースにインポートする代わりに、Data Pump インポートを使用してデータベースをネットワーク・リンク経由で移行することを選択できます。これにより、ダンプ・ファイルを保存、管理、および転送する必要がなくなります。代わりに、データがソース・データベースから抽出されて、データベース・リンク経由でターゲット・データベースに直接挿入されます。

ファイルの保存と管理の必要性が軽減されるだけでなく、ネットワーク・モードにより、Oracle の新しいリリースへのクロス・プラットフォームの移行を 1 つの手順で実行できるため、移行が簡素化されます。Data Pump ネットワーク・モード・ジョブは、常に再開可能です。

Oracle Data Pump を使用した、プラグブル・データベースへの移行

Oracle Data Pump エクスポート/インポートを使用すると、データベースを PDB に柔軟かつ容易に移行できます。異なるオペレーティング・システムにソース・データベースが配置されている場合、ソース・データベースで異なるキャラクタ・セットを使用している場合、および移行プロセス中にソース・データベースの変更や再構成が必要になる場合に、特に役立ちます。

Oracle Data Pump の使用例

Oracle Data Pump を使用してデータベースをアップグレードまたは移行する手順は、以前のリリースと同じです。Oracle Data Pump で従来のダンプ・ファイルを使用して移行する場合、手順は次のようになります。

1. ソース・データベースをダンプ・ファイルにエクスポートします。

```
expdp srcadmin/<passwd> DIRECTORY=src_dir
      DUMPFILE=srcfull.dmp LOGFILE=srcfullexp.log
      FULL=Y METRICS=Y PARALLEL=4
      EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
```

2. ダンプ・ファイルをターゲット・システムにコピーするか、ネットワークにマウントされたディスク上に配置して、ターゲット・システムでダンプ・ファイルを使用できるようにします。
3. ソース・データベースのインポート先となる PDB を含め、ターゲット・システムに CDB を作成します。
4. ターゲットの PDB で、インポートするダンプ・ファイルのディレクトリ・オブジェクトを作成します。
5. Data Pump のダンプ・ファイルをターゲットの PDB にインポートします。

```
impdp pdbadmin/<passwd>@pdb DIRECTORY=src_dir
      DUMPFILE=srcfull.dmp LOGFILE=srcfullimp.log
      PARALLEL=4 METRICS=Y
```

この例では、パフォーマンスを高めるために、PARALLEL パラメータなどの Oracle Data Pump 機能を利用しています。ネットワーク・モードを使用して、ダンプ・ファイルを使わないこともできます。Oracle Data Pump の詳細については、[Oracle Database ユーティリティ](#)を参照してください。

Oracle Database 12c Release 2 では、Data Pump に、メタデータを並列にエクスポートおよびインポートできる機能が追加されました。この機能により、メタデータの多いデータベースの移行時間が短縮されます。

Oracle Data Pump の使用に関する考慮事項

Oracle Data Pump は、Oracle Database 10g 以降で使用できます。

Oracle Data Pump では、SYS スキーマからオブジェクト (GRANT オブジェクトを含む) をエクスポートしません。SYS 所有のオブジェクトのユーザー GRANT など、SYS スキーマにユーザー・オブジェクトがある場合、それらのオブジェクトはターゲット・データベースで別個に再作成する必要があります。

Oracle Data Pump ネットワーク・モードには、Oracle Database ユーティリティに示されているネットワーク・リンクの制限事項が適用されます。

Oracle Database 11g Release 2 (11.2.0.3) 以降では、ダンプ・ファイルへのエクスポート時にパラメータ `VERSION=12` を設定できます。これにより、登録されているデータベース・オプションとコンポーネントのすべてのデータがエクスポートに含まれるように指定されます。設定 `VERSION=12` を使用して生成されたダンプ・ファイルは、Oracle Database 12c 以降でインポートできます。

方法4：オリジナルのエクスポート/インポート

オラクルでは基本的に、Data Pump のエクスポート・ユーティリティとインポート・ユーティリティを使用して、Oracle データベース間でデータを移動することを推奨しています。ただし、古いデータベースを Oracle Database 12c にアップグレードまたは移行するときに、オリジナルのエクスポート (`exp`) とインポート (`imp`) が役立つ場合があります。たとえば、Oracle9i Database から異なるオペレーティング・システム・プラットフォーム上の Oracle Database 12c に移行する必要があります。Oracle Data Pump は Oracle Database 10g 以降でサポートされるため、この操作にはオリジナルのエクスポートを使用します。

オリジナルのエクスポート/インポートを使用した、プラグブル・データベースへの移行

オリジナルのインポート・ユーティリティは、特に古いエクスポート・ダンプ・ファイルを新しいバージョンの Oracle Database に移行する場合に、引き続き完全にサポートされています。オリジナルのエクスポート・ユーティリティを使用して古いデータベースをエクスポートしてから、オリジナルのインポート・ユーティリティを使用してそのデータベースを PDB にインポートできます。これにより、古いプラットフォームのデータベースを CDB に統合できます。

オリジナルのエクスポート/インポートの使用例

オリジナルのエクスポート/インポートを使用して、Oracle9i Database 以前のリリースを Oracle Database 12c に移行できます。このような移行の手順は、次のとおりです。

1. ソース・データベースをダンプ・ファイルにエクスポートします。

```
exp src92admin/<passwd> full=y file=exp9i.dmp
```

2. ダンプ・ファイルを新しいシステムにコピーするか、ネットワークにマウントされたディスクに配置して、ターゲット・システムでダンプ・ファイルを使用できるようにします。
3. ソース・データベースのインポート先となる PDB を含め、ターゲット・システムに CDB を作成します。
4. オリジナルのエクスポート・ダンプ・ファイルをターゲットの PDB にインポートします。

```
imp pdbname/<passwd> full=y file=exp9i.dmp
```

オリジナルのエクスポート/インポートを使用して CDB にアップグレードおよび移行する手順は、Data Pump エクスポート/インポートを使用する手順と概念的に似ていますが、オリジナルのエクスポート/インポートと Data Pump エクスポート/インポート・ユーティリティには、使用上の違いがいくつかあります。

- » Data Pump エクスポート/インポートは通常、オリジナルのエクスポート/インポートよりもはるかに高速であり、特に大容量のデータの場合に高速です。
- » Data Pump エクスポート/インポートでは、データベースのすべてのデータ型と機能がサポートされます。オリジナルのエクスポート/インポートでは、Oracle Database 10g Release 2 以降のリリースで追加されたデータ型や機能のエクスポートはサポートされません。
- » オリジナルのエクスポート/インポートでは、データベースにインストールされているオプションのデータを移行できないことがあります。

そのため、Oracle9i Database 以前のデータベースを移行する場合を除き、オリジナルのエクスポート/インポートではなく、Oracle Data Pump エクスポート/インポートを使用することを推奨します。

オリジナルのエクスポート/インポートの使用に関する考慮事項

Oracle Database 11g から、オリジナルのエクスポートの一般的な使用に関するサポートは終了しています。この方法は、11g 以前のデータベースを移行する場合に限る必要があります。

オリジナルのエクスポート/インポートを使用してデータベースを移行すると、特に 100GB を超えるデータがある場合は非常に時間がかかる可能性があります。

停止時間を最小限に抑えたデータベース・アップグレード

多くのお客様が、特にデータベースのアップグレードなどの計画的なメンテナンス時に、データベースの可用性を最大限に高めることに関心を持っています。個々のアップグレード・シナリオに適用可能なもっとも迅速なアップグレードまたは移行方法を選択することに加えて、以下の手法を組み合わせることで、停止時間をさらに短縮するとともに、システムの可用性を最大限に高めることができます。

Oracle Data Guardによるローリング・アップグレード

Oracle Data Guard は、1 つ以上のスタンバイ・データベースを作成、維持、管理、および監視する包括的な一連のサービスを提供し、本番環境の Oracle データベースを障害やデータ破損から保護します。この機能は、おもにディザスタ・リカバリ・ソリューションとして数多く採用されていますが、Oracle Database のパッチ・セットとフル・リリースのローリング・アップグレードを行うのにも役立ちます。Oracle Data Guard を使用してデータベース・アップグレードに関連する停止時間を最小限に抑えるには、2 つの方法があります。

Oracle Database 10g (10.1.0.3) 以降では、ロジカル・スタンバイ・データベースを使用してローリング・アップグレードを実行できます。ローリング・アップグレード中に、1 度に 1 つずつアップグレードしながら、プライマリとロジカル・スタンバイ・データベースで異なるリリースの Oracle データベースを実行できるため、プライマリ・データベースの停止時間が最小限に抑えられます。SQL Apply を使用してデータベースのローリング・アップグレードを実行するプロセスについては、[Oracle Data Guard 概要および管理](#)を参照してください。

Oracle Database 11g 以降では、フィジカル・スタンバイ・データベースをローリング・アップグレードの出発点として使用してアップグレードを実行できます。フィジカル・スタンバイを使用してデータベース・アップグレードの停止時間を最小限に抑えるプロセスは、一時ロジカル・スタンバイと呼ばれます。詳細については、ホワイト・ペーパー、[フィジカル・スタンバイ・データベースの使用によるデータベース・ローリング・アップグレードの簡略化](#)を参照してください。一時ロジカル・スタンバイは、Oracle Database 11g以降で使用できます。

一時ロジカル・スタンバイは、データベースを新しいハードウェア・システムに移行する場合にも、停止時間を低減するのに非常に役立ちます。同じオペレーティング・システム・プラットフォームを引き続き使用する場合や、サポートされるプラットフォームの間（My Oracle Support Note [413484.1](#)を参照）で移行する場合は、一時ロジカル・スタンバイにより、データベースのアップグレードや移行を含め、計画的なメンテナンス時に、システムの可用性を最大限に高めることができます。

Data Guard のローリング・アップグレードを使用して Oracle Database 12c Release 2 に移行するには、ソース・データベースのバージョンが Oracle Database 11g Release 2 (11.2.0.3) 以降でなければなりません。

Oracle RMANのクロス・プラットフォーム増分バックアップ

Recovery Manager は、データベースでバックアップ・タスクとリカバリ・タスクを実行し、バックアップ戦略に基づく管理作業を自動化する、Oracle Database クライアントです。Recovery Manager を使用すると、データベース・ファイルのバックアップ、リストア、およびリカバリが大幅に簡素化されます。また、データベース移行による停止時間を最小限に抑えるのに役立つこともあります。

非常に大規模なデータベースを新しいハードウェア・システムに移行する場合、通常、データを物理的に移動するのに要する時間が、移行のもっとも時間のかかる部分になります。これは、特にフル・トランスポータブル・エクスポート/インポートまたはトランスポータブル表領域機能を使用する場合に当てはまります。データのエクスポートとインポートを Data Pump で実行しない場合でも、表領域のデータファイルをソース・システムからターゲット・システムにネットワーク経由で移動する必要があります。数テラバイトのデータを単にコピーするのに数時間かかる可能性があるため、コピーを実行している間は、本番データベースを読取り専用にする必要があります。

Oracle Database 11g Release 2 (11.2.0.4) 以降への移行では、Recovery Manager のクロス・プラットフォーム増分バックアップ機能を使用することで、サポートされる Linux プラットフォームにトランスポータブル表領域を移行する際の停止時間を短縮できます。クロス・プラットフォーム増分バックアップを使用するプロセスについては、My Oracle Support Note [1389592.1](#)を参照してください。これを使用すると、データベース全体を移行する場合や、1 つ以上の個々の表領域のセットを移動する場合に、停止時間を短縮できます。



Oracle GoldenGate

Oracle GoldenGate は、異種データのリアルタイムのレプリケーションと変換に対応した、高パフォーマンスのソフトウェア・アプリケーションです。Oracle GoldenGate をこのホワイト・ペーパーで説明している移行方法と組み合わせると、データベースのアップグレードと移行を最小限の停止時間で実行できます。Oracle GoldenGate は非常に柔軟であり、新しいオペレーティング・システム・プラットフォームへの移行、新しいリリースの Oracle Database へのアップグレード、新しいキャラクタ・セットへの移行、またはこれらの変更を組み合わせる場合に、停止時間を最小限に抑えることができます。データベースのアップグレードまたは移行プロジェクトにおける Oracle GoldenGate の使用については、ホワイト・ペーパー、[Zero-Downtime Database Upgrades Using Oracle GoldenGate](#) を参照してください。

まとめ

このホワイト・ペーパーでは、Oracle Database 12c Release 2 へのアップグレードと移行を支援するためにオラクルから提供されている、さまざまなツール、手法、およびユーティリティについての概要を説明しました。また、提供されているさまざまなアップグレードおよび移行方法を選択する上で役立つガイダンスと情報についても説明しました。個々の移行シナリオに使用するのに最適な方法は、ソース・データベースのバージョン、ソースとターゲットのオペレーティング・システム、停止時間要件、および DBA 個人の好みによって異なります。これらの要因に基づいて、個々の移行シナリオに最適な方法を利用できます。

Oracle Database 12c Release 2 へのアップグレードまたは移行により、数多くの利点が得られる可能性があります。新機能を利用することで、パフォーマンスを向上し、セキュリティを強化し、機能を拡張できます。IT インフラストラクチャを最新のものにできます。また、運用効率を高めることで、コストを削減し、生産性を向上させることができます。

データベースのアップグレードまたは移行で最小限の停止時間を達成する必要がある場合、まず、ソース・プラットフォームとターゲット・プラットフォームおよび Oracle Database リリースに基づいて、もっとも迅速なアップグレードまたは移行方法を選択します。その後、選択したアップグレードまたは移行方法を、最小限のダウンタイムを実現する機能および製品と組み合わせて、システムの可用性を最大限に高めることができます。

「当社の 12c へのアップグレード体験は極めて素晴らしいものであり、12c の堅牢性に大変感動しました。」
Neustar, Inc. の著名なエンジニア、Nick Salem 氏

付録A：アップグレードと移行に関するOracle Database 12c（12.1）の新機能

表4：アップグレードと移行に関するOracle Database 12cの新機能

機能	新機能
データベース・アップグレードの並列処理	Oracle Database 12c では、アップグレード・ユーティリティの <code>catupgrd.sql</code> が、新しい並列・アップグレード・ユーティリティの <code>catctl.pl</code> に置き換えられています。アップグレードのスクリプトとプロセスを並列で実行できるため、CPU 容量を最大限に活用でき、アップグレード時間を短縮することができます。データベースのアップグレードを並列・モードまたはシリアル・モードのいずれで実行するのか、選択できます。 Database Upgrade Assistant (DBUA) では、新しい並列・アップグレード・ユーティリティが活用されますが、ユーザーは、使用される並列度を制御できません。
単一コマンドを使用した CDB のアップグレード	単一のコマンドラインまたは DBUA アップグレードを使用して、単一の CDB と複数の PDB をアップグレードできます。これにより、複数のデータベースをアップグレードするのに必要な労力を大幅に削減できます。
アンプラグ、プラグイン、およびアップグレード	CDB 環境において、ある CDB から PDB をアンプラグし、新しいバージョンの CDB にプラグインし、PDB を新しいバージョンにアップグレードできます。これにより、異なるサーバーやコンテナ間で PDB を柔軟に移行したり、アップグレードしたりできます。
フル・トランスポート・エクスポート/インポート	Oracle Database 12c で導入されたこの機能は、Data Pump の使いやすさとトランスポート表領域の速度を組み合わせたものです。
Data Pump での CDB のサポート	Data Pump を使用すると、非 CDB から PDB に、同じ CDB 内または異なる CDB の PDB 間で、および PDB から非 CDB に、すべてのデータベースまたは一部のデータベースを移行できます。
Data Pump による、インポート中のすべての表の COMPRESSION または LOB_STORAGE の変更	インポート・ジョブ中に、マテリアライズド・ビューに記憶域を提供する表を含め、ジョブ内のすべての表の圧縮の種類を変更できるようになりました。これを行うには、TRANSFORM パラメータを使用して TABLE_COMPRESSION_CLAUSE を指定します。また、インポート・ジョブ中に、マテリアライズド・ビューに記憶域を提供する表を含め、ジョブ内のすべての表の LOB 記憶域 (SECUREFILE または BASIC FILE) を変更できるようになりました。これを行うには、TRANSFORM パラメータを使用して LOB_STORAGE オプションを指定します。
アップグレード時の Oracle XML Database のインストール	Oracle Database 12c より、Oracle XML Database (Oracle XML DB) は、Oracle Database の必須コンポーネントになりました。これをアンインストールすることはできないため、新しいデータベースの作成時にこれを含めないようにするオプションはありません。アップグレード中に Oracle XML DB が見つからない場合、Oracle XML DB は SYS_AUX 表領域に自動的にロードされます。アップグレードされた Oracle データベースには Oracle XML DB が含まれます。

付録B：アップグレードと移行に関する

Oracle Database 12c Release 2 (12.2) の新機能

表5：アップグレードと移行に関するOracle Database 12c Release 2の新機能

機能	新機能
新しいアップグレード前のスクリプト	アップグレード前の新しい情報ツールがこのリリース向けに作成され、preupgrd.sql スクリプトは preupgrade.jar ファイルに置き換えられています。この新しいユーティリティは、アップグレード前とアップグレード後の両方で発生する可能性がある問題に対処するための、修正スクリプトを提供します。Database Upgrade Assistant (DBUA) とアップグレード前の情報ツールは、アップグレードするデータベースの詳細なチェックを実行し、検出された問題の修正に特化した特別なスクリプトが生成されます。
再開ポイントの自動検出	アップグレードが中断された場合、'R' パラメータを使用してアップグレードを再開すると、別の実行でアップグレード処理を開始するための適切な場所が自動的に検出されます。
マルチメディアおよび XDB のアップグレードの平行処理	平行処理は、Oracle Database 12c で Core RDBMS サーバー・コンポーネントに追加されたのに加え、Oracle Database 12c Release 2 より、空間、マルチメディア、および XDB コンポーネントで使用されています。
単一コマンドを使用した CDB のアップグレード	単一のコマンドラインまたは DBUA アップグレードを使用して、単一の CDB と複数の PDB をアップグレードできます。これにより、複数のデータベースをアップグレードするのに必要な労力を大幅に削減できます。
Data Pump によるメタデータの平行・エクスポートおよびインポート	Oracle Database 12c Release 2 より、Data Pump でメタデータを平行にインポートできるようになりました。これにより、Data Pump での移行においてメタデータ処理の部分を大幅に加速させることができます。



Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口
電話：+1.650.506.7000
ファクシミリ：+1.650.506.7200

CONNECT WITH US



Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracle および Java は Oracle およびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Intel および Intel Xeon は Intel Corporation の商標または登録商標です。すべての SPARC 商標はライセンスに基づいて使用される SPARC International, Inc. の商標または登録商標です。AMD、Opteron、AMD ロゴおよび AMD Opteron ロゴは、Advanced Micro Devices の商標または登録商標です。UNIX は、The Open Group の登録商標です。0116

Oracle Database 12c Release 2 (12.2.0.1) へのアップグレードと移行

2016年11月

著者：Roy F. Swonger

共著者：Carol Tagliaferri、Mike Dietrich、Jim Stenoish



Oracle is committed to developing practices and products that help protect the environment