

Oracle® Database

異なるエンディアン・オペレーティング・システムでの 非 CDB の PDB への移行と変換

19c

F16179-02(原本部品番号:F10904-02)

2019年5月

タイトルおよび著作権情報

Oracle Database異なるエンディアン・オペレーティング・システムでの非CDBのPDBへの移行と変換, 19c
F16179-02

Copyright © 2018, 2019, Oracle and/or its affiliates. All rights reserved.

原著者: Sunil Surabhi, Nirmal Kumar

原協力者: Lance Ashdown, Padmaja Potineni, Rajesh Bhatiya, Prakash Jashnani, Douglas Williams, Mark Bauer

原協力者: Roy Swonger, Byron Motta, Hector Vieyra Farfan, Carol Tagliaferri, Mike Dietrich, Marcus Doeringer, Umesh Aswathnarayana Rao, Rae Burns, Subrahmanyam Kodavaluru, Cindy Lim, Amar Mbaye, Akash Pathak, Thomas Zhang, Zhihai Zhang

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。お客様との間に適切な契約が定められている場合を除いて、オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。お客様との間に適切な契約が定められている場合を除いて、オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [このドキュメントのユースケース・シナリオ](#)
 - [ドキュメントのアクセシビリティについて](#)
- [1 データベースのトランスポート](#)
 - [プラットフォーム間でのデータ・トランスポート](#)
 - [データのトランスポートに関する一般的な制限事項](#)
 - [データのトランスポートの互換性に関する注意事項](#)
 - [フル・トランスポートブル・エクスポート/インポートに関する制限事項](#)
- [2 ターゲット・オペレーティング・システムのendian形式へのデータ変換](#)
 - [プラットフォーム間でのデータの変換](#)
 - [DBMS_FILE_TRANSFERパッケージを使用したプラットフォーム間でのデータの変換](#)
 - [RMANを使用したプラットフォーム間でのデータの変換](#)
 - [エクスポート後のソース・システムでの表領域の変換](#)
 - [インポート前のターゲット・システムでのデータファイルの変換](#)
- [3 SQL*Plusを使用したデータベースへの接続](#)
 - [ステップ1: コマンド・ウィンドウのオープン](#)
 - [ステップ2: オペレーティング・システムの変数設定](#)
 - [ステップ3: SQL*Plusの起動](#)
 - [ステップ4: SQL*PlusのCONNECTコマンドの発行](#)
- [4 Oracle Databaseの移行](#)
 - [キャラクタ・セットの互換性の確認](#)
 - [エクスポート・ダンプ・ファイルを使用したデータベースのトランスポート](#)

はじめに

このガイドでは、特定のユースケース・シナリオの完了に役立てるために集められた、Oracle Databaseユーザー支援ドキュメントの各トピックについて説明します。

- [このドキュメントのユースケース・シナリオ](#)
- [ドキュメントのアクセシビリティについて](#)

このドキュメントのユースケース・シナリオ

このシナリオ・ドキュメントを使用すると、以前のリリースの非CDBを、異なるエンディアン・オペレーティング・システムで実行されているPDBに変換できます。変換後、そのPDBをOracle Database19cに接続できます。

このシナリオは、次の非CDBに適用されます。

- Oracle Database 11gリリース2(11.2.0.3)
- Oracle Database 11gリリース2(11.2.0.4)
- Oracle Database 12c リリース1(12.1.0.1)
- Oracle Database 12c リリース1(12.1.0.2)
- Oracle Database 12cリリース2 (12.2.0.1)

このシナリオの前提条件

- ソース非CDBおよびターゲットCDBホストは、互換性のあるキャラクタ・セットおよび各国語キャラクタ・セットを使用する必要があります。
- マルチテナント・アーキテクチャを構成オプションとして使用し、CDBおよびデフォルトPDBを持つように、ターゲットCDBホストにOracle Database19cを作成および構成しました。
- ソース非CDBのユーザー定義表領域を読み取り専用モードにする準備ができました。

このシナリオの概要

1. **データベースのトランスポート。**新しいプラットフォームにデータベースをトランスポートします
2. **ターゲット・オペレーティング・システムのendian形式へのデータ変換。** endian形式を変換してから、データを新しいプラットフォームに変換します。
3. **Oracleデータベースへの接続**
4. **Oracle Databaseの移行。**ターゲット・マルチテナント・アーキテクチャ・ホストのPDBにデータベースを移行します。

これらのステップは、このドキュメントの各章に対応しています。

親トピック: [はじめに](#)

ドキュメントのアクセシビリティについて

Oracleのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracleサポートへのアクセス

サポートを購入したオラクル社のお客様は、My Oracle Supportを介して電子的なサポートにアクセスできます。詳細情報は

(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

親トピック: [はじめに](#)

1 データベースのトランスポート

新しいOracle Databaseインスタンスへデータベースをトランスポートできます。

- [プラットフォーム間でのデータ・トランスポート](#)
プラットフォーム間でデータをトランスポートできます。
- [データのトランスポートに関する一般的な制限事項](#)
データのトランスポートに関する一般的な制限事項があります。また、フル・トランスポートابل・エクスポート/インポート、トランスポートابل表領域またはトランスポートابل表に固有の制限事項もあります。
- [データのトランスポートの互換性に関する注意事項](#)
データをトランスポートする場合、Oracle Databaseはターゲット・データベースが稼働する最低限の互換性レベルを計算します。
- [フル・トランスポートابل・エクスポート/インポートに関する制限事項](#)
フル・トランスポートابل・エクスポート/インポートには制限事項があります。

プラットフォーム間でのデータ・トランスポート

プラットフォーム間でデータをトランスポートできます。

プラットフォーム間でデータをトランスポートする機能を使用すると、次のことが可能です。

- データベースをプラットフォーム間で移行できます。
- コンテンツ・プロバイダは、簡単にかつ効率的に構造化データを公開し、別のプラットフォームでOracle Databaseを実行している顧客に配布できます。
- データ・ウェアハウス環境からデータ・マート(多くの場合、小規模プラットフォームで実行されている)へのデータの配布を簡素化できます。
- 異なるオペレーティング・システムまたはプラットフォーム上のOracle Databaseインストール間で読取り専用表領域を共有できます。この場合、次の各項で説明するように、それらのプラットフォームおよびendiannessが同じプラットフォームからストレージ・システムにアクセスできることが前提となります。

多くのプラットフォーム(すべてではありません)では、クロス・プラットフォームでのデータのトランスポートがサポートされます。

V\$TRANSPORTABLE_PLATFORMビューを問い合わせると、サポートされているプラットフォームを参照して、各プラットフォームのendian形式(バイトの並び順)を確認できます。次の問合せを実行すると、プラットフォーム間でのデータのトランスポートがサポートされているプラットフォームが表示されます。

```
COLUMN PLATFORM_NAME FORMAT A40
```

```
COLUMN ENDIAN_FORMAT A14
```

```
SELECT PLATFORM_ID, PLATFORM_NAME, ENDIAN_FORMAT  
FROM V$TRANSPORTABLE_PLATFORM  
ORDER BY PLATFORM_ID;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
3	HP-UX (64-bit)	Big
4	HP-UX IA (64-bit)	Big
5	HP Tru64 UNIX	Little
6	AIX-Based Systems (64-bit)	Big
7	Microsoft Windows IA (32-bit)	Little
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
10	Linux IA (32-bit)	Little
11	Linux IA (64-bit)	Little
12	Microsoft Windows x86 64-bit	Little
13	Linux x86 64-bit	Little
15	HP Open VMS	Little
16	Apple Mac OS	Big
17	Solaris Operating System (x86)	Little
18	IBM Power Based Linux	Big
19	HP IA Open VMS	Little
20	Solaris Operating System (x86-64)	Little
21	Apple Mac OS (x86-64)	Little

ソース・プラットフォームとターゲット・プラットフォームが同じendiannessの場合、データはデータ変換なしでソース・プラットフォームからターゲット・プラットフォームにトランスポートされます。

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endiannessが異なる場合は、トランスポートするデータをターゲット・プラットフォーム

プラットフォームの形式に変換する必要があります。次のいずれかの方法を使用してデータを変換できます。

- DBMS_FILE_TRANSFERパッケージのGET_FILEまたはPUT_FILEプロシージャ

これらのプロシージャのいずれかを使用して、ソース・プラットフォームとターゲット・プラットフォーム間でデータファイルを移動する場合、各データファイル内の各ブロックがターゲット・プラットフォームのendiannessに変換されます。変換はターゲット・プラットフォームで発生します。

- RMAN CONVERTコマンド

ソース・プラットフォームまたはターゲット・プラットフォームでRMAN CONVERTコマンドを実行します。このコマンドにより、トランスポートするデータがターゲット・プラットフォームの形式に変換されます。



注意:

UNDO セグメントを含むデータファイルでは、異なる endian 形式間でのデータファイルの変換はサポートされていません。

データファイルのデータを別のプラットフォームにトランスポートする前に、そのデータが属しているプラットフォームをデータファイル・ヘッダーで識別する必要があります異なるプラットフォームのOracle Databaseインストール間で読取り専用表領域をトランスポートするには、少なくとも1回データファイルを読取り/書込みにします。

関連項目:

[「プラットフォーム間でのデータの変換」](#)

親トピック: [データベースのトランスポート](#)

データのトランスポートに関する一般的な制限事項

データのトランスポートに関する一般的な制限事項があります。また、フル・トランスポートابل・エクスポート/インポート、トランスポートابل表領域またはトランスポートابل表に固有の制限事項もあります。

データをトランスポートする場合は、次の一般的な制限事項に注意してください。

- ソース・データベースとターゲット・データベースで、互換性のあるデータベース・キャラクタ・セットを使用している必要があります。具体的には、次の内容のいずれかを満たしている必要があります。
 - ソース・データベースとターゲット・データベースのデータベース・キャラクタ・セットが同じです。
 - ソース・データベース・キャラクタ・セットがターゲット・データベース・キャラクタ・セットの厳格な(バイナリ)サブセットであり、かつ、次の3つの条件が満たされています。
 - ソース・データベースは、Oracle Database 10g リリース1 (10.1.0.3)以上です。
 - トランスポート対象の表領域に、文字長セマンティクスが使用される表の列が含まれていないか、またはソースとターゲットの両方のデータベースにおいて、データベース・キャラクタ・セットの最大文字幅が同じです。
 - トランスポート対象のデータにはCLOBデータ型の列が含まれていないか、またはソース・データベースとターゲット・データベースにおいて、データベース・キャラクタ・セットが両方ともシングルバイトであるか、両方ともマルチバイトです。
- ソース・データベース・キャラクタ・セットがターゲット・データベース・キャラクタ・セットの厳格な(バイナリ)サブセットであり、かつ、次の2つの条件が満たされています。
 - ソース・データベースは、Oracle Database 10g リリース1 (10.1.0.3)以前です。
 - ソース・データベースとターゲット・データベース・キャラクタ・セットにおいて、最大文字幅が同じです。

注意:



Oracle Database によって認識されるキャラクタ・セット間のサブセットとスーパーセットの関係は、[『Oracle Database グローバリゼーション・サポート・ガイド』](#)を参照してください。

- ソースとターゲットのデータベースで、互換性のある各国語キャラクタ・セットを使用している必要があります。具体的には、次の内容のいずれかを満たしている必要があります。
 - ソースとターゲットのデータベースの各国語キャラクタ・セットが同じです。
 - ソース・データベースはOracle Database 10g リリース1 (10.1.0.3)以上であり、トランスポート対象の表領域にはNCHAR、NVARCHAR2、NLOBのデータ型の列が含まれていません。
- トランスポートابل・エクスポート操作を実行する場合は、次の制限が適用されます。
 - エクスポートを実行するユーザーのデフォルトの表領域を、転送対象となっている表領域のいずれかにすることはできません。
 - エクスポートを実行するユーザーのデフォルトの表領域を、書込み可能にする必要があります。
- 非CDBで、ターゲット・データベースに同じ名前の表領域が含まれている場合は、表領域をトランスポートできません。

CDBで、ターゲットのコンテナに同じ名前の表領域が含まれている場合は、表領域をトランスポートできません。ただし、異なるコンテナには同じ名前の表領域を格納できます。

REMAP_TABLESPACEインポート・パラメータを使用して、データベース・オブジェクトを異なる表領域にインポートできます。または、トランスポート操作を実行する前に、トランスポート対象の表領域またはターゲットの表領域のいずれかの名前を変更できます。

Oracle Database 12cリリース2 (12.2)以降では、Recovery Manager (RMAN)のRECOVERコマンドで、表領域を再マッピングしながら表を異なるスキーマに移動できます。詳細は、『[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。

- CDBでは、デフォルトのデータ・ポンプ・ディレクトリ・オブジェクトDATA_PUMP_DIRはPDBでは機能しません。データ・ポンプ・エクスポートおよびインポートで使用する明示的なディレクトリ・オブジェクトをPDB内に定義する必要があります。
- XMLTypeを含むデータのトランスポートには、次の制限事項があります。
 - ターゲット・データベースにXML DBがインストールされている必要があります。
 - XMLType表が参照するスキーマをXML DB標準スキーマにすることはできません。
 - トランスポートされたXMLType表のスキーマがターゲット・データベース内に存在しない場合は、スキーマがインポートおよび登録されます。ターゲット・データベース内にすでにスキーマが存在する場合は、インポート中にメッセージが表示されます。
 - XMLTypeを含むデータのメタデータは、データ・ポンプのみを使用してエクスポートおよびインポートする必要があります。

次の問合せでは、XMLTypeを含む表領域のリストが返されます。

```
select distinct p.tablespace_name from dba_tablespaces p,  
       dba_xml_tables x, dba_users u, all_all_tables t where  
       t.table_name=x.table_name and t.tablespace_name=p.tablespace_name  
       and x.owner=u.username;
```

XMLTypeの詳細は、『[Oracle XML DB開発者ガイド](#)』を参照してください。

- 解釈がアプリケーション固有で、データベースに対して不透明なタイプ(RAW、BFILEなど)は、トランスポートできますが、クロス・プラットフォームのトランスポート操作では変換されません。このタイプの実際の構造はアプリケーションのみが認識するため、このタイプが新規プラットフォームに移動した後、アプリケーションではendiannessの問題に対処する必要があります。OPAQUE型を使用するタイプとオブジェクトも、直接的または間接的にこの制限の影響を受けます。
- TIMESTAMP WITH LOCAL TIME ZONE (TSLTZ)データを含む表がある表領域をタイム・ゾーンが異なるデータベース間でトランスポートする場合、TSLTZデータを含む表はトランスポートされません。エラー・メッセージで、トランスポートされなかった表の説明が示されます。ただし、表領域内のTSLTZデータを含まない表はトランスポートされます。

データベースのタイム・ゾーンを確認するには、次の問合せを使用します。

```
SELECT DBTIMEZONE FROM DUAL;
```

ALTER DATABASE SQL文を使用すると、データベースのタイム・ゾーンを変更できます。

データ・ポンプを使用すると、トランスポート操作の完了後に、TSLTZデータを含む表の従来のエクスポート/インポートを実行できます。

- プラットフォーム間でのトランスポート操作の一部として、アナリティック・ワークスペースは使用できません。ソース・プラットフォームおよびターゲット・プラットフォームが別な場合、Data Pumpエクスポート/インポートを使用し、アナリティック・ワークスペースをエクスポートおよびインポートします。アナリティック・ワークスペースの詳細は、『[Oracle OLAP DML](#)』

[ファレンス](#)』を参照してください。

注意:



データ・ポンプ・エクスポート・ユーティリティ expdp またはインポート・ユーティリティ impdp は、Oracle サポート・サービスから要求された場合以外、SYSDBA として起動しないでください。SYSDBA は内部的に使用され、一般ユーザーとは異なる特別な機能を持ちます。

親トピック: [データベースのトランスポート](#)

データのトランスポートの互換性に関する注意事項

データをトランスポートする場合、Oracle Databaseはターゲット・データベースが稼働する最低限の互換性レベルを計算します。

ターゲット・データベースが同じプラットフォームにある場合も、別のプラットフォームにある場合も、トランスポート可能な表領域を使用して、同じ互換性または高い互換性が設定されているターゲット・データベースにソース・データベースから表領域または表をトランスポートできます。ソース・データベースの互換性レベルがターゲット・データベースの互換性レベルよりも高い場合、データ・トランスポート操作が失敗します。

次の表に、様々な使用例でのソース・データベースとターゲット・データベースの互換性の最低要件を示します。ソース・データベースとターゲット・データベースの互換性設定は同一である必要はありません。

表1-1 互換性の最低要件

トランスポートの使用例	互換性の最低設定	
	ソース・データベース	ターゲット・データベース
フル・トランスポート可能な・エクスポート/インポートを使用したデータベースのトランスポート	12.0 (Oracle Database 12c以降のデータベースの COMPATIBLE 初期化パラメータの設定) 12 (11.2.0.3 以上のデータベースの VERSION データ・ポンプ・エクスポート・パラメータの設定)	12.0 (COMPATIBLE 初期化パラメータの設定)
トランスポート可能な表領域を使用した、同じプラットフォーム上のデータベース間での表領域のトランスポート	8.0 (COMPATIBLE 初期化パラメータの設定)	8.0 (COMPATIBLE 初期化パラメータの設定)
トランスポート可能な表領域を使用した、ターゲット・データベースとデータベース・ブロック・サイズが異なる表領域のトランスポート	9.0 (COMPATIBLE 初期化パラメータの設定)	9.0 (COMPATIBLE 初期化パラメータの設定)
トランスポート可能な表領域を使用した、異なるプラットフォーム上のデータベース間での表領域のトランスポート	10.0 (COMPATIBLE 初期化パラメータの設定)	10.0 (COMPATIBLE 初期化パラメータの設定)
データベース間での表のトランスポート	11.2.0 (Oracle Database 12c以降のデータベースの COMPATIBLE 初期化パラメータの設定)	11.2.0 (COMPATIBLE 初期化パラメータの設定)



- 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場合、ソース・データベースは Oracle Database 11g リリース 2 (11.2.0.3)以降のデータベースであり、ターゲット・データベースは Oracle Database 12c 以降のデータベースである必要があります。
- Oracle Database 11g リリース 2 (11.2.0.3)以降のデータベースから Oracle Database 12c 以降のデータベースにトランスポートする場合は、VERSION データ・ポンプ・エクスポート・パラメータを 12 以上に設定する必要があります。
- Oracle Database 19c データベースから Oracle Database 19c データベースにトランスポートする場合は、COMPATIBLE 初期化パラメータを 19.0.0 以上に設定する必要があります。

親トピック: [データベースのトランスポート](#)

フル・トランスポートابل・エクスポート/インポートに関する制限事項

フル・トランスポートابل・エクスポート/インポートには制限事項があります。

フル・トランスポートابل・エクスポート/インポートに関する次の制限事項に注意してください。

- フル・トランスポートابل・エクスポート/インポートには、[「データのトランスポートに関する一般的な制限事項」](#)で説明されている一般的な制限事項が適用されます。
- 暗号化された表領域は、endiannessが異なるプラットフォームにはトランスポートできません。
暗号化された表領域をendiannessが同じプラットフォームにトランスポートするには、エクスポート時に ENCRYPTION_PWD_PROMPTエクスポート・ユーティリティ・パラメータをYESに設定するか、ENCRYPTION_PASSWORDエクスポート・ユーティリティ・パラメータを使用します。インポート時は、同等のインポート・ユーティリティ・パラメータを使用し、値をエクスポートに使用されたものと同じパスワードに設定します。
- フル・トランスポートابل・エクスポート/インポートでは、ダイレクト・パスや外部表などの従来のデータ・ポンプ・エクスポート/インポートを使用して、管理表領域内のユーザー定義のデータベース・オブジェクトをエクスポートおよびインポートできます。管理表領域は、Oracle Databaseに用意されたSYSTEM表領域やSYSAUX表領域などの非ユーザー表領域です。
- フル・トランスポートابل・エクスポート/インポートでは、管理表領域(SYSTEMやSYSAUXなど)とユーザー定義表領域の両方で定義されているデータベース・オブジェクトはトランスポートできません。たとえば、パーティション表は、ユーザー定義表領域と管理表領域の両方に格納されることがあります。データベースにこのようなデータベース・オブジェクトがある場合は、それらのすべてのデータベース・オブジェクトが、管理表領域またはユーザー定義表領域のいずれかに格納されるように再定義した後、トランスポートできます。データベース・オブジェクトを再定義できない場合は、従来のデータ・ポンプ・エクスポート/インポートを使用できます。
- フル・トランスポートابل・エクスポート/インポートを使用して、ネットワーク経由でデータベースをトランスポートする場合に、監査証跡情報自体がユーザー定義表領域に格納されていると、管理表領域(SYSTEMやSYSAUXなど)に格納されている表に対して監査を有効にできません。詳細は、[『Oracle Databaseセキュリティガイド』](#)を参照してください。

親トピック: [データベースのトランスポート](#)

2 ターゲット・オペレーティング・システムのendian形式へのデータ変換

ターゲット・オペレーティング・システム・プラットフォームで使用されているendian形式にデータのendian形式を変換して、データ移行を準備します。

- [プラットフォーム間でのデータの変換](#)

トランスポート操作を実行するときに、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endiannessが異なる場合は、トランスポートす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。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endiannessが同じ場合は、データ変換は必要ありません。データを変換するには、DBMS_FILE_TRANSFERパッケージまたはRMAN CONVERTコマンドを使用できます。

- [DBMS_FILE_TRANSFERパッケージを使用したプラットフォーム間でのデータの変換](#)

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すると、データファイルの転送中にプラットフォーム間でデータを変換できます。

- [RMANを使用したプラットフォーム間でのデータの変換](#)

RMAN CONVERTコマンドを使用してデータを変換する場合は、データ・ポンプ・エクスポートを実行した後にソース・プラットフォームでデータを変換するか、またはデータ・ポンプ・インポートを実行する前にターゲット・プラットフォームでデータを変換できます。いずれの場合も、データファイルをソース・システムからターゲット・システムに転送する必要があります。

プラットフォーム間でのデータの変換

トランスポータブル操作を実行するとき、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endiannessが異なる場合は、トランスポートす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。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endiannessが同じ場合は、データ変換は必要ありません。データを変換するには、DBMS_FILE_TRANSFERパッケージまたはRMAN CONVERTコマンドを使用できます。

注意:

これらの項には記載されていない制限事項が適用されることがあります。詳細は、次のドキュメントを参照してください。



- プラットフォームのendiannessをチェックする方法の詳細は、[「プラットフォーム間でのデータのトランスポート」](#)を参照してください
- DBMS_FILE_TRANSFER パッケージに関する制限事項の詳細は、[『Oracle Database PL/SQL パッケージおよびタイプ・リファレンス』](#)を参照してください。
- RMAN CONVERT コマンドに関する制限事項の詳細は、[『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。

親トピック: [ターゲット・オペレーティング・システムのendian形式へのデータ変換](#)

DBMS_FILE_TRANSFERパッケージを使用したプラットフォーム間でのデータの変換

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すると、データファイルの転送中にプラットフォーム間でデータを変換できます。

これらのプロシージャのいずれかを使用して、ソース・プラットフォームとターゲット・プラットフォーム間でデータファイルを移動する場合、各データファイル内の各ブロックがターゲット・プラットフォームのendiannessに変換されます。

この項では、例を使用して、データファイルを異なるプラットフォームに変換するためにDBMS_FILE_TRANSFERパッケージを使用する方法を示します。この例では、次のことを想定しています。

- GET_FILEプロシージャでデータファイルを転送します。
- mytable. 342. 123456789データファイルを異なるプラットフォームに転送します。
- ソース・プラットフォームのendiannessはターゲット・プラットフォームのendiannessと異なります。
- ソース・データベースのグローバル名はdbsa. example. comです。
- ソース・データベースとターゲット・データベースの両方でOracle Automatic Storage Management(Oracle ASM)が使用されています。

注意:



DBMS_FILE_TRANSFER パッケージを使用して、endianness が同じプラットフォーム間でデータファイルを転送することもできます。

GET_FILEプロシージャを使用して転送することによってデータファイルを変換するには、次のステップを実行します。

1. SQL*Plusを使用して、ディレクトリ・オブジェクトを作成できる管理ユーザーとしてソース・データベースに接続します。
2. ターゲット・データベースに転送するデータファイルを格納するディレクトリ・オブジェクトを作成します。

たとえば、+data/dbsa/datafileディレクトリに対してsales_dir_sourceというディレクトリ・オブジェクトを作成するには、次のSQL文を実行します。

```
CREATE OR REPLACE DIRECTORY sales_dir_source
AS '+data/dbsa/datafile';
```

ディレクトリ・オブジェクトの作成時に、指定したファイル・システム・ディレクトリが存在している必要があります。

3. SQL*Plusを使用して、データベース・リンクの作成、ディレクトリ・オブジェクトの作成およびDBMS_FILE_TRANSFERパッケージのプロシージャの実行ができる管理ユーザーとしてターゲット・データベースに接続します。
4. ターゲット・データベースからソース・データベースへデータベース・リンクを作成します。

ソース・データベースの接続されるユーザーには、ステップ2で作成したディレクトリ・オブジェクトについての読取り権限が必要です。

5. ソース・データベースから転送するデータファイルを格納するディレクトリ・オブジェクトを作成します。

DBMS_FILE_TRANSFERパッケージのプロシージャを実行するローカル・データベースのユーザーには、ディレクトリ・オブジェ

クトについての書き込み権限が必要です。

たとえば、+data/dbsb/datafileディレクトリに対してsales_dir_targetというディレクトリ・オブジェクトを作成するには、次のSQL文を実行します。

```
CREATE OR REPLACE DIRECTORY sales_dir_target
AS '+data/dbsb/datafile';
```

6. DBMS_FILE_TRANSFERパッケージのGET_FILEプロシージャを実行して、データファイルを転送します。

たとえば、次のプロシージャを実行し、ステップ4で作成したデータベース・リンクを使用して、mytable. 342. 123456789データファイルをソース・データベースからターゲット・データベースに転送します。

```
BEGIN
  DBMS_FILE_TRANSFER.GET_FILE (
    source_directory_object => 'sales_dir_source',
    source_file_name        => 'mytable. 342. 123456789',
    source_database         => 'dbsa.example.com',
    destination_directory_object => 'sales_dir_target',
    destination_file_name   => 'mytable');
END;
/
```

注意:



この例では、宛先データファイル名はmytableです。Oracle ASMでは、GET_FILEプロシージャのdestination_file_nameパラメータで完全修飾ファイル名形式を使用できません。

関連項目:

- DBMS_FILE_TRANSFERパッケージの使用方法の詳細は、[『Oracle Database PL/SQLパッケージおよびタイプ・リファレンス』](#)を参照してください。
- ASMの完全修飾ファイル名形式の詳細は、[『Oracle Automatic Storage Management管理者ガイド』](#)を参照してください。
- データベース・リンクの作成方法は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

親トピック: [ターゲット・オペレーティング・システムのendian形式へのデータ変換](#)

RMANを使用したプラットフォーム間でのデータの変換

RMAN CONVERTコマンドを使用してデータを変換する場合は、データ・ポンプ・エクスポートを実行した後にソース・プラットフォームでデータを変換するか、またはデータ・ポンプ・インポートを実行する前にターゲット・プラットフォームでデータを変換できます。いずれの場合も、データファイルをソース・システムからターゲット・システムに転送する必要があります。

次のRMAN CONVERTコマンドを使用してデータを変換できます。

- CONVERT DATAFILE
- CONVERT TABLESPACE
- CONVERT DATABASE

注意:



- RMAN CONVERT コマンドにはデータ型の制約が適用されます。
- RMAN CONVERT コマンドでは、UNDO セグメントを含むデータファイルについて、異なる endian 形式間でのデータファイルの変換はサポートされていません。

- [エクスポート後のソース・システムでの表領域の変換](#)

例を使用して、表領域を異なるプラットフォームに変換するために、RMAN CONVERT TABLESPACEコマンドを使用する方法を示します。

- [インポート前のターゲット・システムでのデータファイルの変換](#)

例を使用して、データファイルを異なるプラットフォームに変換するために、RMAN CONVERT DATAFILEコマンドを使用する方法を示します。

関連項目:

- [『Oracle Databaseバックアップおよびリカバリ・リファレンス』](#)
- [『Oracle Databaseバックアップおよびリカバリ・アドバンスド・ユーザズ・ガイド』](#)

親トピック: [ターゲット・オペレーティング・システムのendian形式へのデータ変換](#)

エクスポート後のソース・システムでの表領域の変換

例を使用して、表領域を異なるプラットフォームに変換するために、RMAN CONVERT TABLESPACEコマンドを使用する方法を示します。

この例では、次のことを想定しています。

- sales_1およびsales_2表領域を異なるプラットフォームにトランスポートします。
- ソース・プラットフォームのendiannessはターゲット・プラットフォームのendiannessと異なります。
- 表領域セットをターゲット・システムにトランスポートする前に、ソース・システムでデータを変換します。

- ソース・データベースでデータ・ポンプ・エクスポートを完了しています。

ソース・システムで表領域を変換するには、次のステップを実行します。

1. コマンド・プロンプトで、RMANを起動してソース・データベースに接続します。

```
$ RMAN TARGET /  
  
Recovery Manager: Release 12.1.0.1.0 - Production  
  
Copyright (c) 1982, 2012, Oracle and/or its affiliates. All rights reserved.  
  
connected to target database: salesdb (DBID=3295731590)
```

2. RMAN CONVERT TABLESPACEコマンドを使用してデータファイルを変換し、ソース・プラットフォーム上の一時的な場所に格納します。

この例では、一時的な場所はディレクトリ/tmpと想定し、すでに作成されているとします。変換されたデータファイルの名前は、システムによって割り当てられます。

```
RMAN> CONVERT TABLESPACE sales_1, sales_2  
2> TO PLATFORM 'Microsoft Windows IA (32-bit)'  
3> FORMAT '/tmp/%U';  
  
Starting conversion at source at 30-SEP-08  
using channel ORA_DISK_1  
channel ORA_DISK_1: starting datafile conversion  
input datafile file number=00007 name=/u01/app/oracle/oradata/salesdb/sales_101.dbf  
converted datafile=/tmp/data_D-SALESDB_I-1192614013_TS-SALES_1_FNO-7_03jru08s  
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:45  
channel ORA_DISK_1: starting datafile conversion  
input datafile file number=00008 name=/u01/app/oracle/oradata/salesdb/sales_201.dbf  
converted datafile=/tmp/data_D-SALESDB_I-1192614013_TS-SALES_2_FNO-8_04jru0aa  
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:25  
Finished conversion at source at 30-SEP-08
```

関連項目:

RMANのCONVERTコマンドの詳細は、[『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。

3. Recovery Managerを終了します。

```
RMAN> exit  
Recovery Manager complete.
```

4. データファイルをターゲット・システムに転送します。

親トピック: [RMANを使用したプラットフォーム間でのデータの変換](#)

インポート前のターゲット・システムでのデータファイルの変換

例を使用して、データファイルを異なるプラットフォームに変換するために、RMAN CONVERT DATAFILEコマンドを使用する方法を示します。

変換時、データファイルは、表領域名ではなく、ファイル名で指定します。表領域のメタデータがインポートされるまで、ターゲット・インスタンスでは対象の表領域名を認識できません。

この例では、次のことを想定しています。

- トランスポートする表領域のデータファイルはまだ変換していません。
DBMS_FILE_TRANSFERパッケージを使用してデータファイルをターゲット・システムに転送した場合、データファイルはファイル転送中に自動的に変換されています。[「DBMS_FILE_TRANSFERパッケージを使用したプラットフォーム間でのデータの変換」](#)を参照してください。
- 次のデータファイルを異なるプラットフォームにトランスポートします。
 - C:¥Temp¥sales_101.dbf
 - C:¥Temp¥sales_201.dbf
- ソース・プラットフォームのendiannessはターゲット・プラットフォームのendiannessと異なります。
- データ・ポンプ・インポートを実行する前に、ターゲット・システムでデータを変換します。
- 変換されたデータファイルは、C:¥app¥orauser¥oradata¥orawin¥(ターゲット・システムの既存データファイルの場所)に配置されます。

ターゲット・システムで表領域を変換するには、次のステップを実行します。

1. SQL*Plusを実行している場合は、ホスト・システムに戻ります。

```
SQL> HOST
```

2. RMAN CONVERT DATAFILEコマンドを使用して、データファイルをターゲット・プラットフォームに変換します。

```
C:¥>RMAN TARGET /
```

```
Recovery Manager: Release 12.1.0.1.0 - Production
```

```
Copyright (c) 1982, 2012, Oracle and/or its affiliates. All rights reserved.
```

```
connected to target database: ORAWIN (DBID=3462152886)
```

```
RMAN> CONVERT DATAFILE
```

```
2>' C:¥Temp¥sales_101.dbf',
```

```
3>' C:¥Temp¥sales_201.dbf'
```

```
4>TO PLATFORM="Microsoft Windows IA (32-bit)"
```

```
5>FROM PLATFORM="Solaris[tm] OE (32-bit)"
```

```
6>DB_FILE_NAME_CONVERT=
```

```
7>' C:¥Temp¥', ' C:¥app¥orauser¥oradata¥orawin¥'
```

```
8> PARALLELISM=4;
```

ソースの場所、ターゲットの場所、あるいはその両方でOracle Automatic Storage Management(Oracle ASM)を使用しない場合、ソース・プラットフォームとターゲット・プラットフォームはオプションです。RMANでは、データファイルを調べてソース・プラットフォームを判別し、デフォルトのターゲット・プラットフォームは、変換を実行するホストのプラットフォームになります。

ソースおよびターゲットの場所の両方でOracle ASMを使用する場合は、DB_FILE_NAME_CONVERT句でソース・プラットフォームとターゲット・プラットフォームを指定する必要があります。

関連項目:

RMANのCONVERTコマンドの詳細は、[『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。

3. Recovery Managerを終了します。

```
RMAN> exit  
Recovery Manager complete.
```

親トピック: [RMANを使用したプラットフォーム間でのデータの変換](#)

3 SQL*Plusを使用したデータベースへの接続

SQL*Plusを使用してOracle Databaseインスタンスに接続します。

- [ステップ1: コマンド・ウィンドウのオープン](#)
プラットフォームで必要な処理を実行して、オペレーティング・システムのコマンドを入力できるウィンドウを開きます。
- [ステップ2: オペレーティング・システムの環境変数の設定](#)
プラットフォームによっては、SQL*Plusを起動する前に、環境変数を設定するか少なくとも正しく設定されていることを確認することが必要な場合があります。
- [ステップ3: SQL*Plusの起動](#)
SQL*Plusを起動します。
- [ステップ4: SQL*PlusのCONNECTコマンドの発行](#)
最初にOracle Databaseインスタンスに接続する、または任意の時点で別のユーザーとして再接続するには、SQL*PlusのCONNECTコマンドを発行します。

ステップ1: コマンド・ウィンドウのオープン

プラットフォームで必要な処理を実行して、オペレーティング・システムのコマンドを入力できるウィンドウを開きます。

- コマンド・ウィンドウを開きます。

親トピック: [SQL*Plusを使用したデータベースへの接続](#)

ステップ2: オペレーティング・システムの環境変数の設定

プラットフォームによっては、SQL*Plusを起動する前に、環境変数を設定するか少なくとも正しく設定されていることを確認することが必要な場合があります。

たとえば、ほとんどのプラットフォームでは、環境変数ORACLE_SIDおよびORACLE_HOMEを設定する必要があります。さらに、ORACLE_HOME/binディレクトリを含めるようにPATH環境変数を構成する必要があります。一部のプラットフォームでは、さらに環境変数の設定が必要な場合があります。

- UNIXおよびLinuxでは、オペレーティング・システムのコマンドを入力して環境変数を設定します。
- Windowsでは、Windowsレジストリ内のORACLE_HOMEとORACLE_SIDにOracle Universal Installer (OUI)によって値が自動的に割り当てられます。

インストール時にデータベースを作成しなかった場合は、レジストリ内のORACLE_SIDがOUIによって設定されないため、後でデータベースを作成するときに、ORACLE_SID環境変数をコマンド・ウィンドウから設定する必要があります。

UNIXおよびLinuxのインストールには、oraenvとcoraenvの2つのスクリプトが含まれています。これらのスクリプトを使用すると、環境変数を容易に設定できます。詳細は、*Oracle Database管理者リファレンス for Linux and UNIX-Based Operating Systems*を参照してください。

いずれのプラットフォームでも、異なるOracleホームを使用するインスタンス間で切り替える場合は、ORACLE_HOME環境変数を変更する必要があります。同じOracleホームを複数のインスタンスで共有している場合は、インスタンスを切り替えるときにORACLE_SIDのみを変更する必要があります。

例3-1 UNIXの環境変数の設定(Cシェル)

```
setenv ORACLE_SID orcl
setenv ORACLE_HOME /u01/app/oracle/product/database_release_number/dbhome_1
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib:/usr/lib:/usr/dt/lib:/usr/openwin/lib:/usr/ccs/lib
```

例3-2 UNIXの環境変数の設定(Bashシェル)

```
export ORACLE_SID=orcl
export ORACLE_HOME=/u01/app/oracle/product/database_release_number/dbhome_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/dt/lib:/usr/openwin/lib:/usr/ccs/lib
```

例3-3 Windowsの環境変数の設定

```
SET ORACLE_SID=orawin2
```

[例3-3](#)では、ORACLE_HOMEおよびORACLE_SIDがレジストリに設定されているが、別のインスタンスに接続するにはORACLE_SIDのレジストリ値を上書きする必要があると仮定しています。

Windowsでは、コマンド・プロンプト・ウィンドウで設定した環境変数値でレジストリの値が上書きされます。

親トピック: [SQL*Plusを使用したデータベースへの接続](#)

ステップ3: SQL*Plusの起動

SQL*Plusを起動します。

1. 次のいずれかを行います:

- PATH環境変数にORACLE_HOME/binが含まれていることを確認します。
- ORACLE_HOME/binディレクトリに移動します。PATH環境変数にドット(".")が含まれていることを確認します。

2. 次のコマンドを入力します(UNIXとLinuxでは大/小文字が区別されます)。

```
sqlplus /nolog
```

完全なパスを指定してsqlplusコマンドを実行することもできます。

```
ORACLE_HOME/bin/sqlplus /nolog
```

親トピック: [SQL*Plusを使用したデータベースへの接続](#)

ステップ4: SQL*PlusのCONNECTコマンドの発行

最初にOracle Databaseインスタンスに接続する、または任意の時点で別のユーザーとして再接続するには、SQL*PlusのCONNECTコマンドを発行します。

- SQL*Plusで、CONNECTコマンドを発行します。

例3-4 ローカル・データベース・ユーザーへの接続

次の簡単な例では、ユーザーSYSTEMでローカル・データベースに接続します。SQL*Plusによって、ユーザーSYSTEMのパスワードの入力が求められます。

```
connect system
```

例3-5 SYSDBA権限でのローカル・データベース・ユーザーへの接続

次の例では、SYSDBA権限のユーザーSYSでローカル・データベースに接続します。SQL*Plusによって、ユーザーSYSのパスワードの入力が求められます。

```
connect sys as sysdba
```

ユーザーSYSとして接続するときには、AS SYSDBAを指定して接続する必要があります。

例3-6 SYSBACKUP権限でのローカル・データベース・ユーザーへの接続

次の例では、SYSBACKUP権限のユーザーSYSBACKUPでローカル・データベースに接続します。SQL*Plusによって、ユーザーSYSBACKUPのパスワードの入力が求められます。

```
connect sysbackup as sysbackup
```

ユーザーSYSBACKUPとして接続するときには、AS SYSBACKUPを指定して接続する必要があります。

例3-7 オペレーティング・システム認証によるSYSDBA権限でのローカルな接続

次の例では、オペレーティング・システム認証を使用してSYSDBA権限でローカルに接続します。

```
connect / as sysdba
```

例3-8 簡易接続構文による接続

この例では、簡易接続の構文を使用し、ユーザーsalesadminとして、ホストdbhost.example.comで動作するリモート・データベースに接続します。Oracle Netリスナー(リスナー)は、デフォルト・ポート(1521)でリスニングしています。データベース・サービスは、sales.example.comです。SQL*Plusによって、ユーザーsalesadminのパスワードの入力が求められます。

```
connect salesadmin@"dbhost.example.com/sales.example.com"
```

例3-9 サービス・ハンドラ・タイプを指定した簡易接続構文による接続

次の例は、サービス・ハンドラのタイプが指定されていること以外は[例3-8](#)と同じです。

```
connect salesadmin@"dbhost.example.com/sales.example.com:dedicated"
```

例3-10 デフォルト以外のリスナー・ポートを使用した簡易接続構文による接続

次の例は、デフォルト・ポート番号ではない1522でリスナーがリスニングしていること以外は[例3-8](#)と同じです。

```
connect salesadmin@"dbhost.example.com:1522/sales.example.com"
```

例3-11 ホストIPアドレスを使用した簡易接続構文による接続

この例は[例3-8](#)とほぼ同じで、ホスト名のかわりにホストのIPアドレスが使用されている点のみが異なります。

```
connect salesadmin@"192.0.2.5/sales.example.com"
```

例3-12 IPv6アドレスによる接続

この例では、IPv6アドレスを使用して接続します。角括弧で囲まれている点に注意してください。

```
connect salesadmin@"[2001:0DB8:0:0::200C:417A]/sales.example.com"
```

例3-13 インスタンスの指定による接続

次の例では、接続先のインスタンスを指定し、データベース・サービス名を省略します。インスタンスのみを指定する場合はサービス・ハンドラのタイプを指定できないことに注意してください。

```
connect salesadmin@"dbhost.example.com//orcl"
```

例3-14 ネット・サービス名による接続

次の例では、ユーザーsalesadminとして、ネット・サービス名sales1によって指定されたデータベース・サービスにリモート接続します。SQL*Plusによって、ユーザーsalesadminのパスワードの入力が求められます。

```
connect salesadmin@sales1
```

例3-15 外部認証による接続

次の例では、外部認証を使用して、ネット・サービス名sales1によって指定されたデータベース・サービスにリモート接続します。

```
connect /@sales1
```

例3-16 SYSDBA権限と外部認証による接続

次の例では、外部認証を使用してSYSDBA権限で、ネット・サービス名sales1によって指定されたデータベース・サービスにリモート接続します。

```
connect /@sales1 as sysdba
```

例3-17 サービス名を使用したユーザーとしての接続

次の例では、ユーザーsalesadminとして、ネット・サービス名sales1によって指定されたデータベース・サービスにリモート接続します。データベース・セッションは、rev21エディションで開始されます。SQL*Plusによって、ユーザーsalesadminのパスワードの入力が求められます。

```
connect salesadmin@sales1 edition=rev21
```

注意:



SYSDBA 権限を持つユーザーとしてデータベースに接続しているときに問題が発生した場合は、My Oracle Support ノート 69642.1、233223.1、18089.1 および 747456.1 を参照してください。

親トピック: [SQL*Plusを使用したデータベースへの接続](#)

4 Oracle Databaseの移行

フル・トランスポートابل・エクスポート/インポート機能を使用して、データベース全体を別のOracle Databaseインスタンスにコピーします。

- [キャラクタ・セットの互換性の確認](#)

ソース・データベースおよび宛先データベースでこれらのコマンドを実行し、互換性のあるキャラクタ・セットを見つけます。

- [エクスポート・ダンプ・ファイルを使用したデータベースのトランスポート](#)

エクスポート・ダンプ・ファイルを使用してデータベースをトランスポートできます。

キャラクタ・セットの互換性の確認

ソース・データベースおよび宛先データベースでこれらのコマンドを実行し、互換性のあるキャラクタ・セットを見つけます。

```
SQL> show parameter CHARACTER
```

NAME	TYPE	VALUE
-----	-----	-----
nls_numeric_characters	string	

```
SQL> select * from database_properties where PROPERTY_NAME in ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');
```

PROPERTY_NAME	PROPERTY_VALUE	DESCRIPTION
-----	-----	-----
NLS_NCHAR_CHARACTERSET	AL16UTF16	NCHAR Character set
NLS_CHARACTERSET	WE8MSWIN1252	Character set

```
SQL> select * from nls_database_parameters where parameter like '%SET%';
```

PROPERTY_NAME	VALUE
-----	-----
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_CHARACTERSET	WE8MSWIN1252

親トピック: [Oracle Databaseの移行](#)

エクスポート・ダンプ・ファイルを使用したデータベースのトランスポート

エクスポート・ダンプ・ファイルを使用してデータベースをトランスポートできます。

次のタスクのリストでは、エクスポート・ダンプ・ファイルを使用したデータベースのトランスポート処理の概要を示します。各タスクの詳細は、後続の例で示します。

1. ソース・データベースで、各ユーザー定義表領域を読み取り専用モードに構成し、データベースをエクスポートします。

次のパラメータが指定された値に設定されていることを確認します。

- TRANSPORTABLE=ALWAYS
- FULL=Y

ソース・データベースがOracle Database 11gデータベース(11.2.0.3以降)である場合は、VERSIONパラメータを12以上に設定する必要があります。

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含む表が格納された表領域が含まれている場合は、ENCRYPTION_PWD_PROMPT=YESを指定するか、ENCRYPTION_PASSWORDパラメータを指定する必要があります。

エクスポート・ダンプ・ファイルには、ユーザー定義表領域に格納されたオブジェクトのメタデータ、および管理表領域(SYSTEMやSYSAUXなど)に格納されたユーザー定義オブジェクトのメタデータとデータの両方が含まれています。

2. エクスポート・ダンプ・ファイルをトランスポートします。

エクスポート・ダンプ・ファイルをターゲット・データベースへアクセス可能な場所にコピーします。

3. データベース内のユーザー定義表領域すべてのデータファイルをトランスポートします。

データファイルをターゲット・データベースへアクセス可能な場所にコピーします。

ソース・プラットフォームとターゲット・プラットフォームが異なる場合は、V\$TRANSPORTABLE_PLATFORMビューに対して問合せを実行して、各プラットフォームのendian形式をチェックできます([「プラットフォーム間でのデータのトランスポート」](#)を参照)。

ソース・プラットフォームのendian形式がターゲット・プラットフォームのendian形式と異なる場合は、次のいずれかの方法を使用してデータファイルを変換します。

- 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、データファイルを転送します。これらのプロシージャを使用すると、データファイルがターゲット・プラットフォームのendian形式に自動的に変換されます。
- RMAN CONVERTコマンドを使用して、データファイルをターゲット・プラットフォームのendian形式に変換します。

注意:



UNDO セグメントを含むデータファイルでは、異なる endian 形式間でのデータファイルの変換はサポートされていません。

詳細は、[「プラットフォーム間でのデータの変換」](#)を参照してください。

4. (オプション)ソース・データベースでユーザー定義表領域を読み取り/書き込みモードに戻します。

5. ターゲット・データベースで、データベースをインポートします。

インポートが完了したとき、ユーザー定義表領域は読み取り/書き込みモードになります。

例

データベースをトランスポートするためのタスクは、この例で詳しく説明します。この例では、ソース・プラットフォームがSolarisで、ターゲット・プラットフォームがMicrosoft Windowsであることを前提としています。

また、ソース・プラットフォームには次のデータファイルと表領域があるとします。

表領域	タイプ	データファイル
sales	ユーザー定義	/u01/app/oracle/oradata/mydb/sales01.dbf
customers	ユーザー定義	/u01/app/oracle/oradata/mydb/cust01.dbf
employees	ユーザー定義	/u01/app/oracle/oradata/mydb/emp01.dbf
SYSTEM	管理	/u01/app/oracle/oradata/mydb/system01.dbf
SYSAUX	管理	/u01/app/oracle/oradata/mydb/sysaux01.dbf

この例では、さらに次のことを想定しています。

- ターゲット・データベースは、ソース・データベースからデータを移入する新しいデータベースです。ソース・データベースの名前はmydbです。
- ソース・データベースとターゲット・データベースの両方がOracle Database 19cデータベースです。

エクスポート・ダンプ・ファイルを使用してデータベースをトランスポートするには、次のタスクを実行します。

タスク 1 エクスポート・ダンプ・ファイルの作成

次のステップを実行して、エクスポート・ダンプ・ファイルを生成します。

1. SQL*Plus を起動し、管理者として、あるいは ALTER TABLESPACE または MANAGE TABLESPACE システム権限を持つユーザーとしてデータベースに接続します。
2. データベース内のすべてのユーザー定義表領域を読み取り専用にします。

```
ALTER TABLESPACE sales READ ONLY;  
  
ALTER TABLESPACE customers READ ONLY;  
  
ALTER TABLESPACE employees READ ONLY;
```

3. DATAPUMP_EXP_FULL_DATABASE ロールを持つユーザーとしてデータ・ポンプ・エクスポート・ユーティリティを起動し、フル・トランスポート・エクスポート/インポート・オプションを指定します。

```
SQL> HOST

$ expdp user_name full=y dumpfile=expdat.dmp directory=data_pump_dir
transportable=always logfile=export.log

Password: password
```

トランスポータブル・オプションを使用するかどうかを設定する TRANSPORTABLE=ALWAYS を常に指定する必要があります。

この例では、次のデータ・ポンプ・パラメータを指定します。

- FULL パラメータでは、データベース全体をエクスポートすることを指定します。
- DUMPFILE パラメータでは、作成する構造情報エクスポート・ダンプ・ファイルの名前を expdat.dmp と指定します。
- DIRECTORY パラメータでは、オペレーティング・システムまたは Oracle Automatic Storage Management のダンプ・ファイルの場所を示すディレクトリ・オブジェクトを指定します。DIRECTORY オブジェクトはデータ・ポンプを起動する前に作成し、ディレクトリに対する READ および WRITE オブジェクト権限をエクスポート・ユーティリティを実行するユーザーに付与する必要があります。CREATE DIRECTORY コマンドの詳細は、[『Oracle Database SQL 言語リファレンス』](#)を参照してください。

非 CDB で、ディレクトリ・オブジェクト DATA_PUMP_DIR が自動的に作成されます。このディレクトリへの読み取りおよび書き込みアクセス権が DBA ロールに(したがって、ユーザーSYS および SYSTEM に)自動的に付与されます。

ただし、ディレクトリ・オブジェクト DATA_PUMP_DIR は、PDB では自動的に作成されません。このため、PDB にインポートする場合は、PDB にディレクトリ・オブジェクトを作成し、データ・ポンプを実行するときにそのディレクトリ・オブジェクトを指定します。

関連項目:

- DIRECTORY パラメータを省略する場合のデフォルト・ディレクトリの詳細は、[『Oracle Database ユーティリティ』](#)を参照してください。
- PDB の詳細は、[『Oracle Multitenant 管理者ガイド』](#)を参照してください
- LOGFILE パラメータでは、エクスポート・ユーティリティによって書き込まれるログ・ファイルのファイル名を指定します。この例では、ログ・ファイルの書き込み先はダンプ・ファイルと同じディレクトリですが、ログ・ファイルは別の場所へ書き込むことができます。

Oracle Database 11g リリース 2 (11.2.0.3)または Oracle Database 11g 以上のデータベースでフル・トランスポータブル・エクスポートを実行するには、次の例のように、VERSION パラメータを使用します。

```
expdp user_name full=y dumpfile=expdat.dmp directory=data_pump_dir
transportable=always version=12 logfile=export.log
```

フル・トランスポータブル・インポートは、Oracle Database 12c 以降のデータベースでのみサポートされます。

注意:



この例では、データ・ポンプ・ユーティリティを使用してエクスポートするのは、ユーザー定義表領域のデータ・ディクショナリの構造情報(メタデータ)のみです。実際のデータは管理表領域(SYSTEM および SYSAUX)についてのみアンロードされるため、この操作は大規模なユーザー定義表領域の場合にも比較的短時間で完了します。

4. ログ・ファイルでエラーを確認し、ターゲット・データベースにトランスポートする必要があるダンプ・ファイルとデータファイルを書き留めます。expdp により、これらのファイルの名前とパスが次のようなメッセージに出力されます。

```
*****
Dump file set for SYSTEM.SYS_EXPORT_TRANSPORTABLE_01 is:
  /u01/app/oracle/admin/mydb/dpdump/expdat.dmp
*****
Datafiles required for transportable tablespace SALES:
  /u01/app/oracle/oradata/mydb/sales01.dbf
Datafiles required for transportable tablespace CUSTOMERS:
  /u01/app/oracle/oradata/mydb/cust01.dbf
Datafiles required for transportable tablespace EMPLOYEES:
  /u01/app/oracle/oradata/mydb/emp01.dbf
```

5. 完了した後、終了して SQL *Plus に戻ります。

```
$ exit
```

関連項目:

データ・ポンプ・ユーティリティの使用方法は、『[Oracle Database ユーティリティ](#)』を参照してください

タスク 2 エクスポート・ダンプ・ファイルのトランスポート

ダンプ・ファイルを、DATA_PUMP_DIR ディレクトリ・オブジェクトで指し示されているディレクトリ、または他の任意のディレクトリにトランスポートします。新しい場所はターゲット・データベースへアクセス可能であることが必要です。

ターゲット・データベースで、次の問合せを実行して DATA_PUMP_DIR の場所を確認します。

```
SELECT * FROM DBA_DIRECTORIES WHERE DIRECTORY_NAME = 'DATA_PUMP_DIR' ;
```

OWNER	DIRECTORY_NAME	DIRECTORY_PATH
SYS	DATA_PUMP_DIR	C:\app\orauser\admin\orawin\dpdump

タスク 3 ユーザー定義表領域のデータファイルのトランスポート

データベースのユーザー定義表領域のデータファイルをターゲット・データベースへアクセス可能な場所にトランスポートします。

この例では、次のデータファイルをソース・データベースからターゲット・データベースに転送します。

- sales01.dbf
- cust01.dbf
- emp01.dbf

ソース・プラットフォームとは異なるプラットフォームにデータベースをトランスポートする場合は、ソースおよびターゲット・プラットフォームの両方でプラットフォーム間のデータベース・トランスポートがサポートされているかどうかを確認し、それぞれのプラットフォームの endianness を判別します。両方のプラットフォームの endianness が同じ場合、変換は必要ありません。同じでない場合は、ソース・データベースまたはターゲット・データベースのどちらかでデータベースの各表領域を変換する必要があります。

データベースを異なるプラットフォームにトランスポートする場合は、各プラットフォームで次の問合せを実行できます。問合せで行が返される場合、そのプラットフォームではプラットフォーム間の表領域トランスポートがサポートされています。

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

ソース・プラットフォームでの問合せの結果は次のとおりです。

PLATFORM_NAME	ENDIAN_FORMAT
Solaris[tm] OE (32-bit)	Big

ターゲット・プラットフォームからの問合せの結果は、次のとおりです。

PLATFORM_NAME	ENDIAN_FORMAT
---------------	---------------

この例では、endian 形式が異なることがわかります。したがって、この場合、データベースをトランスポートするには変換が必要です。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、データファイルを転送します。これらのプロシージャを使用すると、データファイルがターゲット・プラットフォームの endian 形式に自動的に変換されます。データファイルを、ターゲット・データベースの既存データファイルの場所にトランスポートします。UNIX および Linux プラットフォームでは、この場所は通常、/u01/app/oracle/oradata/dbname/ または +DISKGROUP/dbname/datafile/ です。また、RMAN CONVERT コマンドを使用してデータファイルを変換できます。詳細は、[「プラットフォーム間でのデータの変換」](#)を参照してください。

注意:



表領域の endianness を変換する必要がない場合は、任意のファイル転送方法を使用してファイルを送ることができます。

タスク 4 (オプション)表領域を読み取り/書き込みモードに戻す

次のように、トランスポートした表領域をソース・データベースで再び読み取り/書き込みモードにします。

```
ALTER TABLESPACE sales READ WRITE;  
ALTER TABLESPACE customers READ WRITE;  
ALTER TABLESPACE employees READ WRITE;
```

インポート・プロセスが成功したことを先に確認するために、このタスクを延期することができます。

タスク 5 ターゲット・データベースでのデータベースのインポート

DATAPUMP_IMP_FULL_DATABASE ロールを持つユーザーとしてデータ・ポンプ・インポート・ユーティリティを起動し、フル・トランスポート・エクスポート/インポート・オプションを指定します。

```
impdp user_name full=Y dumpfile=expdat.dmp directory=data_pump_dir  
transport_datafiles=  
  '/u01/app/oracle/oradata/mydb/sales01.dbf',  
  '/u01/app/oracle/oradata/mydb/cust01.dbf',  
  '/u01/app/oracle/oradata/mydb/emp01.dbf'  
logfile=import.log
```

Password: *password*

この例では、次のデータ・ポンプ・パラメータを指定します。

- FULL パラメータでは、データベース全体を FULL モードでインポートすることを指定します。

- DUMPFIL パラメータでは、インポートされるユーザー定義表領域のメタデータおよび管理表領域のメタデータとデータの両方が含まれるエクスポート・ファイルを指定します。
- DIRECTORY パラメータでは、エクスポート・ダンプ・ファイルの場所を識別するディレクトリ・オブジェクトを指定します。DIRECTORY オブジェクトはデータ・ポンプを起動する前に作成し、ディレクトリに対する READ および WRITE オブジェクト権限をインポート・ユーティリティを実行するユーザーに付与する必要があります。CREATE DIRECTORY コマンドの詳細は、『[Oracle Database SQL 言語リファレンス](#)』を参照してください。

非 CDB で、ディレクトリ・オブジェクト DATA_PUMP_DIR が自動的に作成されます。このディレクトリへの読取りおよび書き込みアクセス権が DBA ロールに(したがって、ユーザーSYS および SYSTEM に)自動的に付与されます。

ただし、ディレクトリ・オブジェクト DATA_PUMP_DIR は、PDB では自動的に作成されません。このため、PDB にインポートする場合は、PDB にディレクトリ・オブジェクトを作成し、データ・ポンプを実行するときそのディレクトリ・オブジェクトを指定します。

関連項目:

- DIRECTORY パラメータを省略する場合のデフォルト・ディレクトリの詳細は、『[Oracle Database ユーティリティ](#)』を参照してください。
- PDB の詳細は、『[Oracle Multitenant 管理者ガイド](#)』を参照してください
- TRANSPORT_DATAFILES パラメータによって、インポートするすべてのデータファイルを識別します。

多くのデータファイルがある場合は、PARFILE パラメータで指定されたパラメータ・ファイルで TRANSPORT_DATAFILES パラメータを複数回指定できます。

- LOGFILE パラメータでは、インポート・ユーティリティによって書き込まれるログ・ファイルのファイル名を指定します。この例では、ログ・ファイルの書き込み先はダンプ・ファイルの読取り元と同じディレクトリですが、ログ・ファイルは別の場所へ書き込むことができます。

この文が正常に実行された後、インポート・ログ・ファイルをチェックして、予期しないエラーが発生していないことを確認します。

多数のデータファイルを扱う場合、データファイル名のリストを文の行で指定することは煩雑です。また、文の行制限を超える場合もあります。このような場合には、インポート・パラメータ・ファイルを使用できます。たとえば、次のようにしてデータ・ポンプ・インポート・ユーティリティを起動できます。

```
impdp user_name parfile='par.f'
```

たとえば、par.f には次の行が含まれる場合があります。

```
FULL=Y  
DUMPFILE=expdat.dmp  
DIRECTORY=data_pump_dir  
TRANSPORT_DATAFILES=  
'/u01/app/oracle/oradata/mydb/sales01.dbf',  
'/u01/app/oracle/oradata/mydb/cust01.dbf',  
'/u01/app/oracle/oradata/mydb/emp01.dbf'  
LOGFILE=import.log
```

注意:

- インポート中に、メタデータのロードのためにユーザー定義表領域を一時的に読取り/書込みにする場合があります。インポート中にデータに対するユーザー変更が行われないことを確認してください。インポートが正常に完了したとき、すべてのユーザー定義表領域は読取り/書込みになります。
- ネットワーク・データベース・インポートを実行する場合は、TRANSPORTABLE パラメータを always に設定する必要があります。
- CDB 内の PDB にインポートする場合は、ユーザー名の後に PDB の接続識別子を指定します。たとえば、PDB の接続識別子が hrpdb である場合は、Oracle Data Pump インポート・ユーティリティを実行するとき、次のように入力します。

```
impdp user_name@hrpdb ...
```

関連項目:

- インポート・ユーティリティの使用方法は、『[Oracle Database ユーティリティ](#)』を参照してください。
- 『[Oracle Multitenant 管理者ガイド](#)』

親トピック: [Oracle Databaseの移行](#)