



Oracle ホワイト・ペーパー
2014年1月

Oracle Database 12c : フル・トランスポート ブル・エクスポート/インポート

はじめに	3
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メリット	4
フル・トランスポータブル・エクスポート/インポートによる プラグブル・データベースのサポート	5
フル・トランスポータブル・エクスポート/インポートの内部詳細	5
フル・トランスポータブル・エクスポート/インポートの概要	6
フル・トランスポータブル・エクスポート	7
フル・トランスポータブル・インポート	7
エンディアン変換	8
例：フル・トランスポータブルを使用した非CDBからCDBへの移動	9
結論	12
付録：フル・トランスポータブル・エクスポート/インポートの制限事項	13

はじめに

Oracle Database 12cには、Oracle Multitenantオプションなどの重要な新機能が含まれています。このマルチテナント・アーキテクチャには、プラグブル・データベース (PDB) の作成機能が備わっています。これは、複数のデータベースを単一のマルチテナントのコンテナ・データベース (CDB) に統合する新しい方法です。多くの顧客は、プラグブル・データベースを使用した統合データベース環境に移行することで、データベース管理コストを削減し、ハードウェアの利用を効率化し、スケール・メリットを実現できることを待ち望んでいます。

データベースの移行に使用する従来の手法で、プラグブル・データベースをシームレスに利用できます。オリジナルのエキスポート/インポート、Data Pumpエキスポート/インポート、およびトランスポート/インポート領域はすべて、以前のバージョンのOracle Databaseから、Oracle Database 12cで稼働するプラグブル・データベースへの移行に使用できます。また、画期的な新機能であるフル・トランスポート/インポートを利用できます。この機能を使用すると、以前よりも迅速で、簡単かつ効率的にOracle Database 12cに移行できます。

このホワイト・ペーパーでは、Oracle Database 12cのフル・トランスポート/インポート機能について説明します。フル・トランスポート/インポートを使用することによるメリットの概要について紹介した後は、機能の動作について説明します。その後、フル・トランスポート/インポートの詳細な例を使用して、この機能を使用する構文とプロセス・フローを示します。このホワイト・ペーパーではプラグブル・データベース環境でのフル・トランスポート/インポートの使用に重点を置っていますが、この機能は一般的にOracle Database 12cの非CDBデータベースへの移行にも使用可能です。

フル・トランス
ポートは、Oracle
Database 11g
Release 2
(11.2.0.3) 以降で
利用できます。フ
ル・トランスポー
タブル・インポー
トは、Oracle
Database 12c以降
で利用できます。

フル・トランスポートابل・エクスポート/インポートを使用するメリット

フル・トランスポートابل・エクスポート/インポートは、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/インポートのユーザーになじみのある使いやすい操作性と、トランスポートابل表領域で実現可能な移行の速度を兼ね備えています。これら3つの移行手法の要約と、フル・トランスポートابل・エクスポート/インポートとの比較対照により、フル・トランスポートابل・エクスポート/インポートが従来の利用可能な移行手法よりも迅速で簡単かつ効率的である理由が分かります。

オリジナルのエクスポート/インポートは、一般的にもっとも時間のかかるデータベース移行の方法です。オリジナルのエクスポート/インポートは、多くのDBAの主流のユーティリティとして、何年にもわたって利用されてきましたが、大規模（100GB以上）データベースでの使用を念頭に置いて設計されておらず、現代のITシステムのデータ容量に対応できるようにパフォーマンスを拡張することができません。さらに、オリジナルのエクスポート/インポートには、Oracle Database 10g以降のリリースで追加された新しいデータベース機能をサポートするための機能強化が行われていません。

そのため、オリジナルのエクスポート/インポートは広く普及した信頼性の高い移行手法ですが、Oracle9i以前のデータベースのみで使用する必要があります。

Oracle Data Pumpは、Oracle Database 10gで導入されており、大量のデータを処理できるように設計されています。Oracle Data Pumpには、パラレル・ワーカー・プロセスなどの手法が採用されており、移動対象のデータに対する最適なアクセス方法の選択機能と、柔軟かつ強力な除外/追加機能が搭載されているため、Oracleデータベース間のデータ移動のもっとも一般的な方法として、事実上、オリジナルのエクスポート/インポートに取って代わりました。Oracle Data Pumpのコマンドラインは、オリジナルのエクスポート/インポートのコマンドラインと同一ではありませんが、DBAによく知られた特色を持ち、一般的には非常に使いやすいと考えられています。ただし、これらすべてのメリットがあっても、パフォーマンスの面ではData Pumpよりもトランスポートابل表領域が上回る場合があります。移行対象のデータが非常に大きい（数百ギガバイト以上）か、移行対象の索引が大量にある場合、Data Pumpも迅速に機能しますが、他の方法のほうがより迅速に機能する場合があります。

トランスポートابل表領域は、一般的にはデータベース間でユーザー・データおよびアプリケーション・データを移動する最速の方法です。この方法では、表領域のデータファイルがソース・データベースからターゲットにまとめて移動されるためです。通常、データファイル全体の移動は、個々の行のエクスポート/インポートはもちろん、データ・ブロックのエクスポート/インポートよりもはるかに高速です。ただし、従来のトランスポートابل表領域では、移行先データベースでこれらの表領域のデータファイルを効果的に使用するために必要なユーザーとアプリケーションのメタデータを移動するため、きわめて複雑な一連の手順を実行する必要がある場合があります。このように、非常に高速であるが他の手法よりも複雑であることが、トランスポートابل表領域を使用した移行の特徴となります。

Oracle Database 12cでは、速度とユーザビリティの面で最適な機能がフル・トランスポートابل・エクスポート/インポートに統合されています。フル・トランスポートابل・エクスポート/インポートのコマンドラインは、Oracle Data Pumpのコマンドラインと同一です。フル・トランスポートابل・エクスポート/インポートでは、データベース・リンク経由でメタデータを移動する機能などのData Pumpのオプションを活用できます。さらに、単一のインポート・コマンドを使用した全データベースの移行を実行できます。

同時に、フル・トランスポータブル・エクスポート/インポートでは、ユーザー・データとアプリケーション・データの移動用に、トランスポータブル表領域のメカニズムが採用されています。これにより、大量のデータも、きわめて高速に移行できます。もっとも重要な点は、フル・トランスポータブル・エクスポート/インポートにより、従来のトランスポータブル表領域の操作で必要となっていた複雑な一連の手順を実行することなく、データベースの移行に必要なシステム、ユーザー、およびアプリケーションのすべてのメタデータを移動できることです。

このように、フル・トランスポータブル・エクスポート/インポートは、Oracle Data Pumpの使いやすさとトランスポータブル表領域のパフォーマンスを兼ね備えているため、迅速かつ簡単にデータベースを移行できる機能となっています。

フル・トランスポータブル・エクスポート/インポートによるプラグブル・データベースのサポート

フル・トランスポータブル・エクスポート/インポートは、移行先としてプラグブル・データベースを使用することを想定して設計されました。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、非CDBデータベースからPDB、PDBから別のPDB、またはPDBから非CDBデータベースに移行できます。データとメタデータをインポートおよびエクスポートする際には、プラグブル・データベースは非CDBデータベースとまったく同様に動作します。

非CDBデータベースからプラグブル・データベースに移行する手順は、次のとおりです。

1. プラグブル・データベースの作成コマンドを使用して移行先のCDBに新しいPDBを作成します。
2. ソース・データベースのユーザーおよびアプリケーションの表領域を読み取り専用を設定します。
3. 表領域のデータファイルを移行先にコピーします。
4. DATAPUMP_IMP_FULL_DATABASE権限を持つアカウントを使用して、次のいずれかの操作を実行します。
 - FULL=Y TRANSPORTABLE=ALWAYSオプションを含むexpdpを使用してソース・データベースからエクスポートし、impdpを使用してターゲット・データベースにインポートします。
 - impdpを使用して、データベース・リンク経由でソースからターゲットにインポートします。
5. 通常実施される移行後の検証またはテストを実行します。

フル・トランスポータブル・エクスポート/インポートの内部詳細

ここでは、フル・トランスポータブル・エクスポート/インポートの基盤となるメカニズムについて説明します。この項は、詳細な技術的論文として記述されているわけではなく、フル・トランスポータブル・エクスポート/インポートで実行される内容と、フル・トランスポータブル・エクスポート/インポートによってユーザビリティとパフォーマンスの最適な組合せが実現される仕組みをより深く理解できるようにすることを目的としています。

フル・トランスポート/エクスポート/インポートの概要

フル・トランスポート/エクスポート/インポートの内容を理解するために、データの移動における従来の方法とトランスポートの方法の違いと、管理表領域とユーザー表領域の区別について把握しておく必要があります。

データの移動に従来の方法を使用する場合は、Oracle Data Pumpで外部表かダイレクト・パス・アンロードのいずれかが使用されて、データが抽出されます。アンロードされているデータの構造とタイプにより、これら2つのアクセス方法から使用される方法が決定されますが、いずれの方法でも、データの論理サブセットがOracleデータベースから効率的に抽出されます。

対照的に、データと索引のトランスポートの移動には、1つ以上の表領域のデータファイルの物理的な移動が含まれます。表領域のデータファイル内部のデータ・セグメントは、個別には読み取られません。

代わりに、エクスポート処理によって、各データファイル内のストレージを含むオブジェクトを記述するメタデータが抽出され、各ファイルが単一のエンティティとして移動されます。トランスポート表領域を使用した大量のデータの移動は、従来のデータの移動よりも高速になります。これは、個々のデータの行または索引エントリを解析および抽出する必要がないためです。トランスポート方式で個々の表またはパーティションを移動できますが、この場合も同様に表領域のデータファイル全体が移動されます。

従来のデータの移動とトランスポートのデータの移動の違いを理解しておく、管理表領域とユーザー表領域の区別について考える際に役立ちます。管理表領域は、フル・トランスポート/エクスポートのためのオラクル独自の表領域であり、SYSTEM、SYSAUX、TEMP、およびUNDOなどの種類があります。これらの表領域には、Oracleデータベースの中核機能とオラクル独自のデータベース・コンポーネント（Oracle Spatial、Oracle Text、OLAP、JAVAVM、およびXML Databaseなど）に対応する、プロシージャ、パッケージ、およびシード・データが含まれています。対照的に、ユーザー表領域は、データベース・ユーザーまたはアプリケーションによって定義された表領域です。この表領域には、ユーザー・データ、アプリケーション・データ、およびデータベースのユーザーによって定義されたその他の情報が格納されます。

フル・トランスポート/エクスポートの最初の手順は、移行先データベースを作成することです。この新しく作成されたデータベースには、オラクル独自のコンポーネントおよびパッケージを備えた、ターゲット環境に適切な一連の管理表領域が含まれます。Oracle Database 12c以降では、オラクル独自のオブジェクトは、Oracle Data Pumpでエクスポートされることも、インポートされることもありません。そのため、ソース・データベースの管理表領域からデータを移行する際は、フル・トランスポート/エクスポートで従来のデータの移動方法が使用されて、移行先データベースに追加する必要があるユーザー定義オブジェクトのデータとメタデータのみが抽出されます。

一方、ユーザー表領域は、全体エクスポートですべてのユーザーとアプリケーションのデータおよびメタデータがソースから移行先システムに移行されることを前提とした場合、完全な表領域のデータファイルとして移行先データベースに移動されます。そのため、ユーザー表領域はトランスポートに移動され、ユーザー・データの移行時に最大のパフォーマンスを実現します。

ユーザー表領域と管理表領域の両方にデータベース・オブジェクト（パーティション表など）が格納されている場合は、フル・トランスポート/フル・エクスポート/フル・インポートのみにおいて、考慮すべき点が1つあります。一般的に、この方法でオブジェクトを格納することは適切な方法ではありませんが、実行可能です。ストレージを含むオブジェクトがユーザー表領域と管理表領域の両方にある場合、データの転送前にそのオブジェクトを再定義するか、従来のData Pumpエクスポート/フル・インポートを使用できます。このホワイト・ペーパーの後半部分にある例では、フル・トランスポート/フル・エクスポートの開始前にこの状況を検出する方法が示されています。

フル・トランスポート/フル・エクスポート

フル・トランスポート/フル・エクスポート/フル・インポートにおける高いユーザビリティは、エクスポートですべてのユーザー・オブジェクトと管理オブジェクトのメタデータを抽出できることが要点となっています。Oracleデータベース・コンポーネントが内部的に使用するコールアウト・メカニズムとAPIが提供されているため、インポート時にデータベースのフル・コピーを作成するために必要なメタデータを完全に抽出できます。

通常、管理表領域に存在するすべてのユーザー・オブジェクトと管理オブジェクトはアンロードされます。これには、メタデータ（表の定義など）とデータ（その表の行など）の両方が含まれます。それに対して、ユーザー表領域に格納されているオブジェクトには、Data Pumpでアンロードされたメタデータのみが含まれます。そのオブジェクトのデータは、表領域のデータファイルに格納されてフル・トランスポート/フル・エクスポートに移動されます。

フル・トランスポート/フル・エクスポートの起動

Oracle Data Pumpエクスポート (`expdp`) は、フル・トランスポート/フル・エクスポートで使用できるコマンドライン・インタフェースです。パラメータ・ファイルまたはコマンドラインで2つのパラメータを指定することで、フル・トランスポート/フル・エクスポートを開始できます。指定するパラメータは、`TRANSPORTABLE=ALWAYS`と`FULL=Y`です。これらのパラメータ値により、Data Pumpで従来のエクスポート方法ではなく、フル・トランスポート/フル・エクスポートが使用されるように指定されます。

また、ソース・データベースのCOMPATIBLEデータベース初期化パラメータが12.0以上の値に設定されていない場合は、特別な考慮が必要です。たとえば、Oracle Database 11g Release 2 (11.2.0.3) からフル・トランスポート/フル・エクスポートを実行する場合、このことが当てはまります。この場合、エクスポートの結果がOracle Database 12c Release 1 (12.1) 以降のデータベースにインポートされることを示すために、Data Pumpパラメータ`VERSION=12`も指定する必要があります。

フル・トランスポート/フル・インポート

フル・トランスポート/フル・インポートは、従来のData Pumpインポートと同様に、ダンプ・ファイルのインポートにも、データベース・リンクを介したソース・データベースから移行先データベースへの直接的なインポートにも使用できます。ダンプ・ファイルを使用せずにインポートする機能により、フル・トランスポート/フル・インポートはデータベースの移行に不可欠なツールとなっています。

フル・トランスポートابل・インポートの起動

ダンプ・ファイルをインポートする場合、Oracle Data Pumpでは、ダンプ・ファイルの生成方法が従来の方法であるか、フル・トランスポートابل・エクスポートであるかを判断できます。そのため、ファイルベースのフル・トランスポートابل・インポートで必要になるのは、ダンプ・ファイルの名前と、TRANSPORT_DATAFILES=<datafile_names>パラメータを使用して転送されるユーザー表領域のデータファイルのリストを指定することだけです。複数のデータファイルを指定する場合は、このパラメータで、ファイルのカンマ区切りのリストを使用することも、複数回指定することもできます。オラクルでは、ファイル名の前後の引用符の適切な使用などに関する問題を回避するために、パラメータ・ファイルの使用を推奨しています。

ダンプ・ファイルを使用せずにソース・データベースからターゲット・データベースに直接移行する場合は、フル・トランスポートابل・モードの処理を開始するためのいくつかのパラメータを指定する必要があります。これは、実際にはネットワーク・モードのインポートで、ソース・データベースからのエクスポートとターゲット・データベースへのインポートが1つの処理として実行されるためです。ネットワークベースのフル・トランスポートابل・インポートでは、次のパラメータを指定する必要があります。

- FULL=Y TRANSPORTABLE=ALWAYS：これを指定することで、フル・トランスポートابل・エクスポートが実行されます。
- TRANSPORT_DATAFILES=<datafile_names>：これは、ファイルベースのフル・トランスポートابل・インポートで使用されます。
- VERSION=12：ソース・データベースのCOMPATIBLE設定が12.0未満の場合に指定します。フル・トランスポートابل・インポートでは、ソース・データベースをOracle Database 11g Release 2 (11.2.0.3) 以降にする必要があります。
- NETWORK_LINK=<dblink_name>：ソース・データベースからデータおよびメタデータが送信される際に経由するデータベース・リンクを指定します。

ユーザー表領域は、フル・トランスポートابل・インポート中に移行先システムで読取り/書込み可能な状態にすることができます。インポート・ジョブ中に他のユーザー・アプリケーションが実行されていないことが重要です。インポート・ジョブが正常に完了すると、すべてのユーザー表領域が読取り/書込み可能な状態に設定されます。

エンディアン変換

ソースとターゲットの両方のプラットフォームでエンディアン形式が同じである場合、同じプラットフォーム上であるかのようにデータファイルを転送できます。ソース・プラットフォームとターゲット・プラットフォームでエンディアンネスが異なる場合、Oracle Recovery Manager (Oracle RMAN) のCONVERTコマンドか、DBMS_FILE_TRANSFERパッケージのGET_FILEまたはPUT_FILEプロシージャを使用して、転送されるデータファイルをターゲット・プラットフォームの形式に変換する必要があります。

暗号化された表領域は、エンディアンネスが異なるプラットフォームには転送できません。

例：フル・トランスポートブルを使用した非CDBからCDBへの移動

この例では、フル・トランスポートブルを使用して、Oracle Solaris x86サーバーから、Oracle Linux上で稼働するCDB内のOracle Database 12cのPDBに、Oracle Database 11g Release 2 (11.2.0.3) データベースを移行します。この例ではダンプ・ファイルが使用されており、ソース・データベースからフル・トランスポートブル・エクスポートを実行した後で、移行先のデータベースでフル・トランスポートブル・インポートを実行する必要があります。ソース・データベースには、hr_1とhr_2の2つのユーザー表領域があります。

表1：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_1	あり	/u01/app/oracle/oradata/hr_db/hr_101.dbf
HR_2	なし	/u01/app/oracle/oradata/hr_db/hr_201.dbf

この例では、ターゲット・データベースのOracle SIDがHR_PDBであり、ターゲットPDBのデータファイルが移行先サーバーの/u01/app/oracle/oradata/hr_pdb/ディレクトリに格納されていることを前提としています。

この例では表領域は暗号化されていません。透過的なデータ暗号化 (TDE) の設定と管理に使用するコマンドは、Oracle Database 11g Release 2とOracle Database 12cとの間で変更されました。データベースへのTDEの実装について詳しくは、『Oracle Database Advanced Securityガイド』を参照してください。

手順1：両方のプラットフォームのエンディアンネスの確認

プラットフォームのエンディアンネスを確認するには、各プラットフォームで次の問合せを実行します。

```
SQL> SELECT d.PLATFORM_NAME, ENDIAN_FORMAT
       FROM V$TRANSPORTABLE_PLATFORM tp, V$DATABASE d
       WHERE tp.PLATFORM_ID = d.PLATFORM_ID;
```

この場合は、Oracle Solaris x86とOracle Enterprise Linuxの両方がリトル・エンディアン形式であるため、エンディアンの変換は不要です。

手順2：転送される一連の表領域が自己完結型であることの確認

一連の表領域を転送する前に、転送の対象である表領域に格納されたオブジェクトと、転送の対象ではないオブジェクトとの間に論理的依存関係も物理的依存関係もないことを確認する必要があります。たとえば、あるオブジェクトがユーザー表領域と管理表領域の両方に格納されている状況を検出する必要があります。これは完結チェックと呼ばれます。

表領域hr_1およびhr_2が自己完結型で、転送後に参照整合性制約が有効化することの検証が含まれるかどうかを確認するには、ソース・データベースで次のコマンドを実行します。

```
SQL> EXECUTE DBMS_TTS.TRANSPORT_SET_CHECK('hr_1,hr_2', TRUE);
```

フル・トランスポートブル・エクスポート/インポートでこのチェックを実行する場合は、データベースのすべてのユーザー表領域を含める必要があります。

このPL/SQLプロシージャを起動した後は、TRANSPORT_SET_VIOLATIONSビューをFROM句で指定してSELECT文を実行することで、すべての違反を表示できます。

```
SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

一連の表領域が自己完結型である場合は、このビューは空になります。違反が表示される場合は、フル・トランスポートブルの処理を続行する前にこれらの問題を解決する必要があります。

手順3：ソース・データベースでのディレクトリ・オブジェクトの作成

フル・トランスポートブル・エクスポートの開始前に、ダンプ・ファイルの書き込み先の場所を指すディレクトリ・オブジェクトを作成します。このディレクトリへの読取りと書き込みのアクセス権限は、DBAロールと、ユーザーSYSおよびSYSTEMに自動的に付与されます。

```
SQL> CREATE DIRECTORY dp_dir AS '/u01/app/datafiles';
```

このディレクトリ・オブジェクトは、フル・トランスポートブル・エクスポートの完了後にデータベースから削除できます。

手順4：表領域hr_1およびhr_2の読取り専用モードへの設定

エクスポートの期間中は、転送される表領域を読取り専用モードにする必要があります。この場合、ソース・データベースで2つのコマンドを発行する必要があります。

```
SQL> ALTER TABLESPACE hr_1 READ ONLY;
```

```
SQL> ALTER TABLESPACE hr_2 READ ONLY;
```

フル・トランスポートブル・エクスポートが完了し、表領域のデータファイルが移行先システムにコピーされたら、表領域を読取り/書き込みステータスに戻すことができます。

METRICS=Yと
EXCLUDE=STATISTICS
の使用は必須ではありませんが、Data Pump
エクスポートでは一般的に指定することが推
奨されています。

手順5: ソース・データベースでのフル・トランスポータブル・エクスポートの起動

DATAPUMP_EXP_FULL_DATABASEロールのユーザーとして、Data Pumpエクスポート・ユーティリティを起動します。

```
$ expdp system/manager full=y transportable=always version=12 ¥
    directory=dp_dir dumpfile=full_tts.dmp ¥
    metrics=y exclude=statistics ¥
    encryption_password=secret123word456 ¥
    logfile=full_tts_export.log
```

ソース・データベースがOracle Database 11g Release 2 (11.2.0.3) であるため、VERSION=12パラメータが必要です。現在のバージョンよりも大きいバージョン番号がexpdpコマンドで許可されるのは、この場合のみです。ソース・データベースがOracle Database 12cで、COMPATIBLEが12.0以上に設定されている場合、VERSIONパラメータは不要です。

エクスポート・コマンドが完了したら、エクスポート・ログ・ファイルに、ターゲットに移動する必要があるすべての表領域のデータファイルを示すリストが表示されます。

手順6: 表領域のデータファイルとエクスポート・ダンプ・ファイルのソースからターゲットへの転送

このデータベースを新しいサーバーに移行している最中であるため、移行先データベースからアクセス可能な場所に表領域のデータファイルとエクスポート・ダンプ・ファイルをコピーする必要があります。たとえば、移行先システムで次のコマンドを発行できます。

```
$ cd /u01/app/oracle/oradata/hr_pdb/
$ cp /net/<source-server>/u01/app/oracle/oradata/hr_db/hr_101.dbf .
$ cp /net/<source-server>/u01/app/oracle/oradata/hr_db/hr_201.dbf .
$ cp /net/<source-server>/u01/app/datafiles/full_tts.dmp .
```

手順7: 移行先データベースでのディレクトリ・オブジェクトの作成

Data Pumpダンプ・ファイルをHR_PDBのoradataディレクトリにコピーしているため、このインポートで同じディレクトリを指すようにディレクトリ・オブジェクトを作成します。このディレクトリ・オブジェクトは、PDBコンテナに接続するユーザーが作成する必要があります。

```
SQL> CREATE DIRECTORY dp_dir AS '/u01/app/oracle/oradata/hr_pdb';
SQL> GRANT read, write on directory dp_dir to system;
```

このディレクトリ・オブジェクトは、フル・トランスポータブル・インポートの完了後にデータベースから削除できます。

impdpコマンドの接続文字列に、PDBのサービス名を明示的に指定する必要があります。

手順8: 移行先データベースでのフル・トランスポートブル・インポートの起動

DATAPUMP_IMP_FULL_DATABASEロールのユーザーとして、Data Pumpインポート・ユーティリティを起動します。

```
$ impdp system/manager@hr_pdb directory=dp_dir ¥
  dumpfile=full_tts.dmp logfile=full_tts_imp.log ¥
  metrics=y ¥
  encryption_password=secret123word456 ¥
  transport_datafiles='/u01/app/oracle/oradata/hr_pdb/hr_101.dbf',¥
  '/u01/app/oracle/oradata/hr_pdb/hr_201.dbf'
```

この例では、コマンドラインで指定された、いくつかのパラメータが示されていますが、多くの場合、コマンドラインの空白文字と引用符に関する問題を回避するために、Data Pumpパラメータ・ファイルの使用が推奨されます。

この文が正常に実行されたら、移行先データベースで、ユーザー表領域が自動的に読取り/書込みモードに設定されます。インポート・ログ・ファイルで想定外のエラーが発生していないことを確認し、通常実施される移行後の検証とテストを実行します。

手順9: (オプション) ソース・データベースにおけるユーザー表領域の読取り/書込みモードへのリストア

フル・トランスポートブル・エクスポートの完了後に、必要に応じてソース・データベースでユーザー定義の表領域を読取り/書込みモードに戻すことができます。

```
SQL> ALTER TABLESPACE hr_101 READ WRITE;
SQL> ALTER TABLESPACE hr_201 READ WRITE;
```

移行先データベースでは、フル・トランスポートブル・インポートの完了時にすべての表領域が自動的に読取り/書込みモードに設定されます。

結論

フル・トランスポートブル・エクスポート/インポートは、データベースの移行プロセスを大幅に簡素化するOracle Database 12cの新機能です。フル・トランスポートブル・エクスポート/インポートには、Oracle Data Pumpの使いやすさとトランスポートブル表領域のパフォーマンスが統合されています。この機能により、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) 以降の場合に、1回の操作でOracle Database 12cにアップグレードまたは移行できます。フル・トランスポートブル・エクスポート/インポートは、プラグラブル・データベースへの移行に不可欠なツールです。これにより、マルチテナント・アーキテクチャに移行すると、それに付随してコストが削減され、スケール・メリットが得られるため、これらを活かすことができます。

付録：フル・トランスポートابل・エクスポート/インポートの制限事項

フル・トランスポートابل・エクスポート/インポートでは、従来のトランスポートابل表領域のユーザビリティが大幅に強化されていますが、一方で留意する必要があるデータの転送に関する制限事項があります。

- 従来のトランスポートابل表領域と同様に、フル・トランスポートابل・エクスポート/インポートでは、エクスポートの期間中に、転送されるユーザー表領域を読み取り専用モードにする必要があります。このことが望ましくない場合は（特に、本番環境での使用前にこの方法をテストするときに当てはまる可能性があります）、バックアップから表領域を転送するOracle RMANの機能を使用できます。バックアップからの表領域の転送について詳しくは、『Oracle Databaseバックアップおよびリカバリ・ユーザズ・ガイド』を参照してください。
- エクスポート対象として選択したストレージを含むすべてのオブジェクトで、ストレージ・セグメント全体が管理表領域、非トランスポートابل表領域、ユーザー定義のトランスポートابل表領域のいずれかにすべて含まれる必要があります。単一のオブジェクトのストレージを、2種類の表領域にまたがらせることはできません。複数の表領域に格納されている場合は、トランスポートابل・モードが妨げられて、エクスポートが失敗します。この場合、通常どおりにエクスポートを実行するか、この条件を満たすようにオブジェクトのストレージを再定義する必要があります。
- ソース・プラットフォームとターゲット・プラットフォームでエンディアンネスが異なる場合、転送されるデータをターゲット・プラットフォームの形式に変換する必要があります。
- データの変換には、Oracle RMANのCONVERT DATAFILEコマンドを使用できます。
- または、DBMS_FILE_TRANSFERパッケージのGET_FILEかPUT_FILEプロシージャを使用して転送することで、データファイルをターゲット・プラットフォームの形式に変換できます。
- 暗号化された表領域は、エンディアンネスが異なるプラットフォームには転送できません。
- 暗号化された表領域を同じエンディアンネスのプラットフォームに転送するには、エクスポートの際にexpdpコマンドでENCRYPTION_PASSWORDパラメータを使用し、インポートの際にimpdpコマンドでENCRYPTION_PASSWORDパラメータに同じ値を使用します。
- XDBリポジトリはフル・トランスポートابل・エクスポート/インポートではサポートされていません。ただし、ユーザー定義のXMLスキーマはエクスポート/インポートのすべての形式（Data Pumpエクスポート/インポート、トランスポートابل表領域、フル・トランスポートابل・エクスポート/インポート）でサポートされています。
- 自動ワークロード・リポジトリ（AWR）はフル・トランスポートابل・エクスポート/インポートではサポートされていません。Oracleデータベース間でAWRデータを移動するには、awrextr.sqlスクリプトとawrload.sqlスクリプトを使用します。
- ソースとターゲットのデータベースで、互換性のあるデータベースのキャラクタ・セットを使用する必要があります。ここで挙げた内容と、データの転送に関するその他の一般的な制限事項について詳しくは、『Oracle Database管理者ガイド』を参照してください。



Oracle Database 12c :
フル・トランスポータブル・
エクスポート/インポート
2014年1月
著者 : Roy F. Swonger
貢献者 : George Claborn、Carol Palmer、
Jim Stenois

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

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

oracle.com



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

Copyright © 2012, Oracle and/or its affiliates. All rights reserved.本書は情報提供のみを目的としており、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。

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。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。UNIXはX/Open Company, Ltd.によってライセンス提供された登録商標です。0410

Hardware and Software, Engineered to Work Together