



# Oracle Databaseのアップグレードと移行



現在のOracleデータベースからOracle Databaseへのアップグレード  
または移行に利用可能な最適な方法の選び方と使い方

2021年1月 | バージョン3.00  
Copyright © 2021, Oracle and/or its affiliates

## 本書の目的

本書では、Oracle Databaseで利用可能なアップグレードおよび移行方法の概要を示します。本書は、御社がOracle Databaseへのアップグレードのビジネス上の利点を評価し、ITプロジェクトを計画するのを支援することのみを目的としています。

## 免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。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ください。本書に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。

製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

## 目次

本書の目的	1
免責事項	1
はじめに	3
アップグレードと移行の比較	3
データベースのアップグレード	3
データベースの移行	3
マルチテナント・アーキテクチャ	4
データベースのアップグレードまたは移行方法の選択	4
Oracle Database 19cへの直接アップグレード	4
アップグレード方法と移行方法	5
アップグレードおよび移行方法の詳細	7
方法1：Oracle AutoUpgrade Toolまたはコマンドラインによるアップグレード	7
AutoUpgrade	7
手動によるコマンドライン・アップグレード	8
非CDBデータベースのCDBへのプラグイン	9
1つのCDBと複数のPDBのアップグレード	9
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	10
フル・トランスポートابل・エクスポート/インポート	10
トランスポートابل表領域	12
方法3：Oracle Data Pumpエクスポート/インポート	13
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	13
Oracle Data Pumpネットワーク・モード	13
方法4：オリジナルのエクスポート/インポート	14
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ابل・データベースへの移行	14
結論	15



## はじめに

Oracle Database 19cは、新機能と強化機能を備えるとともに、長期サポートがコミットメントされており、既存のOracle データベースの魅力的なアップグレード・ターゲットとなっています。Oracle Database 19c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、全く異なるオペレーティング・システムへの移行などの取り組みの一環として行われることがあります。新バージョンのOracle Databaseへのアップグレードでは、Oracle Cloudに移行するケースがますます増えています。

アップグレードと移行のシナリオには多くの違いがあるため、オラクルでは、ご使用のデータベースをOracle Database 19cにアップグレードおよび移行するための方法をいくつか用意しています。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や、お客様に特有の要件を満たす最適な方法を選択する際に鍵となる考慮すべきおもな要素について学べます。

## アップグレードと移行の比較

用語である"アップグレード"と"移行"は、他の状況では同義語として使用されることが多いですが、Oracle Databaseで用いられている文脈の場合"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データベースへのデータの移動について説明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確保するように、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。

- システム表とビュー内の列の追加、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
- データベースの種類、ユーザー、ロール、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 사용되는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

用語である"移行"は、Oracleデータベースに適用できる、いくつかの異なった種類のものに当てはまります。データベース・バージョンに加えて、このような変更には、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（ハードウェアまたは仮想化環境）
- 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）

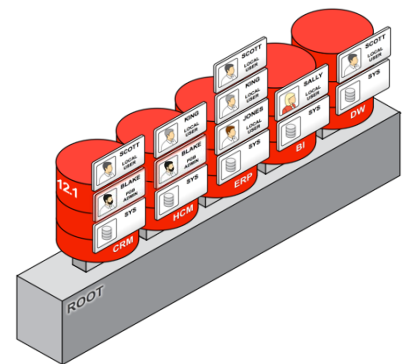
データベースの移行は、2つの重要な点でデータベースのアップグレードと異なります。1つ目は、データベースの移行には通常、データベース内のユーザー・データとアプリケーション・データの移動または変更が伴う点です。つまり、データベースのサイズがデータベース移行プロジェクトに大きく影響します。2つ目は、上記のいずれの移行も、データベース上で実行でき、新しいバージョンにアップグレードする必要がない点です。このホワイト・ペーパーでは、Oracle Database 19cへのアップグレードと移行の両方を同時に実行するための技法について説明します。

## マルチテナント・アーキテクチャ

6年以上前、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、およびスキーマではない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、お客様では複数のデータベースを容易に統合できます。また、このアーキテクチャでは、PDBへの移行という、お客様の検討対象となるもう1つの非常に重要なデータベース移行が導入されています。

ある場合には、従来のデータベース・アーキテクチャ（非CDBとも呼ばれます）からPDBへの移行を、データベースの他のアップグレード・タスクや移行タスクと組み合わせることで1回の操作で行うことができます。また別の場合には、PDBへの移行に追加の手順が伴うこともあります。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の方法の選択

提供されているいくつかのアップグレードおよび移行から、特定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、プロジェクトのいくつかの重要な特性を分析する必要があります。それぞれの特色は、Oracle Database 19cにアップグレードまたは移行する際の所定の方法の適性に影響します。

- 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。ただし、上記の特性に基づいて、あらゆるシナリオに最適な方法があります。

## Oracle Database 19cへの直接アップグレード

直接アップグレードでは、Oracle AutoUpgrade Tool（AutoUpgrade）を使用して、データベースをOracle Database 19c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。（注：コマンドライン・アップグレード（catctl.pl/dbupgrade）およびDatabase Upgrade Assistant（DBUA）は、従来のアップグレード手法としてサポートされます。）

表1：Oracle Database 19Cへの直接アップグレード・パス

ソース・リリース	ソースのパッチ・セット	AutoUpgrade Toolはサポートされていますか？
Oracle Database 12c	12.1.0.2	はい
	12.1.0.1	なし。別の方法を使用してください。
Oracle Database 11g Release 2	11.2.0.4	はい
	11.2.0.1、11.2.0.2、11.2.0.3	なし。別の方法を使用してください。
Oracle Database 11g Release 1以前	すべて	なし。別の方法を使用してください。

AutoUpgradeアップグレードがサポートされない場合、Oracle Database 19c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。中間的な手法を使用することもできます-11.02.0.4以降にアップグレードしてから、AutoUpgradeを使用して19cにアップグレードします。

## 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、いくつかの状況で使用できるバリエーションがあります。

これらの方法は次のとおりです。

1. Oracle AutoUpgrade Toolによるアップグレードまたはコマンドラインによる手動アップグレード
2. トランスポータブル表領域（TTS）のエクスポート/インポート -Oracle Databaseのフル・トランスポータブル・エクスポート/インポート機能、または従来のTTSモードを使用します
3. Oracle Data Pumpエクスポート/インポート -ダンプ・ファイルまたはネットワーク・モードを使用します
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	高速性	ソースの最小バージョン	新規サーバーへの移行	エンディアンネスの変更	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、およびアップグレード	中	最高速	12.1.0.2	はい	なし	なし
AutoUpgrade Tool	低	最高速	11.2.0.4	はい	なし	なし
コマンドライン・アップグレード (& DBUA)	中	最高速	11.2.0.4	なし	なし	なし
フル・トランスポートابل・エクスポート/インポート	中	より高速	11.2.0.4	はい	あり	なし
トランスポートابل表領域	高	より高速	8.1.5	はい	あり (10.1以降)	なし
Data Pump expdp/impdp	中	高速	10.1	はい	あり	あり
オリジナルのエクスポート/インポート	中	低速	5	はい	ありはい	あり

## アップグレードおよび移行方法の詳細

### 方法1：Oracle AutoUpgrade Toolまたはコマンドライン・アップグレード

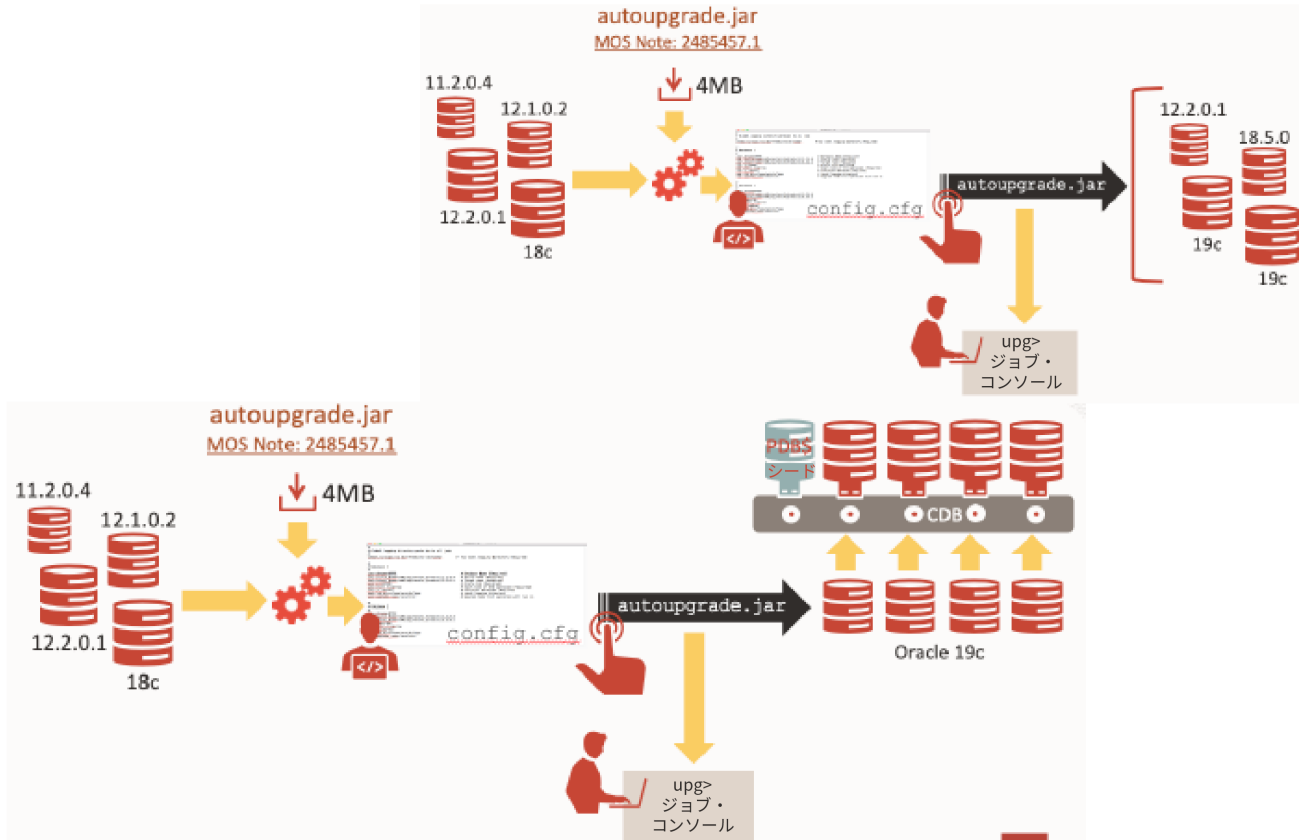
#### AutoUpgrade

Oracle Database 19cにアップグレードする場合にもっとも簡単に推奨されるのは、Oracle AutoUpgrade Toolを使用して既存のデータベースをアップグレードする方法です。この方法は、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推奨されており、「インプレース」のアップグレードとも呼ばれます。

[AutoUpgrade Tool](#)により、Oracle Databaseの11.2.0.4以降のリリースから19c以降へのアップグレード・プロセスが自動化されます。（リリース18.5および12.2もサポートされたターゲットです。）AutoUpgradeは、単一のコマンドおよび1つの構成ファイルを使用して、1つまたは複数のソース・データベースを分析して修正スクリプトを適用し、アップグレードを実行して、アップグレード後のチェックおよび構成の移行を完了します。AutoUpgradeは常に再開可能で、パラレル機能を使用します（catctl。）これはOracle Databaseサーバーを実行するすべてのオペレーティング・システムをサポートされ、つまりEnterprise EditionとStandard Edition 2、およびOracle Real Application Cluster（Oracle RAC）データベース・アップグレードが対象になります。

AutoUpgradeは、非CDBおよびCDBのアップグレードも、非CDBからPDBへの変換もサポートします。非CDBからPDBへのアップグレードの場合、事前作成されたCDBの場所が構成ファイルによって指定されます。AutoUpgradeによってPDBがプラグインされ、アップグレードと非CDBからPDBへの変換が1回で実行されます。





## AutoUpgradeの使用に関する考慮事項

同じ構成ファイルの異なるリリースから異なるターゲット・リリースへ同時にアップグレードできます。AutoUpgradeコンソールにより、アップグレード・プロセスを監視します。また、AutoUpgradeは、通常のデータベース・ロギングに加えて、アップグレード・プロセスの厳しいロギングを実行します。

Oracle Databaseの新しいリリース用のバイナリをダウンロードしたら、新バージョン用のOracleホームを設定してからAutoUpgradeを使用してください。また、AutoUpgradeを起動する前に、最新のRelease Update (RU) をOracleホームに適用することも推奨されています。詳しくは、[Oracle Databaseのアップグレード・クイック・スタート・ガイド](#)、Oracleドキュメント・ライブラリの最新の[Oracle Databaseアップグレード・ガイド](#)、および[My Oracle Support Note \(Doc ID 2485457.1\)](#)を参照してください。

## 手動によるコマンドライン・アップグレード

Oracle Database 19cでは、コマンドライン・アップグレード・ユーティリティ ([catctl.pl/dbupgrade](#)) を使用します。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、AutoUpgradeやDBUAと同様に、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同じ時間で実行されます。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、データベース・アップグレードと同時にデータベースを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。

正常なアップグレードを妨げる可能性がある一般的な問題に対処するため、アップグレード前情報ツール ([preupgrade.jar](#)) 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いくつかのステップがそれぞれのフェーズに存在します。Oracle Database 19c以降では、下記の状況で、データベースをCDBプラグイン可能です。

- アップグレード前のフェーズ

- 新しいアップグレード前の情報ツール (preupgrd.jar) 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
- 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
- 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
- パラレル・アップグレード・ユーティリティ (catctl.pl) を実行する
- アップグレード後のフェーズ
- 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
- 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
- utltp.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
- 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
- CDBに接続し、CREATE PLUGGABLE DATABASE コマンドを発行して非CDBデータベースをPDBに移行する
- noncdb\_to\_pdb.sql スクリプトを実行し、非CDBを完全に使用可能な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、または暗号化された列を持つ表が含まれている場合は、Oracle DatabaseのSQL文 ADMINISTER KEY MANAGEMENTのエクスポート機能とインポート機能を使用して、それらの表領域または表に関連付けられているキーをソース・データベースから新しいPDBに移行する必要があります。

以上の手順がコマンドライン・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。個人の好み以外での一般的なユースケースは、コマンドライン・アップグレード、新しいハードウェアに移行するが、同じオペレーティング・システム・アーキテクチャを引き続き使用するというものです。

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。

コマンドライン・アップグレードでは、中断されたポイントを自動的に検出し、'-R'パラメータを使用してアップグレードを再開できます。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、『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。

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

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。一般に、現行のハードウェア・システムでOracleデータベースをアップグレードする場合に使用できるのは、DBUAのみです。これは、DBUAにより、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が実行されたから、ターゲットのOracleホームに切り替わり、アップグレードとアップグレード後のステップが実行されるためです。

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での介入が必要な場合があります。

## 非CDBデータベースのCDBへのプラグイン

Oracle Database 19c以降にアップグレードしたら、CDBにデータベースをプラグインが可能です。ソース・データベースとターゲットのCDBが次の要件を満たしている必要があります。

- 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・プラットフォームでエンディアンが同じである
- PDBのコンポーネントがCDBで使用可能なコンポーネントのサブセットになっている
- 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。

## 1つのCDBと複数のPDBのアップグレード

Oracle Database 19c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を単一のコマンドでアップグレードする
- 一度に単一のPDBまたはPDBのサブセットをアップグレードする
- 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
- 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
- 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。

表3：CDBのアップグレード方法の選択

“全部同時”アップグレード	アンプラグ/プラグイン/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可能	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
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	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
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	新規サーバーへの移行が容易である
個々のPDBの停止時間が長くなる	個々のPDBの停止時間が短くなる
CDBで元のデータベースSIDが維持される	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が必要で、複数の方法を組み合わせることもできます。たとえば、単一のPDBを、残りのPDBをソース・バージョンに残しながら、アンプラグし、新しいCDBにプラグインし、アップグレードします。次に、元のCDBをすべて同時にアップグレードし、最初の（アップグレード済みの）PDBを、アンプラグとプラグインを使用して元のCDBに再び移動できます。

## 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

トランスポータブル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は、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、トランスポータブル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、トランスポータブル表領域では、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 19c以降では、フル・トランスポータブル・エクスポート/インポート機能により、トランスポータブル表領域のスピードと、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場合の大幅に簡単なプロセスが組み合わされています。詳しくは、Oracle Support Noteの『*Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup*』（Doc ID 2471245.1）を参照してください。

### フル・トランスポータブル・エクスポート/インポート

フル・トランスポータブル・エクスポート/インポートはOracle Databaseの機能で、これにより、トランスポータブル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、SYS\_AUXやSYSTEMなどの非トランスポータブル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・トランスポータブル・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・トランスポータブル・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・トランスポータブル・エクスポート/インポートの使用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: Full Transportable Export/Import』を参照してください。Oracle Database 19cでのフル・トランスポータブル・エクスポート/インポートの全体的なプロセスは変更されていません。

### フル・トランスポータブル・エクスポート/インポートを使用した、プラグラブル・データベースへの移行

フル・トランスポータブル・エクスポート/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・トランスポータブル・エクスポート/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ابل・エクスポート/インポートの使用例

フル・トランスポートابل・エクスポート/インポート機能の完全な実装は、Oracle Database 19cを含むOracle Database 12c以降のリリースに含まれています。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4.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
ENGINDEXES	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ابل・エクスポート/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(**READ ONLY**)に設定します。
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 DUMPFILE=src112fulltts.dmp VERSION=12 FULL=Y  
TRANSPORTABLE=ALWAYS EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS  
ENCRYPTION_PASSWORD=<enc_passwd> METRICS=Y  
LOGFILE=src112fullttsexp.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(**READ ONLY**)に設定します。
4. Oracle Database 19c環境で、フル・トランスポート/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/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しでも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/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/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/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/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/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/エクスポート/インポート表領域

トランスポート/エクスポート/インポート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/エクスポート/インポート表領域を使用した、プラガブル・データベースへの移行

フル・トランスポート/エクスポート/インポートと同様に、トランスポート/エクスポート/インポート表領域は、PDBに移行するために使用できます。トランスポート/エクスポート/インポート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/エクスポート/インポート表領域の移行と同じです。

### トランスポート/エクスポート/インポート表領域の使用例

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/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/エクスポート/インポート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/エクスポート/インポート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/エクスポート/インポート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/エクスポート/インポート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 19cでも変更ありません。

## トランスポート/エクスポート/インポート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。

トランスポータブル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。

トランスポータブル表領域は、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 RMANCONVERT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に変換する必要があります。Oracle RMANCONVERT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、BasicFiles LOB のSecureFiles LOBへの変更、データベース内の表のパーティション化の変更などの興味深い方法があります。

### 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、インポート先、ソース・データベースのテストやチューニングが必要な場合に特に役立ちます。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にも、ダンプ・ファイルにエクスポートすると役立ちます。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、プラガブル・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るはずで

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  
DUMPFIL=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・ファイルを不要にすることもできます。Oracle Data Pumpの詳細については、『[Oracle Databaseユーティリティ](#)』ガイドを参照してください。

---

Oracle Database 12c Release 2では、Data Pumpによって、メタデータを並列にエクスポートおよびインポートす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が加速されます。

---

## 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される必要があります。

Oracle Data Pumpネットワーク・モードには、『[Oracle Databaseユーティリティ](#)』ガイド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

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るとします。Oracle Data PumpはOracle Database 10g以降で使用できるため、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、プラガブル・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、廃止されたプラットフォームのデータベースもCDBに統合できます。

## 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。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にも、ダンプ・ファイルにエクスポートすると役立ちます。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、プラガブル・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・ファイルを不要にすることもできます。Oracle Data Pumpの詳細については、『Oracle Databaseユーティリティ』ガイドを参照してください。

---

Oracle Database 12c Release 2では、Data Pumpによって、メタデータを並列にエクスポートおよびインポートす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が加速されます。

---

## 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される必要があります。

Oracle Data Pumpネットワーク・モードには、『[Oracle Databaseユーティリティ](#)』ガイド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。

## 結論

このホワイト・ペーパーでは、Oracle Database 19c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から選択するのに役立つガイドライン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 19c以降にアップグレードまたは移行することには、組織にとって多くの利点があります。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。選択したアップグレードまたは移行方法を、最小限の停止時間を実現する機能および製品と組み合わせ、システムの可用性を最大限に高めることができます。

## CONNECT WITH US

+1.800.ORACLE1までご連絡いただくか、[oracle.com](http://oracle.com)をご覧ください。  
北米以外の地域では、[oracle.com/contact](http://oracle.com/contact)で最寄りの営業所をご確認いただけます。

 [blogs.oracle.com](http://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

 [twitter.com/oracle](https://twitter.com/oracle)

Copyright © 2021, 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、ここに記載されてい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。0120

Oracle Databaseのアップグレードと移行

2021年2月

著者：Roy Swonger

共著者：Mike Dietrich、Bill Beauregard

