

# Oracle® Database

## AutoUpgrade による同じオペレーティング・システム での非 CDB の PDB へのアップグレードと変換

### 19c

F16178-04(原本部品番号:F10903-06)

2022年7月

# タイトルおよび著作権情報

Oracle Database同じオペレーティング・システムでの非CDBのPDBへのアップグレードと変換のためのAutoUpgradeの使用, 19c

F16178-04

[Copyright ©](#) 2018, 2021, Oracle and/or its affiliates.

原著者: Nirmal Kumar, Sunil Surabhi, Douglas Williams

協力著者: Padmaja Potineni, Rajesh Bhatiya, Prakash Jashnani, Mark Bauer

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

# 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 Oracle Databaseをアップグレードする前の互換性の確認](#)
  - [Oracle Databaseの互換性レベルの確認](#)
  - [Oracle DatabaseのCOMPATIBLE初期化パラメータの値](#)
- [2 Oracle Databaseのアップグレードの準備](#)
  - [新しいOracleホームへのOracleソフトウェアのインストール](#)
    - [アップグレード時のOracleホームの新しい場所の選択](#)
    - [単一インスタンス用の新しいOracle Databaseソフトウェアのインストール](#)
  - [Oracle Databaseをアップグレードする前のバックアップ計画の準備](#)
  - [Oracle Databaseのアップグレードを開始する前に完了するデータベースの準備作業](#)
    - [Oracle Databaseをアップグレードする場合のリリースの更新および要件](#)
    - [Oracle Databaseのアップグレード時のOracle Net Servicesに関する推奨事項](#)
    - [パスワードの大/小文字の区別とアップグレードについて](#)
    - [大/小文字を区別しないパスワード・バージョンを使用しているアカウントがあるかどうかの確認](#)
    - [読取り専用の表領域を使用したアップグレードの実行](#)
  - [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)
    - [Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード](#)
    - [Oracle Database Vaultが関連する一般的なアップグレード・シナリオ](#)
  - [AutoUpgrade処理モードを実行するための準備](#)
  - [AutoUpgradeでのアップグレード前情報チェック](#)
  - [AutoUpgradeの構成ファイルの作成](#)
  - [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)
    - [defer\\_standby\\_log\\_shipping](#)
    - [drop\\_grp\\_after\\_upgrade](#)
    - [enable\\_local\\_undo](#)
    - [manage\\_network\\_files](#)
    - [remove\\_underscore\\_parameters](#)
    - [restoration](#)
    - [target\\_base](#)
    - [target\\_home](#)
    - [target\\_version](#)
  - [AutoUpgrade構成ファイルのローカル・パラメータ](#)
    - [add\\_after\\_upgrade\\_pfile](#)
    - [add\\_during\\_upgrade\\_pfile](#)
    - [after\\_action](#)
    - [before\\_action](#)
    - [catctl\\_options](#)
    - [checklist](#)

- [del\\_after\\_upgrade\\_pfile](#)
- [del\\_during\\_upgrade\\_pfile](#)
- [env](#)
- [keep\\_source\\_pdb](#)
- [log\\_dir](#)
- [manage\\_standbys\\_clause](#)
- [pdb](#)
- [raise\\_compatible](#)
- [remove\\_underscore\\_parameters](#)
- [restoration](#)
- [revert\\_after\\_action](#)
- [revert\\_before\\_action](#)
- [run\\_utlpr](#)
- [sid](#)
- [skip\\_tde\\_key\\_import](#)
- [source\\_base](#)
- [source\\_dblink](#)
- [source\\_home](#)
- [source\\_ldap\\_admin\\_dir](#)
- [source\\_tns\\_admin\\_dir](#)
- [start\\_time](#)
- [target\\_base](#)
- [target\\_cdb](#)
- [target\\_pdb\\_copy\\_option=file\\_name\\_convert](#)
- [target\\_pdb\\_name](#)
- [target\\_ldap\\_admin\\_dir](#)
- [target\\_tns\\_admin\\_dir](#)
- [timezone\\_upg](#)
- [tune\\_setting](#)
- [upgrade\\_node](#)
- [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)
  - [add\\_after\\_upgrade\\_pfile](#)
  - [add\\_during\\_upgrade\\_pfile](#)
  - [after\\_action](#)
  - [autoupg\\_log\\_dir](#)
  - [before\\_action](#)
  - [catctl\\_options](#)
  - [del\\_after\\_upgrade\\_pfile](#)
  - [del\\_during\\_upgrade\\_pfile](#)
  - [drop\\_grp\\_after\\_upgrade](#)
  - [keystore](#)
  - [raise\\_compatible](#)
  - [target\\_base](#)

- [target\\_home](#)
- [target\\_version](#)
- [upgradexml](#)
- [AutoUpgradeを使用した非CDBからPDBへのアップグレードの理解](#)
- [非CDBからPDBへのアップグレードのガイドラインおよび例](#)
- [AutoUpgradeを使用した切断/接続アップグレードの理解](#)
- [AutoUpgradeの非CDBからPDBの構成ファイルの例](#)
- [3 AutoUpgradeによる非CDBからPDBへのアップグレードおよび変換](#)
  - [同じサーバー上にソースおよびターゲットのデータベース・ホームがあるAutoUpgrade\(通常\)](#)
  - [異なるサーバー上にソースおよびターゲット・データベース・ホームがあるAutoUpgrade](#)
- [4 Oracle Databaseのアップグレード後の作業](#)
  - [アップグレード後の状態ツールを使用したアップグレードの確認](#)
  - [Oracle Databaseのアップグレード後に必要な作業](#)
    - [手動アップグレード後のLinuxおよびUnixシステム上での環境変数の設定](#)
    - [PL/SQLパッケージおよび依存プロセスの確認](#)
    - [Oracle Databaseのアップグレード後のDBMS\\_STATSパッケージで作成された統計表のアップグレード](#)
    - [Oracle XML DBに対するFTPとHTTPのポートおよびHTTP認証の構成](#)
    - [Oracle Databaseのアップグレード後のOracle Textが提供するナレッジ・ベースのインストール](#)
    - [読取り専用OracleホームでのDEMOディレクトリの置換](#)
    - [外部ネットワーク・サービスへのアクセス制御リスト\(ACL\)の構成](#)
    - [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)
      - [Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード](#)
      - [Oracle Database Vaultが関連する一般的なアップグレード・シナリオ](#)
    - [SQLNET.ALLOWED\\_LOGON\\_VERSIONパラメータの動作の確認](#)
  - [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)
    - [データベースのバックアップ](#)
    - [AutoUpgradeアップグレード後チェックの実行](#)
    - [DBMS\\_STATSを使用した固定オブジェクトの統計の再収集](#)
    - [パスワードのリセットによる大/小文字区別の強制](#)
    - [10Gパスワード・バージョンを使用するユーザーのパスワードの確認と再設定](#)
    - [Oracle Grid Infrastructure、Oracle ASMおよびOracle Clusterwareの理解](#)
    - [Oracle Grid InfrastructureのインストールとアップグレードおよびOracle ASM](#)
    - [新機能の適宜追加](#)
    - [必要な新しい管理手順の作成](#)
    - [ロールバック・セグメントから自動UNDOモードへの移行](#)
    - [LONGデータ型からLOBデータ型への表の移行](#)
    - [アップグレードしたOracle Databaseの統合監査の使用への移行](#)
      - [Oracle Databaseでの統合監査の移行プロセスの理解](#)
      - [Oracle Databaseでの統合監査への移行](#)
      - [統合監査への移行後の古い監査レコードの管理について](#)
      - [統合監査機能の削除](#)
      - [統合監査を使用しない場合のドキュメント参照の取得](#)

- [再構築対象のOracle Text索引の識別](#)
- [DBMS\\_SCHEDULERジョブの削除および再作成](#)
- [アップグレード後の統合監査レコードの転送](#)
  - [アップグレード後の監査レコードの転送について](#)
  - [アップグレード後の監査レコードの転送](#)
- [Oracle Databaseのアップグレード後のリカバリ・カタログ・アップグレードについて](#)
- [アップグレードされたデータベースでの無効なリリース更新バグ修正の有効化](#)
- [アップグレードした本番Oracle Databaseのテストについて](#)
- [Oracle Databaseのアップグレード後のタイムゾーン・ファイルのバージョンのアップグレード](#)

# はじめに

このガイドは、特定のユースケース・シナリオを完了する助けとなるよう収集されたOracle Databaseユーザー支援ドキュメントのトピック集です。

- [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、AutoUpgradeユーティリティを使用して、以前のリリースの非CDBを新しいリリースのOracle DatabaseのPDBにアップグレードおよび変換することが容易になります。

このシナリオの前提条件

- マルチテナント・コンテナ・データベース(CDB)デプロイメントを使用して、サーバーに新しいOracle Databaseリリースをインストールしておきます

データベースをバックアップすることをお勧めします。

このシナリオの概要

- Oracle Databaseをアップグレードする前の互換性の確認。以前のリリースがこのアップグレード・シナリオと互換性があることを確認します。
- Oracle Databaseのアップグレードの準備。このアップグレード・シナリオのステップを確認し、準備作業を完了します。
- Oracle Databaseのアップグレード。AutoUpgradeユーティリティを使用して、データベースを非CDBからマルチテナントOracle Database 19cのPDBにアップグレードおよび変換します。
- Oracle Databaseのアップグレード後の作業。このアップグレード後タスクの基本リストを完了します。

前述のステップはそれぞれ、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 Oracle Databaseをアップグレードする前の互換性の確認

Oracle Databaseをアップグレードする前に、Oracle Databaseサーバーのアップグレード互換性マトリックスを確認します。

- [Oracle Databaseの互換性レベルの確認](#)  
このSQL問合せを使用して、データベースに設定されているCOMPATIBLE初期化パラメータの値を確認します。
- [Oracle DatabaseのCOMPATIBLE初期化パラメータの値](#)  
Oracle Database 19cのCOMPATIBLE初期化パラメータのデフォルト値、最小値および最大値を確認してください。

## Oracle Databaseの互換性レベルの確認

このSQL問合せを使用して、データベースに設定されているCOMPATIBLE初期化パラメータの値を確認します。

```
SQL> SELECT name, value FROM v$parameter  
WHERE name = 'compatible';
```

## Oracle DatabaseのCOMPATIBLE初期化パラメータの値

Oracle Database 19cのCOMPATIBLE初期化パラメータのデフォルト値、最小値および最大値を確認してください。

COMPATIBLEパラメータのデフォルト値と最小値

CDBまたは非CDBインスタンスの場合、RUまたはRURのCOMPATIBLEパラメータを変更しないでください。次の表に、直接アップグレードがサポートされている以前のリリースと、Oracle Database 19cのCOMPATIBLEパラメータのデフォルト値と最小値を示します。

COMPATIBLE初期化パラメータの設定がOracle Database 19cへの直接アップグレードには小さすぎる

ソースのOracle DatabaseのCOMPATIBLEパラメータがOracle Database 19cへのアップグレードには小さすぎる値に設定されている場合、ソースのOracle DatabaseリリースのCOMPATIBLE設定を少なくともOracle Database 19cアップグレードでサポートされている最小値に増やす必要があります。Oracle Databaseの最小のCOMPATIBLE設定は11.2.0です。

たとえば、ソースのOracle DatabaseリリースがOracle DatabaseリリースがOracle Database 11gリリース2 (11.2.0.4)であるが、互換性のあるパラメータが10.0.0に設定されている場合(COMPATIBLE = 10.0.0)、アップグレードを開始する前に、互換性パラメータをアップグレードでサポートされている最小値(COMPATIBLE = 11.2.0)に変更する必要があります。

ノート:

COMPATIBLE パラメータを変更した後に、データベースを再起動する必要があります。

表1-1 COMPATIBLE初期化パラメータ

<b>Oracle Databaseのリリース</b>	<b>デフォルト値</b>	<b>最小値</b>
Oracle Database 19c	19.0.0	11.2.0
Oracle Database 12c リリース 2 (12.2)	12.2.0	11.2.0
Oracle Database 12c リリース 1(12.1)	12.0.0	11.0.0
Oracle Database 11g リリース 2(11.2)	11.2.0	10.0.0

## 2 Oracle Databaseのアップグレードの準備

Oracle Databaseをアップグレードする前に、新機能を確認し、手順を実行してデータベースをアップグレードに備えます。



ノート:

特に、アップグレード処理をテストし、バックアップ計画を準備することをお勧めします。

- [新しいOracleホームへのOracleソフトウェアのインストール](#)  
ターゲットのOracleホームの新しい場所を選択してから、単一インスタンス用の新しいOracle Databaseソフトウェアをインストールします。
- [Oracle Databaseをアップグレードする前のバックアップ計画の準備](#)  
アップグレードの成功を保証する適切なバックアップ計画を設計して実行する必要があります。
- [Oracle Databaseのアップグレードを開始する前に完了するデータベースの準備作業](#)  
Oracle Databaseのアップグレードを開始する前に、これらのデータベースの準備作業を完了していることを確認してください。
- [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)  
ターゲットのデータベース・リリースによっては、Oracle Database Vaultを無効にしないとOracle Databaseのアップグレードを完了できない場合があります。
- [AutoUpgrade処理モードを実行するための準備](#)  
AutoUpgrade処理モードを実行する前に、準備を完了している必要があります。
- [AutoUpgradeを使用したアップグレード前情報のチェック](#)  
Oracle Databaseをアップグレードする前に実行する必要があるタスクのチェックリストを取得するには、analyzeモードでAutoUpgradeユーティリティ(autoupgrade.jar)を実行します。
- [AutoUpgradeの構成ファイルの作成](#)  
AutoUpgradeを使用してアップグレードを完了するには、まず、新しいリリースのOracleホームからAutoUpgradeで構成ファイルを作成します。
- [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)  
ローカルに変更可能なグローバル・パラメータは、グローバルにも、必要に応じてローカルにも設定されるパラメータです。これにより、AutoUpgradeジョブの処理をより適切に制御できます。
- [AutoUpgrade構成ファイルのローカル・パラメータ](#)  
AutoUpgradeユーティリティのアップグレード用に特定のOracle Databaseの情報を構成するには、AutoUpgradeローカル・パラメータに情報を指定します。
- [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)  
構成ファイルで扱われるすべてのOracle Databaseアップグレードのパラメータに対するデフォルトの動作を指定するには、オプションのAutoUpgradeグローバル・パラメータを使用できます。
- [AutoUpgradeを使用した非CDBからPDBへのアップグレードの理解](#)  
1回の操作で非CDBを新しいCDBのPDBにアップグレードおよび変換するか、非CDBデータベースを既存のCDBのPDBにアップグレードしてから変換できます。
- [非CDBからPDBへのアップグレードのガイドラインおよび例](#)  
変換前に、データファイルおよびデータベースをバックアップし、ソースのOracle Databaseリリースに関するガイドラインに従います。
- [AutoUpgradeを使用した切断/接続のアップグレードの理解](#)

AutoUpgradeでは、以前のリリースのソース・コンテナ・データベース(CDB)からプラグブル・データベース(PDB)を切断し、新しいリリースのターゲットCDBに接続して、PDBをターゲットCDBリリースにアップグレードするために必要なすべてのステップを完了できます。

- [AutoUpgradeの非CDBからPDBの構成ファイルの例](#)

これらの例を使用して、AutoUpgradeの独自のOracle Databaseアップグレード構成ファイルを変更する方法を理解します。

## AutoUpgradeでのアップグレード前情報チェック

Oracle Databaseをアップグレードする前に完了する必要があるタスクのチェックリストを取得するには、analyzeモードでAutoUpgradeユーティリティ(`autoupgrade.jar`)を実行します。

Oracleでは、Oracle Databaseをアップグレードする前に、AutoUpgradeの最新リリースをダウンロードしてanalyzeモードで実行することをお勧めします。AutoUpgradeでは、アップグレードを開始する前に対処する必要がある問題を識別できます。場合によっては、一部の問題を解決できるスクリプトをAutoUpgradeで生成することもできます。

ヒント:



Mike Dietrich のアップグレード・ブログで、アップグレードの準備に役立つヒントおよび推奨事項について確認することを検討してください。Oracle Database 19c の場合、My Oracle Support のノート Oracle Database 19c の重要な推奨個別パッチ(ドキュメント ID 555.1)を確認して、Oracle Database 19c の最も重要なパッチ・バンドルに関する情報を取得します。

### 関連項目

- [My Oracle Support AutoUpgradeツール\(ドキュメントID 2485457.1\)](#)
- [My Oracle Support Oracle Database 19cの重要な推奨個別パッチ\(ドキュメントID 555.1\)](#)
- [Upgrade your Database – NOW! Mike Dietrich's Oracle Database Upgrade Blog](#)

親トピック: [Oracle Databaseのアップグレードの準備](#)

## AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ

ローカルに変更可能なグローバル・パラメータは、グローバルにも、必要に応じてローカルにも設定されるパラメータです。これにより、AutoUpgradeジョブの処理をより適切に制御できます。

使用上のノート

ローカルに変更可能なグローバル・パラメータは必須パラメータです。これらのパラメータは、AutoUpgrade構成ファイルでグローバルまたはローカルのいずれかに定義する必要があります。ローカルで変更可能なグローバル・パラメータでは、接頭辞`global`を使用して、AutoUpgrade構成ファイルのすべてのジョブのグローバル・パラメータとして値を設定できますが、同じ構成ファイル内の特定のジョブのローカル・ジョブ接頭辞で同じパラメータを再設定することもできます。ローカルで変更可能なグローバル・パラメータのみを、各AutoUpgradeジョブのローカル・パラメータとして設定することもできます。



ノート:

これらのパラメータは、My Oracle Support からダウンロードできる AutoUpgrade の最新バージョンで使用できます。

ローカルで変更可能なグローバル・パラメータがグローバル接頭辞とローカル・ジョブ接頭辞の両方で設定されている場合、ローカルに変更されたパラメータ値は、パラメータで使用する接頭辞によって識別されるジョブのグローバル・パラメータ値をオーバーライドします。使用する構文の形式は、`global.target_home=Global target Oracle home`および `database.target_home=local target Oracle home`です。

例:

AutoUpgrade構成ファイルでは、必須パラメータ`target_home`が1つのOracleホーム・パスにグローバルに設定されます。ただし、構成ファイルでは、同じパラメータが別のOracleホーム・パスに対してローカルに設定されます。AutoUpgradeは構成ファイル内のジョブを処理すると、接頭辞`upgrade3`で定義されたジョブの`target_home`にローカルに定義されたパスを使用し、グローバル・パラメータ設定をオーバーライドします。

```
global.target_home=/u01/app/oracle/21.0.0/dbhome01
upgrade3.target_home=/u03/app/oracle3/12.2.0.1/dbhome3
```

表2-1 AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ

パラメータ	説明
<code>defer_standby_log_shipping</code>	(オプション)プライマリ・データベースにフィジカル・スタンバイ・データベースがある場合、アップグレード前にプライマリ・データベースからスタンバイ・データベースへのログの送信を遅延します。Autoupgradeによりログ送信が遅延すると、ログ送信が遅延され、アップグレードが正常に完了した後にプライマリ・データベースからセカンダリ・データベースへの送信ログを再度有効にする必要があることが通知されます。デフォルトのオプションは No です。デフォルトを Yes に変更すると、ログ送信は遅延されます。
<code>drop_grp_after_upgrade</code>	(オプション)データベースのアップグレード後に、保証付きリストア・ポイント(GRP)を削除します。このオプションを選択した場合、アップグレードが正常に完了した後に GRP が削除されます。  オプション:  [yes   no]  デフォルト値は no です。  例:  <code>global.drop_grp_after_upgrade=yes</code> <code>sales.drop_grp_after_upgrade=yes</code>

パラメータ	説明
enable_local_undo	<p>(オプション) CDB のアップグレードの場合、ALTER DATABASE LOCAL UNDO ON; 文を実行して、CDB\$ROOT のアップグレード前に LOCAL UNDO を有効にする必要があるかどうかを指定します。使用可能な値は[YES   NO]です。デフォルト値は NO です。</p> <p>ローカル UNDO を最初に有効にすると、PDB\$SEED 内の UNDO 表領域のサイズは、CDB\$ROOT 内の UNDO 表領域のサイズの係数として決定されます。デフォルトは、UNDO 表領域サイズの 30%です。CDB 内の他のすべての PDB は、PDB\$SEED からこのプロパティを継承します。新しい UNDO 表領域を割り当てるのに十分な領域があることを確認します。</p>
manage_network_files	<p>アップグレード中にネットワーク・ファイルを処理するかどうかを指定します。</p> <p>オプション:</p> <p>[FULL SKIP IGNORE_READ_ONLY]</p> <p>FULL: (デフォルト)ネットワーク・ファイルのターゲット Oracle ホームへのコピーおよびマージ中に発生したすべての例外を呼び出します。</p> <p>SKIP: アップグレード後にネットワーク・ファイルを処理しません。</p> <p>IGNORE_READ_ONLY: ネットワーク・ファイルをコピーおよびマージしようとしたが、ターゲット・ファイルが読取り専用の場合、アップグレード中に例外は発生しません。</p> <p>oranfstab、ldap.ora、tnsnames.ora、sqlnet.ora および listener.ora の各ネットワーク・ファイルが処理されます。</p>
remove_underscore_parameters	<p>(オプション)構成ファイル内のすべての Oracle Database について、アップグレード中およびアップグレード後に PF FILE ファイルからアンダースコア(非表示)パラメータを削除します。アンダースコアのパラメータは、Oracle サポートの指示がある場合のみ使用できます。</p> <p>オプション:</p> <p>[yes   no]</p> <p>デフォルト値は no です。</p> <p>例:</p> <p>global.remove_underscore_parameters=yes</p>

パラメータ	説明
restoration	<p>(オプション、Enterprise Edition でのみ使用可能)データベース・リストアの保証付きリストア・ポイント(GRP)を生成します。このオプションを選択すると、データベース・バックアップおよびデータベースのリストアの両方を DBA で手動で実行する必要があります。</p> <p>オプション:</p> <p>[yes   no]</p> <p>デフォルト値は yes。</p> <p>例:</p> <p>global.restoration=no</p> <p>Standard Edition はフラッシュバック・データベースをサポートしていないため、このオプションは Standard Edition では使用できません。データベースが Standard Edition の Oracle Database の場合、別個のフォールバック・メカニズムを持つようにする必要があります。</p>
target_version	<p>(オプション)AutoUpgrade でアップグレードを実行するターゲット・リリース・バージョンを指定します。AutoUpgrade では、このパラメータで指定したリリース・バージョン情報を使用して、アップグレード対象のターゲットの Oracle Database リリースに対して正しいチェックおよび修正が使用されるようにします。このパラメータの形式は、有効な Oracle バージョンのピリオド区切りの値です。</p> <p>有効な値</p> <ul style="list-style-type: none"> <li>● 12.2</li> <li>● 18</li> <li>● 19</li> <li>● 21</li> </ul> <p>このオプションは、ターゲット・ホームがシステムに存在しないか、ターゲット・ホームが 12.2 リリースの場合のみ必要です。それ以外の場合、AutoUpgrade はターゲット・リリース値を導出できます。</p> <p>例:</p> <p>global.target_version=18 employees.target_version=12.2</p>
target_home	<p>ターゲット Oracle ホーム(ORACLE_HOME)パスを指定します。</p>

パラメータ	説明
	<p>例:</p> <pre>global.target_home=/u01/app/oracle/21.0.0/dbhome01 sales4.target_home=/u04/app/oracle4/21.0.0/dbhome04</pre> <p>モードが ANALYZE または FIXUPS の場合、パラメータ target_home はオプションです。</p>
target_base	<p>(オプション)ターゲット Oracle ホームのターゲット ORACLE_BASE パスを指定します。</p> <p>例:</p> <pre>global.target_base=/u01/app/oracle sales4.target_base=/u04/app/oracle4</pre>

- [defer\\_standby\\_log\\_shipping](#)  
プライマリ・データベースから任意のスタンバイ・データベースへのログ送信を遅延します。すべてのログ・アーカイブの保存先(log\_archive\_dest\_n)が遅延に設定されます。
- [drop\\_grp\\_after\\_upgrade](#)  
データベースのアップグレード後に、保証付きリストア・ポイント(GRP)を削除します。
- [enable\\_local\\_undo](#)  
CDBのアップグレードの場合、CDB\$ROOTのアップグレード前にLOCAL UNDOを有効にする必要があるかどうかを指定します。
- [manage\\_network\\_files](#)  
アップグレード中にネットワーク・ファイルを処理するかどうかを指定します。
- [remove\\_underscore\\_parameters](#)  
構成ファイル内のすべてのOracle Databaseについて、アップグレード中およびアップグレード後にPF FILEファイルからアンダースコア(非表示)パラメータを削除します。
- [restoration](#)  
(Enterprise Editionでのみ使用可能)データベース・リストアの保証付きリストア・ポイント(GRP)を生成します。
- [target\\_base](#)  
ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。
- [target\\_home](#)  
(ターゲット・ホームがシステム上にない場合で、アップグレード・モードおよびデブロイ・モードの場合は必須です。分析および修正モードの場合はオプションです。)ターゲットOracleホーム(ORACLE\_HOME)パスを指定します。
- [target\\_version](#)  
(ターゲットOracleホームがシステム上にないか、リリース12.2の場合は必須) AutoUpgradeでアップグレードを実行するターゲット・リリース・バージョンを指定します。

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

## defer\_standby\_log\_shipping

プライマリ・データベースから任意のスタンバイ・データベースへのログ送信を遅延します。すべてのログ・アーカイブの保存先

(log\_archive\_dest\_n)が遅延に設定されます。

#### 使用上のノート

デフォルトでは、ログ送信はアップグレードの一部として実行されます。Autoupgradeによりログ送信が遅延すると、ログ送信が遅延され、アップグレードが正常に完了した後にプライマリ・データベースからセカンダリ・データベースへの送信ログを再度有効にする必要があることが通知されます。

#### ノート:



この構成ファイル・パラメータは、スタンバイ・データベースだけでなく、Oracle Zero Data Loss Recovery Appliance (ZDLRA)リアルタイム・ログ・トランスポート、Oracle GoldenGate ダウンストリーム取得など、プライマリ・データベースから REDO を受信するすべての製品またはサービスに影響します。

#### オプション

[yes | no]

デフォルト値はnoです

デフォルトはnoです(ログ出送信は遅延されません)。デフォルトをYesに変更するとログ送信は遅延され、アップグレード後はこれを手動で再度有効化する必要があります。

#### 例

```
defer_standby_log_shipping=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## drop\_grp\_after\_upgrade

データベースのアップグレード後に、保証付きリストア・ポイント(GRP)を削除します。

#### 使用上のノート

このオプションを選択した場合、アップグレードが正常に完了した後にGRPが削除されます。

#### オプション

[yes | no]

デフォルト値はnoです。

#### 例

```
global.drop_grp_after_upgrade=yes
```

```
sales.drop_grp_after_upgrade=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## enable\_local\_undo

CDBのアップグレードの場合、CDB\$ROOTのアップグレード前にLOCAL UNDOを有効にする必要があるかどうかを指定します。

使用上のノート

このオプションを選択した場合、AutoUpgradeはアップグレード前に次の文を実行します: ALTER DATABASE LOCAL UNDO ON;。

ローカルUNDOを最初に有効にすると、PDB\$SEED内のUNDO表領域のサイズは、CDB\$ROOT内のUNDO表領域のサイズの係数として決定されます。デフォルトは、UNDO表領域サイズの30%です。CDB内の他のすべてのPDBは、PDB\$SEEDからこのプロパティを継承します。新しいUNDO表領域を割り当てるのに十分な領域があることを確認します。

オプション

[yes | no]

デフォルト値はnoです。

例

```
enable_local_undo=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## manage\_network\_files

アップグレード中にネットワーク・ファイルを処理するかどうかを指定します。

使用上のノート

このオプションを選択すると、AutoUpgradeは、指定したオプションに応じてネットワーク・ファイルを処理します。

oranfstab、ldap.ora、tnsnames.ora、sqlnet.oraおよびlistener.oraの各ネットワーク・ファイルが処理されます。

オプション

[FULL|SKIP|IGNORE\_READ\_ONLY]

- FULL: (デフォルト)ネットワーク・ファイルのターゲットOracleホームへのコピーおよびマージ中に発生したすべての例外を呼び出します。
- SKIP: アップグレード後にネットワーク・ファイルを処理しません。
- IGNORE\_READ\_ONLY: ネットワーク・ファイルをコピーおよびマージしようとしたが、ターゲット・ファイルが読み取り専用の場合、アップグレード中に例外は発生しません。

例

```
manage_network_files=ignore_read_only
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## remove\_underscore\_parameters

構成ファイル内のすべてのOracle Databaseについて、アップグレード時およびアップグレード後にPFILFILEファイルからアンダースコア(非表示)パラメータを削除します。

使用上のノート

アンダースコアのパラメータは、Oracleサポートの指示がある場合のみ使用できます。

オプション

[yes | no]

デフォルト値はnoです。

例

```
global.remove_underscore_parameters=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## restoration

(Enterprise Editionでのみ使用可能)データベース・リストアの保証付きリストア・ポイント(GRP)を生成します。

使用上のノート

このオプションは、データベース・バックアップおよびデータベースのリストアをDBAによって手動で実行する必要があるかどうかを決定します。

Standard Editionはフラッシュバック・データベースをサポートしていないため、このオプションはStandard Editionでは使用できません。データベースがStandard EditionのOracle Databaseの場合、別個のフォールバック・メカニズムを持つようにする必要があります。

オプション

[yes | no]

デフォルト値はyes。

例

```
global.restoration=no
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## target\_base

ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。

例

```
global.target_base=/u01/app/oracle
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## target\_home

(ターゲット・ホームがシステム上にない場合で、アップグレード・モードおよびデブレイ・モードの場合は必須です。分析および修正モードの場合はオプションです。)ターゲットOracleホーム(ORACLE\_HOME)パスを指定します。

使用上のノート

AutoUpgradeでは、このパラメータで指定したリリース・バージョン情報を使用して、アップグレード対象のターゲットのOracle Databaseリリースに対して正しいチェックおよび修正が使用されるようにします。このパラメータの形式は、有効なOracleバージョンのピリオド区切りの値です。

このオプションは、ターゲット・ホームがシステムに存在しないか、ターゲット・ホームが12.2リリースの場合のみ必要です。それ以外の場合、AutoUpgradeはターゲット・リリース値を導出できます。

オプション

有効な値

- 12.2
- 18
- 19
- 21

例

```
global.target_version=19
employees.target_version=12.2
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## target\_version

(ターゲットOracleホームがシステム上にないか、リリース12.2の場合は必須) AutoUpgradeでアップグレードを実行するターゲット・リリース・バージョンを指定します。

使用上のノート

AutoUpgradeでは、このパラメータで指定したリリース・バージョン情報を使用して、アップグレード対象のターゲットのOracle Databaseリリースに対して正しいチェックおよび修正が使用されるようにします。このパラメータの形式は、有効なOracleバージョンのピリオド区切りの値です。

このオプションは、ターゲット・ホームがシステムに存在しないか、ターゲット・ホームが12.2リリースの場合のみ必要です。それ以外の場合、AutoUpgradeはターゲット・リリース値を導出できます。

オプション

有効な値

- 12.2
- 18
- 19
- 21

例

```
global.target_version=19
employees.target_version=12.2
```

**親トピック:** [AutoUpgrade構成ファイルのローカルに変更可能なグローバル・パラメータ](#)

## AutoUpgradeを使用した切断/接続アップグレードの理解

AutoUpgradeでは、以前のリリースのソース・コンテナ・データベース(CDB)からプラガブル・データベース(PDB)を切断し、新しいリリースのターゲットCDBに接続して、PDBをターゲットCDBリリースにアップグレードするために必要なすべてのステップを完了できます。

アップグレードの構成方法に応じて、AutoUpgradeを使用した切断/接続PDBアップグレードには2つのワークフローがあります。

- 1つ以上のプラガブル・データベースを1つのソースCDBから切断し、新しいリリースのターゲットCDBに接続します。
- 異なるソースCDBから複数のプラガブル・データベースを切断し、新しいリリースのターゲットCDBに接続します。

アップグレード後、PDBごとに、データベース・リスナーおよびローカル・ネーミング・パラメータ(tnsnames.oraファイル)を構成する必要があります。Oracle Wallet管理も構成する必要があります。

注意:

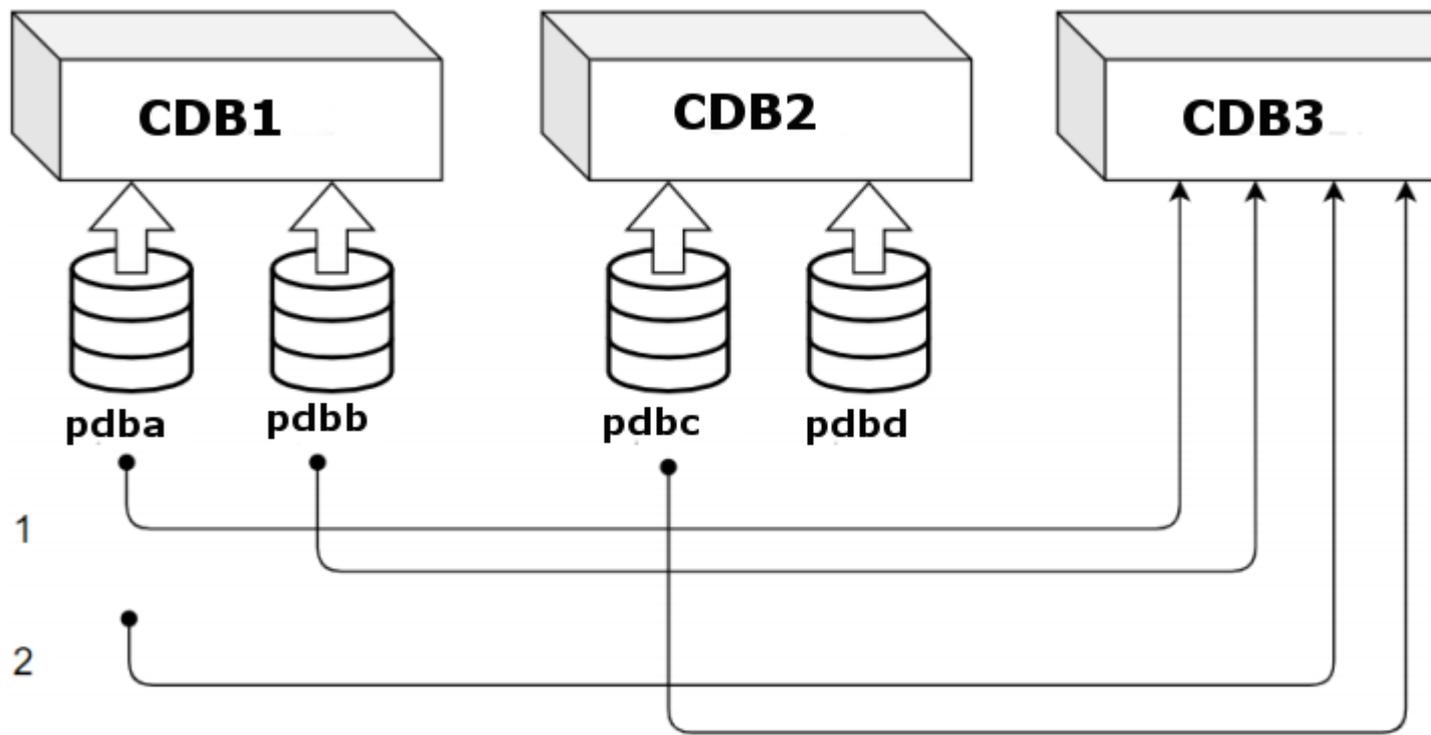
データベースに対するその他の変更と同様に、AutoUpgrade を実行して変換およびアップグレードを完了する前に、予期しないデータ損失を防ぐために信頼性のあるバックアップ計画を実装することをお勧めします。

AutoUpgrade がこの手順を開始した後、切断/接続 PDB アップグレードをロールバックするオプションはありません。AutoUpgrade では、透過的データ暗号化(TDE)を使用する PDB または暗号化された表領域を含む PDB の切断/接続アップグレードはサポートされていません。

次の図は、使用できる2つの切断/接続オプションを示しています。

1. あるソースのOracle Database 12.2 CDB (pdbaおよびpddbを含むCDB1)からPDBを切断し、新しいリリースのターゲットCDB (CDB3)に接続します。
2. 複数のソースCDB (CDB1、pdbaおよびpddb上のOracle Database 12.2)およびOracle Database 18c、CDB2、pdbcおよびpddb)からPDBを切断し、新しいリリースのターゲットCDB (CDB3)に接続します。

図2-1 ソースからターゲットへの切断/接続アップグレード



#### ソースCDBとターゲットCDBの要件

切断/接続アップグレードを実行するには、ソースCDBとターゲットCDBが次の条件を満たしている必要があります。

- 切断/接続アップグレードを開始する前に、ターゲット・リリースCDBを作成し、CDBをオープンしていること。
- ソースCDBとターゲットCDBのエンディアン形式が同じであること。
- ターゲット・リリースCDB用に構成されたOracle Databaseコンポーネントのセットに、ソースCDBで使用可能なすべてのコンポーネントが含まれていること。
- ソースCDBとターゲットCDBに、互換性のある文字セットと各国語文字セットがあること。
- ターゲットCDBリリースへの直接のアップグレードでは、ソースCDBリリースがサポートされている必要があります。
- ソースCDBおよびターゲットCDBに対して外部認証(オペレーティング・システム認証)が有効になっていること
- ソースCDBのOracle Application Expressインストール・タイプは、ターゲットCDBのインストール・タイプと一致する必要があります。
- CDBに接続する非CDB Oracle Databaseには、既存の保証付きリストア・ポイント(GRP)を含めないでください。

#### 注意:

AutoUpgrade を、Oracle Data Guard 構成の一部である CDB への切断/接続アップグレードを実行するために使用しないでください。切断/接続を使用して Oracle Data Guard 構成の CDB に PDB をアップグレードするには、次の My Oracle Support ノートで説明されている手順を使用して手動でアップグレードを実行する必要があります。

[Oracle Multitenant での遅延 PDB リカバリおよび STANDBYS = NONE 機能の使用\(ドキュメント ID](#)

## 切断/接続アップグレードの機能

切断/接続アップグレードを選択すると、AutoUpgrade構成ファイルの構成方法に応じて、AutoUpgradeを使用してアップグレード時に次のオプションを実行できます。

- ソースCDBにあるPDB名を保持するか、PDB名を変更できます。
- 古いファイルをすべて保持しながら、データ・ファイルのコピーをターゲットCDBに作成できます。
- データファイルをターゲットの場所にコピーしてから、ソースCDBで古いファイルを削除できます。
- 1つのPDBを処理することも、包含リストにリンクして多数のPDBを1つのアップグレード手順で処理することもできます。処理できるPDBの数の制限は、サーバー制限とCDB上のPDBの制限のみです。

### 例2-1 切断/接続アップグレード用のAutoUpgrade構成ファイル

切断/接続PDBアップグレード・オプションを使用するには、AutoUpgrade構成ファイルでソースCDBおよびPDBのシステム識別子パラメータtarget\_cdbを設定する必要があります。target\_cdbパラメータの値は、非CDBのOracleデータベースを接続するテナント・データベースのOracleシステム識別子(SID)を定義します。次に例を示します。

```
global.autoupg_log_dir=/home/oracle/autoupg
global.autoupg_log_dir=/home/oracle/autoupg
upg1.sid=CDB122
upg1.source_home=/u01/app/oracle/product/12.2.0/dbhome_1
upg1.target_home=/u01/app/oracle/product/19.1.0/dbhome_1
upg1.target_version=19.1.0
upg1.target_cdb=cdb21x
upg1.target_pdb_name.pdb_2=depsales
upg1.target_pdb_copy_option.pdb_2=file_name_convert=('pdb_2','depsales')
```

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# 新しいOracleホームへのOracleソフトウェアのインストール

ターゲットのOracleホームの新しい場所を選択し、単一インスタンス用の新しいOracle Databaseリリース・ソフトウェアをインストールします。

- [アップグレード時のOracleホームの新しい場所の選択](#)  
新しいリリースのOracle Databaseに、現行のリリースのOracleホームとは別のOracleホームの場所を選択する必要があります。
- [単一インスタンス用の新しいOracle Databaseソフトウェアのインストール](#)  
この手順の概要を使用することで、単一インスタンス・デプロイメント用に新しいOracle Databaseリリースのソフトウェアのインストールが容易になります。

親トピック: [Oracle Databaseのアップグレードの準備](#)

## アップグレード時のOracleホームの新しい場所の選択

新しいリリースのOracle Databaseに、現行のリリースのOracleホームとは別のOracleホームの場所を選択する必要があります。

別のインストール場所を使用すると、新しいOracleソフトウェアとともに、既存のOracleソフトウェアをインストールしたままにできます。個別のインストール場所を使用することで、本番環境全体を置き換える前にテスト・データベース上でアップグレード処理をテストできます。

データベースをアップグレードする際、データベースが非CDBかCDBにかかわらず、新しいOracleホームをインストールするための新しい場所が必要です。

切断/接続アップグレードを使用してPDBをアップグレードする場合、PDBを接続するターゲットCDBがPDBの場所になります。ターゲットCDBにはすでにOracleホームが存在するため、ターゲットOracleホームをインストールするための新しい場所を選択する必要はありません。

## 単一インスタンス用の新しいOracle Databaseソフトウェアのインストール

この手順の概要を使用することで、単一インスタンス・デプロイメント用に新しいOracle Databaseリリースのソフトウェアのインストールが容易になります。

このリリースの新しいOracle Databaseソフトウェアをインストールするには:

1. Oracleオペレーティング・システム固有のドキュメントの指示に従って、Oracle Databaseソフトウェアのインストールを準備します。
2. Oracle Universal Installerを起動して、ソフトウェアのみのインストールを選択します。  
  
Oracle Databaseソフトウェアのインストールが正常に完了したら、「終了」ボタンをクリックしてOracle Universal Installerを閉じます。
3. Oracle Label Security、Oracle Database Vault、またはこれらの両方を使用する場合、データベース・エディションの選択ページで「Enterprise Edition」を選択し、「オプションの選択」をクリックして、コンポーネント・リストから一

方または両方のコンポーネントを有効にします。

# Oracle Databaseをアップグレードする前のバックアップ計画の準備

アップグレードを確実に成功させるため、適切なバックアップ計画を設計して実行する必要があります。

Oracle Database Enterprise Editionの場合、主なフォールバック・メカニズムはフラッシュバック・データベースです。ただし、フラッシュバック・データベースを使用して切断/接続アップグレードやPDB変換を元に戻すことはできません。切断/接続アップグレードの場合は、RMANバックアップなどの他のフォールバック計画を使用します。

AutoUpgradeを使用する場合は、AutoUpgrade構成ファイル内でtarget\_pdb\_copy\_option=file\_name\_convertを指定することをお勧めします。ここで、file\_name\_convertは、データ・ファイルに接頭辞として付ける変換パターンです。これを行うと、AutoUpgradeは、データベースを接続する前にデータ・ファイルのコピーを作成するようにデータベースに指示します。この方法の使用を選択すると、元のデータベースをフォールバックとして使用できます。ただし、データ・ファイルのコピーを作成する場合はアップグレードにさらにディスク領域と時間が必要になることに留意してください。

バックアップ戦略を展開するには、次のような問題について考慮する必要があります。

- 業務上、本番データベースの実行不可能状態の許容範囲がどの程度の期間か。
- 可用性要件を満たすには、どのバックアップ戦略が必要か。
- サイトから離れた安全な場所にバックアップをアーカイブする必要があるか。
- 正しくバックアップされたことを確認するためのテストを実行するか。
- どのくらいの時間でバックアップをリストアできるか(オフサイト記憶域でのバックアップを含む)。
- 障害リカバリ手順は正常にテストされているか。

バックアップ計画は、これらの問題のすべてに答え、データベースを正常にバックアップおよびリカバリするための手順を備えている必要があります。RMANを使用したバックアップ計画の実装の詳細は、*Oracle Database*バックアップおよびリカバリ・ユーザー・ガイドを参照してください。

また、ダウングレードの準備を確実に行うには、ダウングレードに関する章を確認し、リリースの準備に必要な準備ステップを完了してください。

## 関連項目

- [データベースのバックアップ](#)
- [フラッシュバック・データベースおよびリストア・ポイントの使用](#)

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# Oracle Databaseのアップグレードを開始する前に完了するデータベースの準備作業

Oracle Databaseのアップグレードを開始する前に、これらのデータベースの準備作業を完了していることを確認してください。

- [Oracle Databaseをアップグレードする場合のリリースの更新および要件](#)  
アップグレードを開始する前に、新しいリリースのOracleホームを最新のリリース更新(Update)に更新します。
- [Oracle Databaseのアップグレード時のOracle Net Servicesに関する推奨事項](#)  
リスナーが新しいリリースのOracleホームで実行されていることを確認する必要があります。
- [パスワードの大/小文字の区別とアップグレードについて](#)  
デフォルトで、Oracle Database 12cリリース2 (12.2)以降のリリースでは、排他モード認証プロトコルが使用されません。排他モードは大/小文字を区別しないパスワードを基にした認証をサポートしません。
- [大/小文字を区別しないパスワード・バージョンを使用しているアカウントがあるかどうかの確認](#)  
次の手順を使用して、アップグレードするOracle Databaseに大/小文字を区別しないパスワード・バージョンを使用しているアカウントや構成パラメータがないか、確認します。
- [読取り専用の表領域を使用したアップグレードの実行](#)  
アップグレード中にスキーマベースの表領域をオフラインに設定するには、-Tオプションを指定してパラレル・アップグレード・ユーティリティを使用します。

親トピック: [Oracle Databaseのアップグレードの準備](#)

## Oracle Databaseをアップグレードする場合のリリースの更新および要件

アップグレードを開始する前に、新しいリリースのOracleホームを最新のリリース更新(Update)に更新します。

新しいOracle Databaseリリースのソフトウェアには、リリースの時点で最新のOracle Databaseのすべての更新を含む完全なリリースが含まれます。

アップグレードを開始する前に、新しいリリースのOracleホームを最新の四半期リリース更新(Update)に更新することをお勧めします。

My Oracle Supportで、最新更新の入手方法に加え、ライフサイクル管理に使用するツールについて詳細に説明したノートが提供されています。次に例を示します。

- My Oracle Supportノート555.1のOracle Database 19cの重要な推奨個別パッチには、Oracle Database 19cの特定の重要度のパッチのリストが含まれています。
- My Oracle Supportノート2118136.2には、環境に必要な更新、リビジョン、パッチ・セット更新(PSU)、SPU (CPU)、バンドル・パッチ、パッチセットおよびベース・リリースの選択に役立つダウンロード・アシスタントが含まれています。ここから始めることをお勧めします。
- My Oracle Supportノート1227443.1には、Oracle DatabaseのPSU/BP/Update/Revisionに関する既知の問題のリストが含まれています。このノートには、Oracle Database、Oracle Grid InfrastructureおよびOracle JavaVMコンポーネント(OJVM)のすべての既知の問題ノートに関する情報が記載されています。

### 関連項目

- [Oracle Database 19cの重要な推奨個別パッチ\(ドキュメントID 555.1\)](#)

- [My Oracle Supportノート2118136.2](#)
- [My Oracle Supportノート1227443.1](#)

## Oracle Databaseのアップグレード時のOracle Net Servicesに関する推奨事項

リスナーが新しいリリースのOracleホームで実行されていることを確認する必要があります。

アップグレードするOracle Databaseでリスナーが構成されていない場合、アップグレードを開始する前に、Oracle Net Configuration Assistant (NETCA)を実行して、listener.oraファイルを含むOracle Databaseの新しいリリースのリスニング・プロトコルのアドレスおよびサービス情報を構成する必要があります。現在のリスナーには、以前のリリースのOracle Databaseとの下位互換性があります。

Oracle Real Application Clusters Oracle DatabaseまたはOracle Database 12cより古いリリースをアップグレードする場合は、次の追加情報を確認してください。

Oracle RACデータベースのアップグレードでは、リスナーは通常、グリッド・インフラストラクチャのアップグレード中に移行されます。Oracle Grid Infrastructureホームでlsnrctlコマンドを使用してリスナーを管理する必要があります。以前のリリースのOracleホームの場所からlsnrctlコマンドを使用しないでください。

### 関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)

## パスワードの大/小文字の区別とアップグレードについて

デフォルトで、Oracle Database 12cリリース2 (12.2)以降のリリースでは、排他モード認証プロトコルが使用されます。排他モードは大/小文字を区別しないパスワードを基にした認証をサポートしません。

サーバーが排他モードで実行されている場合、10Gパスワード・バージョンだけを持つアカウントにはアクセスできなくなります。

以前のリリースのOracle Databaseでは、SEC\_CASE\_SENSITIVE\_LOGON=FALSEを設定することにより、大/小文字を区別しないパスワード・ベースの認証を許可するように認証プロトコルを構成できます。Oracle Database 12cリリース2 (12.2)以降は、デフォルトのパスワード・ベースの認証プロトコル構成で、大/小文字を区別しない10Gパスワード・バージョンの使用が排除されています。デフォルトでは、SQLNET.ORAパラメータSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERが12(排他モード)に設定されています。データベースが排他モードで設定されている場合、パスワード・ベースの認証プロトコルでは大/小文字を区別するいずれかのパスワード・バージョン(11Gまたは12C)が、認証されるアカウントに存在する必要があります。このモードは、以前のリリースで使用されていた10Gパスワード・バージョンの使用を排除します。Oracle Database 12cリリース2以降のリリースにアップグレードした後は、大/小文字を区別しない10Gパスワード・バージョンしか持たないアカウントにはアクセスできなくなります。この変更が発生するのは、サーバーがデフォルトで排他モードで実行されるためです。Oracle Databaseが排他モードで構成されていると、古い10Gパスワード・バージョンを使用してクライアントを認証できません。サーバーにはそのクライアントの認証に使用するパスワード・バージョンがありません。

セキュリティを高めるためには、大/小文字を区別するパスワード・ベースの認証を有効にしておくことをお勧めします。この設定が

デフォルトです。ただし、新しいOracle Databaseリリースへのアップグレード中に、大/小文字を区別する認証を一時的に無効にすることはできません。アップグレード後に、パスワード・バージョンを管理するための実装計画の一環として、大/小文字を区別するパスワード・ベースの認証機能を有効にするかどうかを決定できます。

アップグレード前に、デフォルトのパスワード・ベースの認証プロトコル構成に対するこの変更により影響を受けるかどうかを判定することをお勧めします。次のことを確認してください。

- 10Gの大/小文字を区別しないパスワード認証バージョンだけを持つアカウントがあるかどうかを確認します。
- 重要なパッチ更新CPUOct2012またはそれ以降のパッチ更新を適用していないOracle Database 11gリリース2 (11.2.0.3)データベースがあるかどうか、かつ、大/小文字を区別しない10Gパスワード・バージョンを持たないアカウントがあるかどうかを確認します。
- FALSEに設定した非推奨のパラメータSEC\_CASE\_SENSITIVE\_LOGONがないことを確認します。このパラメータをFALSEに設定すると、大/小文字を区別するパスワード・バージョン(11Gおよび12Cパスワード・バージョン)を認証に使用できません。

大/小文字を区別しないバージョンを使用するアカウントのオプション

大/小文字を区別しない10Gパスワード・バージョンのみを持つユーザー・アカウントがある場合は、次の代替方式のいずれかを選択する必要があります。

- アップグレード前に、10Gパスワード・バージョンだけを持つそれぞれのアカウントのパスワード・バージョンを更新します。パスワード・バージョンを更新するには、10Gパスワード・バージョンを使用するユーザー・パスワードを期限切れにして、そうしたユーザーにアカウントにログインするように要求します。ユーザーがログインを試行すると、サーバーはパスワード・バージョンのリストを自動的に更新しますが、これには大/小文字を区別するパスワード・バージョンが含まれます。
- SQLNET.ORAパラメータSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERの設定を、排他モード以外の設定に変更します。たとえば、SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER=11です。

#### 関連項目

- [Oracle Database 2日データベース管理者](#)
- [『Oracle Database Net Servicesリファレンス・ガイド』](#)
- [『Oracle Databaseセキュリティ・ガイド』](#)

## 大/小文字を区別しないパスワード・バージョンを使用しているアカウントがあるかどうかの確認

次の手順を使用して、アップグレードするOracle Databaseに大/小文字を区別しないパスワード・バージョンを使用しているアカウントや構成パラメータがないか、確認します。

デフォルトで、Oracle Database 12cリリース2 (12.2)以降のリリースでは、10Gパスワード・バージョンは生成されることも許可されることもありません。

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERを、大/小文字を区別しないバージョンを使用できる許容度の高い認証プロトコルに設定しておらず、かつ、大/小文字を区別しないパスワード・バージョンで認証されるユーザー・アカウントがデータベースでロックされないようにする場合は、該当するアカウントを確認して、それらが大/小文字を区別するパスワード・バージョンを使用していることを確認します。

## 例2-2 大/小文字を区別しない(10G)バージョンを使用するユーザー・アカウントの検索

管理ユーザーとしてSQL\*Plusにログインし、次のSQL問合せを入力します。

```
SELECT USERNAME, PASSWORD_VERSIONS FROM DBA_USERS;
```

次の結果は、アカウントのパスワード・バージョンを示しています。

USERNAME	PASSWORD_VERSIONS
JONES	10G 11G 12C
ADAMS	10G 11G
CLARK	10G 11G
PRESTON	11G
BLAKE	10G

この例では、各ユーザー・アカウント・パスワードの使用中のバックグラウンド検証バージョンは異なります。

- JONESはOracle Database 10Gで作成されましたが、JONESのパスワードは、SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータの設定が8に設定されたときにOracle Database 12Cでリセットされました。その結果、このパスワードのリセットによって3つのバージョンがすべて作成されました。11Gおよび12Cは大/小文字を区別するパスワードを使用します。
- ADAMSおよびCLARKは最初に10Gバージョンで作成され、次に、以前のリリースからインポートされた後に11Gバージョンで作成されました。これらのアカウント・パスワードはその後、非推奨のパラメータSEC\_CASE\_SENSITIVE\_LOGONをTRUEに設定した場合、11Gでリセットされました。
- BLAKEのパスワードは10Gバージョンで作成され、パスワードはリセットされませんでした。その結果、ユーザーBLAKEは引き続き10Gパスワード・バージョンを使用し、これは大/小文字を区別しないパスワードを使用します。

ユーザーBLAKEはアップグレード前には10Gパスワード・バージョンのみを持っています。

```
SQL> SELECT USERNAME, PASSWORD_VERSIONS FROM DBA_USERS;  
USERNAME PASSWORD_VERSIONS  
-----  
BLAKE 10G
```

パスワードに対する措置をせずにただ、新しいOracle Databaseリリースにアップグレードした場合、そのアカウントにはアクセスできなくなります。アップグレード前に、(SQLNET.ORAパラメータSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERを、より許容度の高い認証モードに設定して)システムが排他モードで構成されていないことを確認します。

## 例2-3 大/小文字を区別しないパスワードを持つアカウントの修正

次の手順を実行します。

1. 次のSQL問合せを使用して、10Gパスワード・バージョンだけを持つアカウントを検索します。

```
select USERNAME  
  from DBA_USERS  
 where ( PASSWORD_VERSIONS = '10G '  
        or PASSWORD_VERSIONS = '10G HTTP '  
        and USERNAME <> 'ANONYMOUS' );
```

2. SQLNET.ORAパラメータSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERの設定を、該当するアカウントに適切なレベルに

編集することにより、システムが排他モードで実行されるようにシステムを構成します。次に例を示します。

```
SQLNET.ALLOWED_LOGON_VERSION_SERVER=11
```

この変更を行ってから、アップグレードに進みます。

- アップグレードが終了したら、次のコマンド構文を使用してステップ1で検出したアカウントを期限切れにします。usernameは、ステップ1の問合せで返されたユーザーの名前です。

```
ALTER USER username PASSWORD EXPIRE;
```

- パスワードを期限切れにしたユーザーに、ログインするよう依頼します。
- これらのユーザーがログインするときに、パスワードをリセットするよう、プロンプトが表示されます。システムは10Gパスワード・バージョンに加えて、欠落している11Gおよび12Cパスワード・バージョンを内部的に生成します。システムは許容モードで実行中のため、10Gパスワード・バージョンは引き続き存在します。
- ユーザーが接続しているクライアント・ソフトウェアに05L\_NP機能フラグがあることを確認します。

ノート:

Oracle Database リリース 11.2.0.4 以降のすべてのクライアント、および Oracle Database リリース 12.1 以降のすべてのクライアントには 05L\_NP 機能があります。他のクライアントでは CPUOct2012 パッチを適用して 05L\_NP 機能を取得する必要があります。

05L\_NP 機能フラグは、*Oracle Database Net Services* リファレンスのパラメータ SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER に関する項を参照してください。

- すべてのクライアントに05L\_NP機能が導入されたら、次の手順を使用してサーバー・セキュリティを元の排他モードに上げます。
  - インスタンス初期化ファイルからSEC\_CASE\_SENSITIVE\_LOGON設定を削除するか、SEC\_CASE\_SENSITIVE\_LOGONインスタンス初期化パラメータをTRUEに設定します。次に例を示します。

```
SEC_CASE_SENSITIVE_LOGON = TRUE
```
  - サーバーのSQLNET.ORAファイルからSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータを削除するか、サーバーのSQLNET.ORAファイルのSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERの値を12に変更して元の排他モードに設定します。次に例を示します。

```
SQLNET.ALLOWED_LOGON_VERSION_SERVER = 12
```

- 次のSQL問合せを使用して、まだ10Gパスワード・バージョンを持っているアカウントを検索します。

```
select USERNAME
  from DBA_USERS
 where PASSWORD_VERSIONS like '%10G%'
    and USERNAME <> 'ANONYMOUS';
```

- ステップ8の問合せから戻されたアカウントのリストを使用して、まだ10Gパスワード・バージョンを持っているすべてのアカウントを期限切れにします。次の構文を使用して、アカウントを期限切れにします。usernameは問合せで返されたリストにある名前です。

```
ALTER USER username PASSWORD EXPIRE;
```

10. 期限切れにしたアカウントのユーザーに、アカウントにログインするよう依頼します。

ユーザーはログインするときに、パスワードを再設定するようプロンプトが表示されます。システムは、アカウントの11Gおよび12Cパスワード・バージョンだけを内部的に生成します。システムは排他モードで実行されているため、10Gパスワード・バージョンは生成されません。

11. ステップ1の問合せを再度実行して、システムがセキュア・モードで実行されていることを確認します。どのユーザーも検出されないことを確認します。この問合せでどのユーザーも検出されない場合、それは、10Gパスワード・バージョンがシステムに残っていないことを意味します。

#### 例2-4 SEC\_CASE\_SENSITIVE\_LOGON SetがFALSEになっているものがないことの確認

SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータが12または12aに設定されている場合、Oracle DatabaseはSEC\_CASE\_SENSITIVE\_LOGONのFALSE設定の使用を妨げません。この設定により、アップグレードされたデータベースのすべてのアカウントがアクセス不能になる可能性があります。

```
SQL> SHOW PARAMETER SEC_CASE_SENSITIVE_LOGON
```

NAME	TYPE	VALUE
sec_case_sensitive_logon	boolean	FALSE

次のコマンドを使用して、このパラメータを変更できます。

```
SQL> ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = TRUE;  
System altered.
```

ノート:



パラメータ SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER の値が 12 よりも許容度の高いバージョン(11 など)に変更されないかぎり、SEC\_CASE\_SENSITIVE\_LOGON パラメータを FALSE に設定しないでください。

#### 関連項目

- [『Oracle Database Net Servicesリファレンス・ガイド』](#)
- [『Oracle Databaseセキュリティ・ガイド』](#)

## 読取り専用の表領域を使用したアップグレードの実行

アップグレード中にスキーマベースの表領域をオフラインに設定するには、-Tオプションを指定してパラレル・アップグレード・ユーティリティを使用します。

Oracle Databaseでは以前のリリースで作成されたファイル・ヘッダーを読み取ることができるため、アップグレード時にそれらに対して処理を行う必要はありません。READ ONLY表領域のファイル・ヘッダーは、それらがREAD WRITEに変更されたときに更新されます。

ノート:



非 CDB から PDB への変換を実行する場合、読取り専用表領域の使用は有効なフォールバック・オプションでは

ありません。非 CDB から PDB への変換の間には、表領域が、変換中はオンラインである必要があります。これは、アップグレード中に各データ・ファイル・ヘッダーの変更が必要になるためです。

アップグレードで致命的なエラーが発生し、その結果アップグレードで表領域をオンラインに戻すことができなくなった場合、アップグレード・ログ・ファイルを参照してください。ログ・ファイルには、表領域を使用可能にするために必要な実際のSQL文が含まれています。表領域をオンラインに戻すには、データベースにログ・ファイルのSQL文を実行するか、PDBごとにログ・ファイルを実行する必要があります。

#### アップグレード・ログ・ファイルの表領域コマンドの表示

致命的なアップグレードの失敗が発生した場合は、ログ・ディレクトリ(Oracle\_base/cfgtoollogs/dbua)に移動してログ・ファイルのコマンドを手動で実行して表領域を表示させることができます。表領域コマンドは次のログ・ファイルで表示できます。

- 非CDBのアップグレード: catupgrd0.log
- PDBデータベース: catupgrdpdbname0.log (pdbnameはアップグレードするPDBの名前です)。

各ログ・ファイルの先頭で、次のようなSQL文が見つかります。これは、表をREAD ONLYに設定する文です。

```
SQL> ALTER TABLESPACE ARGROTBLSPA6 READ ONLY;
Tablespace altered.
SQL> ALTER TABLESPACE ARGROTBLSPB6 READ ONLY;
Tablespace altered.
```

各ログ・ファイルの末尾近くには、表をREAD WRITEに再設定するSQL文が見つかります。

```
SQL> ALTER TABLESPACE ARGROTBLSPA6 READ WRITE;
Tablespace altered.
SQL> ALTER TABLESPACE ARGROTBLSPB6 READ WRITE;
Tablespace altered.
```

参照:

データベース間での表領域の転送の詳細は、[Oracle Database管理者ガイド](#)を参照してください。

# Oracle Databaseのアップグレード後のOracle Database Vaultの有効化

ターゲットのデータベース・リリースによっては、Oracle Database Vaultを無効にしないとOracle Databaseのアップグレードを完了できない場合があります。

- [Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード](#)  
ターゲットのOracle Databaseリリースが12.2以上である場合は、Oracle Database Vaultを無効にせずにアップグレードできます。
- [Oracle Database Vaultが関連する一般的なアップグレード・シナリオ](#)  
アップグレード後にOracle Database Vaultを有効にする必要があるかどうかは、ソースのOracle Databaseリリースによって異なります。

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

## Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード

ターゲットのOracle Databaseリリースが12.2以上である場合は、Oracle Database Vaultを無効にせずにアップグレードできます。

ソースのOracle DatabaseリリースでOracle Database Vaultを有効にしていた場合は、Oracle Database Vaultを事前に無効にしなくても、Oracle DatabaseをOracle Database 18c以降のリリースにアップグレードできます。ソースのOracle DatabaseリリースがOracle Database 12cリリース1 (12.1)以上である場合は、アップグレード後に、アップグレード前に用意したものと同一強化設定を使用してOracle Database Vaultが有効になります。たとえば、ソース・データベースがOracle Databaseリリース12.1であり、そのリリースでOracle Database Vaultを無効にしていた場合は、アップグレード後も無効になります。アップグレード前にソースのOracle Databaseリリース12.1データベースでOracle Database Vaultを有効にしていた場合は、アップグレード後もOracle Database Vaultは有効になります。

アップグレード前に手動でOracle Database Vaultを無効にした場合は、アップグレード後に手動でOracle Database Vaultを有効にする必要があります。

アップグレード前にOracle Database Vaultを有効にしていなかった場合は、アップグレード後に手動で有効にできます。

プロシージャ`dvsys.dbms_macadm.enable_dv()`を使用して、アップグレードしたデータベースでOracle Database Vaultを有効にします。このプロシージャは、DV\_OWNERを付与されたユーザー・アカウントを使用して実行します。プロシージャを実行したら、そのプロシージャを有効にするためにデータベース・インスタンスを再起動します。

### 関連項目

- [Oracle Database Vaultの管理者ガイド](#)

# Oracle Database Vaultが関連する一般的なアップグレード・シナリオ

アップグレード後にOracle Database Vaultを有効にする必要があるかどうかは、ソースのOracle Databaseリリースによって異なります。

- Oracle Database 11gリリース2 (11.2)以下からのアップグレード: アップグレード後、Oracle Database Vaultはデフォルトで無効になります。
- Oracle Database 12cリリース1 (12.1)以上からのアップグレード: アップグレード後、Oracle Database Vaultはアップグレード前に用意したものと同一強化ステータスになります。

表2-2 Oracle Database Vaultの一般的なアップグレード・シナリオおよびアップグレード準備作業

ソース・データベース・リリース	ターゲット・データベース・リリース	アップグレード前にDatabase Vaultを無効にする必要があるかどうか	アップグレード後のDatabase Vaultのステータス
11.2 以前	12.1	必要	無効。アップグレード後に Database Vault を手動で有効にする必要があります。
11.2 以前	12.2、18.1 以降	不要	無効。アップグレード後に Database Vault を手動で有効にする必要があります。
12.1、12.2、18.1 以降	12.2、18.1 以降	不要	Database Vault はアップグレード前に用意したものと同一強化ステータスになります。

# AutoUpgrade処理モードを実行するための準備

AutoUpgrade処理モードを実行する前に、準備を完了している必要があります。

AutoUpgrade処理モードを使用する前に、次の要件を満たしていることを確認してください。

- ユーザー構成ファイルが作成されていること。
- ソースOracle Databaseリリースは、元のOracleホームで稼働しています。AutoUpgradeの再起動の場合は、アップグレード・フローのフェーズに対応するOracleホームでデータベースを起動する必要があります。
- データベースが実行されているサーバーが、サーバー・ホスト・ファイル(たとえば/etc/hosts)またはドメイン・ネーム・サーバー(DNS)に登録されていること。

ターゲット・データベースが配置されているサーバーにログインしていて、データベースがlocalhostで実行されている場合、またはAutoUpgradeが実行されている場合は、AutoUpgradeのconfigファイルからhostnameパラメータを削除します。

- コンテナ・データベース(CDB)で、プラグブル・データベース(PDB)のサブセットをアップグレードする場合、アップグレードを実行するPDBはオープンされ、AutoUpgradeローカル・パラメータpdbを使用しユーザー構成ファイルで構成されます。PDBのリストを指定しない場合、AutoUpgradeはCDB上のすべてのPDBをアップグレードします。
- AutoUpgrade jarファイル(autoupgrade.jar)がダウンロード済または使用可能であり、Java 8配布を使用して実行できること。
- AutoUpgradeをバッチまたはスクリプトで実行する場合は、コマンドでnoconsoleパラメータを使用してAutoUpgradeをコールしておきます。

Oracle Database 19c (19.3)以降のターゲットのOracleホームでは、デフォルトでautoupgrade.jarファイルが存在します。ただし、AutoUpgradeを使用する前に、My Oracle Supportドキュメント2485457.1から入手可能な最新バージョンをダウンロードすることをお勧めします。

## 関連項目

- [My Oracle Supportのドキュメント2485457.1](#)

親トピック: [Oracle Databaseのアップグレードの準備](#)

# AutoUpgradeの構成ファイルの作成

AutoUpgradeを使用してアップグレードを完了するには、まず、新しいリリースのOracleホームからAutoUpgradeで構成ファイルを作成します。

次の例では、パラメータ`sample_config_file`を使用してAutoUpgradeユーティリティが実行されます。このパラメータは、AutoUpgradeを実行しているユーザーのホームで構成ファイルを生成します。このファイルを編集すると、アップグレードの環境パスと設定およびアップグレード・プリファレンスを提供できます。構成ファイル(`config`)を生成するには、`sample_config_file`パラメータを使用して新しいリリースのOracle DatabaseホームからAutoUpgradeを実行し、出力ファイル名を指定します。この例では、ユーザー`oracle`が以前のリリースのOracleホームの場所に移動します。この場合、これはOracle Database 12cリリース2 (12.2)になります。

```
cd /u01/app/oracle/product/12.2/
```

次に、OracleユーザーがOracle Database 19cのOracleホームからAutoUpgradeを起動し、ユーザー・ホーム・ディレクトリ`/home/oracle`に構成ファイルを作成します。

```
java -jar /u01/app/oracle/product/19/rdbms/admin/autoupgrade.jar -create_sample_file config  
Created sample configuration file /home/oracle/sample_config.cfg
```

構成ファイルを作成したら、優先テキスト・エディタでファイルを開き、環境に必要なパラメータ設定を変更します。

```
cd /  
vi sample_config.cfg
```

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# AutoUpgrade構成ファイルのローカル・パラメータ

AutoUpgradeユーティリティのアップグレード用に特定のOracle Databaseの情報を構成するには、AutoUpgradeローカル・パラメータに情報を指定します。

## 使用上のノート

ローカル・パラメータは、AutoUpgrade構成ファイルで設定されたグローバル・パラメータよりも優先されます。ローカルで設定する必要があるか、ローカルで変更可能なグローバル・パラメータとして設定する必要があるローカル・パラメータは(「必須」)で示されます。すべてのローカル・パラメータは接頭辞を使用します(例: 特定のデータベースまたはアップグレードを識別するために定義する値で識別)。接頭辞は、構成ファイルでパラメータが適用される特定のアップグレード・ジョブを識別します。

例: 構成ファイルの最初のアップグレードのパラメータのセットでは、接頭辞salesが使用され、構成ファイル内の次のアップグレードのパラメータのセットでは、接頭辞employeesが使用されます。

```
sales.source_home=/u01/app/oracle/12.2/dbhome1
.
.
employees.sid=salescdb
employees.source_home=/03/app/oracle/21/dbhome1
```

- [add\\_after\\_upgrade\\_pfile](#)  
(オプション)アップグレード後にパラメータを追加するPFILEのパスおよびファイル名を指定します。
- [add\\_during\\_upgrade\\_pfile](#)  
(オプション)アップグレード中にパラメータを追加するPFILEのパスおよびファイル名を指定します。
- [after\\_action](#)  
(オプション)deployモードでは、接頭辞アドレスで識別されるデータベースのデプロイ・ジョブの完了後に実行するカスタム・アクションを指定します。
- [before\\_action](#)  
(オプション) deployモードでは、接頭辞で指定された特定のデータベース・ジョブのアップグレード・ジョブの起動前に実行するカスタム・アクションを指定します。すべてのアップグレード・ジョブの前にスクリプトを実行する場合は、ローカル・パラメータ (global.before\_action)を使用してそのスクリプトを指定します
- [catctl\\_options](#)  
(オプション) AutoUpgradeがcatctl.plに送信してデフォルトの動作をオーバーライドするために選択できるcatctl.plオプションのセットを1つ以上指定します。
- [checklist](#)  
(オプション)ポリシーまたはセキュリティ上の問題のために自動的に実装しない修正など、AutoUpgradeによって実行される修正のデフォルト・リストをオーバーライドするために使用できるチェックリストへのパスを指定します。
- [del\\_after\\_upgrade\\_pfile](#)  
(オプション)アップグレード後にパラメータを削除するPFILEのパスおよびファイル名を指定します。
- [del\\_during\\_upgrade\\_pfile](#)  
(オプション)アップグレード中にパラメータを削除するPFILEのパスおよびファイル名を指定します。
- [env](#)

- (オプション) ORACLE\_SID、ORACLE\_HOME、ORACLE\_BASEおよびTNS\_ADMINを除く、オペレーティング・システムに設定されているカスタム・オペレーティング・システム環境変数を指定します。
- [keep\\_source\\_pdb](#)  
(オプション)切断/接続アップグレード操作のソースPDBが、ソースCDBから削除されるのではなく、閉じた状態に保持されるかどうかを指定します。
  - [log\\_dir](#)  
(オプション)パラメータの接頭辞で識別されるアップグレード・ジョブに含まれるデータベース・セットにあるデータベース・アップグレード用に生成されるログ・ファイルの場所を設定します。
  - [manage\\_standbys\\_clause](#)  
(オプション) DB\_UNIQUE\_NAMEで特定したスタンバイOracle Data Guardスタンバイ・データベースをAutoUpgradeプラグイン・アップグレードから除外して、スタンバイ・データベース・ファイルを再利用できるかどうかを指定します。
  - [pdbs](#)  
(オプション)アップグレードを実行するPDBのリストを設定します。このパラメータは、マルチテナント・アーキテクチャ(CDB)データベースのアップグレードにのみ適用されます。非CDBデータベースを接続およびアップグレードしている場合、このパラメータは無視されます。
  - [raise\\_compatible](#)  
(オプション)アップグレードが正常に完了した後、Oracle DatabaseのCOMPATIBLE初期化パラメータをターゲット・リリースのデフォルト値まで増やします。
  - [remove\\_underscore\\_parameters](#)  
(オプション)構成ファイル内のすべてのOracle Databaseについて、アップグレード中およびアップグレード後にPFILEファイルからアンダースコア(非表示)パラメータを削除します。
  - [restoration](#)  
(オプション)データベースのリストアに保証付きリストア・ポイント(GRP)を生成します。
  - [revert\\_after\\_action](#)  
(オプション)接頭辞によって指定された特定のデータベース・ジョブに対してシステムのリストアが完了し、データベースが稼働した後にオペレーティング・システムで実行するカスタム・アクションを指定します。
  - [revert\\_before\\_action](#)  
(オプション)接頭辞によって指定された特定のデータベース・ジョブに対してシステムのリストアが完了し、データベースが稼働する前にオペレーティング・システムで実行するカスタム・アクションを指定します。
  - [run\\_utltp](#)  
(オプション)アップグレードの一環としてutltpを有効または無効にします。
  - [sid](#)  
(必須)アップグレードするデータベースのOracleシステム識別子(SID)を示します。
  - [skip\\_tde\\_key\\_import](#)  
(オプション) yesに設定すると、アップグレードは実行されますが、ソース・データベースKeyStoreのターゲット・データベースへのインポートはスキップされ、エラーは発生しません。
  - [source\\_base](#)  
(オプション)ソースOracleホームのソースORACLE\_BASEパスを指定します。
  - [source\\_dblink](#)  
(オプション)切断/接続再配置(ホット・クローン)アップグレード用に設定されたデータベース・リンクを指定します。
  - [source\\_home](#)  
(分析モード、修正モードおよびデプロイ・モードの場合は必須です。アップグレード・モードの場合はオプションです。)アップグレードするデータベースの現在のOracleホーム(ORACLE\_HOME)。

- [source\\_ldap\\_admin\\_dir](#)  
(オプション)ソース・データベース・ホームのLDAP\_ADMINディレクトリへのパスを指定します。
- [source\\_tns\\_admin\\_dir](#)  
(オプション)ソース・データベース・ホームのTNS\_ADMINディレクトリへのパスを指定します。
- [start\\_time](#)  
(オプション)アップグレード・ジョブを実行する将来の起動時間を設定します。このパラメータは、アップグレード・ジョブをスケジュールしてサーバーの負荷を分散し、複数のジョブが即時に起動されないようにするために使用します。
- [target\\_base](#)  
(オプション)ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。
- [target\\_cdb](#)  
(オプション)非CDB Oracleデータベースが接続されるターゲットCDBのSIDを指定します。このパラメータは、非CDBのOracleデータベースをアップグレードおよび変換する場合に必須です。
- [target\\_pdb\\_copy\\_option=file\\_name\\_convert](#)  
(オプション)非CDBデータベースをPDBに変換する場合、または別のソースCDBの既存のPDBを指定のターゲットCDBのPDBに変換する場合にAutoUpgradeが実行する、プラグブル・データベースを作成する文で使用されるfile\_name\_convertオプションを指定します。
- [target\\_pdb\\_name](#)  
(オプション)ターゲットCDBに接続された後に、非CDBのソースOracleデータベースに割り当てる名前を指定します。
- [target\\_ldap\\_admin\\_dir](#)  
(オプション)ターゲット・データベース・ホームのLDAP\_ADMINディレクトリへのパスを指定します。
- [target\\_tns\\_admin\\_dir](#)  
(オプション)ターゲット・データベース・ホームのTNS\_ADMINディレクトリへのパスを指定します。
- [timezone\\_upg](#)  
(オプション) AutoUpgradeプロセスの一環としてタイムゾーンのアップグレードの実行を有効または無効にします。
- [tune\\_setting](#)  
(オプション)指定したワークフロー・オプションに応じて、実行時にAutoUpgradeの動作を変更する特別なワークフローを有効にします。
- [upgrade\\_node](#)  
(オプション)現在のユーザー構成が有効なノードを指定します。デフォルト値はlocalhostです。

親トピック: [Oracle Databaseのアップグレードの準備](#)

## add\_after\_upgrade\_pfile

(オプション)アップグレード後にパラメータを追加するPFILEのパスおよびファイル名を指定します。

例

```
sales3.add_after_upgrade_pfile=/path/to/my/pfile_add.ora
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## add\_during\_upgrade\_pfile

(オプション)アップグレード時にパラメータを追加するPFILEのパスおよびファイル名を指定します。

例

```
sales3. add_during_upgrade_pfile=/path/to/my/newpfile.ora
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## after\_action

(オプション) deployモードでは、接頭辞アドレスで識別されるデータベースのデプロイ・ジョブの完了後に実行するカスタム・アクションを指定します。

使用上のノート

使用するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)
- Oracle SQLファイル(.sql)。接頭辞によって指定されたローカル操作のみ。

デフォルトでは、スクリプトが失敗すると、AutoUpgradeが引き続き実行されます。オペレーティング・システムがスクリプトの失敗を検出した場合、Yフラグを使用してAutoUpgradeが停止するように指定します。スクリプトが0とは異なるステータスで終了すると、失敗して完了したとみなされます。

グローバルのafter\_actionパラメータとは対照的に、ローカルのafter\_actionパラメータはSQLスクリプトを指定でき、非CDBのOracleホーム上のターゲットOracleデータベース・バイナリまたはCDB\$ROOTを使用してデータベースで実行されます。追加のコンテナ固有のアクションを実行する場合、コード内でこれらを設定する必要があります。より複雑なシナリオでは、シェルでコンテナ固有のアクションを実行できます。

例

AutoUpgradeが処理を開始した後に、指定したスクリプトを実行します。スクリプトが失敗した場合にAutoUpgradeを停止するには、次のようにYフラグを設定します。

```
sales2. after_action=/user/path/script.sh Y
```

AutoUpgradeがデプロイ・オプションの処理を開始した後、指定したスクリプトを実行します。スクリプトが失敗した場合に実行を継続するには、次のようにAutoUpgradeを設定します。

```
sales3. after_action=/user/path/script.sh
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## before\_action

(オプション) deployモードでは、接頭辞で指定された特定のデータベース・ジョブのアップグレード・ジョブの起動前に実行するカ

スタム・アクションを指定します。すべてのアップグレード・ジョブの前にスクリプトを実行する場合は、ローカル・パラメータ (global.before\_action)を使用してそのスクリプトを指定します

#### 使用上のノート

使用するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)
- Oracle SQLファイル(.sql)。接頭辞によって指定されたローカル操作の場合のみ。

デフォルトでは、スクリプトが失敗すると、AutoUpgradeが引き続き実行されます。オペレーティング・システムがスクリプトの失敗を検出した場合、Yフラグを使用してAutoUpgradeが停止するように指定します。スクリプトが0とは異なるステータスで終了すると、失敗して完了したとみなされます。

グローバルbefore\_actionパラメータとは対照的に、ローカルbefore\_actionパラメータはSQLスクリプトを指定でき、これは前のリリースのOracle Databaseバイナリを使用して、ソース・データベースのOracleホームのデータベースで実行されます。スクリプトは非CDBのOracleホームまたはCDB\$ROOTで実行されます。追加のコンテナ固有のアクションを実行する場合、コード内でこれらを設定する必要があります。より複雑なシナリオでは、シェルでコンテナ固有のアクションを実行できます。

#### 例

AutoUpgradeがdeployモードの処理を開始する前に、指定したスクリプトを実行します。スクリプトが失敗した場合にAutoUpgradeを停止するには、次のようにYフラグを設定します。

```
sales.before_action=/user/path/script.sh Y
```

AutoUpgradeが処理を開始する前に、指定したスクリプトを実行します。スクリプトが失敗した場合実行を継続するには、次のようにAutoUpgradeを設定します。

```
sales4.before_action=/user/path/script.sh
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## catctl\_options

(オプション) AutoUpgradeがcatctl.plに送信してデフォルトの動作をオーバーライドするために選択できるcatctl.plオプションのセットを1つ以上指定します。

#### 使用上のノート

使用可能なcatctl.plオプション:

- -n パラレル操作に使用するプロセス数。リプレイ・アップグレードの場合、アップグレードに使用されるパラレル・プロセスの数は、CPU\_COUNTを4で割った値にデフォルト設定されます。クラシック・アップグレードの場合、CDB\$ROOTのデフォルトは8です。

- -N PDBのアップグレード時に使用するプロセッサの数。リプレイ・アップグレードの場合、アップグレードに使用されるパラレル・プロセスの数は、CPU\_COUNTを4で割った値にデフォルト設定されます。クラシック・アップグレードの場合、デフォルトは2です
- -t クラシック・アップグレードでSQLを実行し、デフォルトのリプレイ・アップグレード方法を上書きします
- -T オフラインのユーザー・スキーマベースの表領域を取得します。
- -z catcon.plの本番デバッグ情報を有効にします。

例

```
sales4.catctl_options=-t
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## checklist

(オプション)ポリシーまたはセキュリティ上の問題のために自動的に実装しない修正など、AutoUpgradeによって実行される修正のデフォルト・リストをオーバーライドするために使用できるチェックリストへのパスを指定します。

使用上のノート

他のAutoUpgradeモードでこのパラメータを使用するには、analyzeモードでAutoUpgradeを実行する必要があります。AutoUpgradeで分析が終了すると、事前チェック・ディレクトリ(dbname\_checklist.cfg)の下のデータベース名によって識別されるチェックリスト・ファイルを検索できます。ファイルを手動で更新して、AutoUpgradeでバイパスする修正を除外し、ファイルを新しい名前でも保存します。AutoUpgradeを再度実行すると、作成したチェックリスト・ファイルを指すパラメータを指定し、個々のデータベースに対して実行された修正を変更できます。チェックリスト・ファイル・パスを指定しない場合、アップグレード時に実行される修正のセットは、指定したデプロイ・モードで作成される最新バージョンのチェックリスト・ファイルです。

例

```
sales.checklist=/u01/app/oracle/upgrade-jobs/salesdb_checklist.cfg
```

前述の例では、salesdb\_checklist.cfgはデータベースsalesdbのチェックリスト構成ファイルです。

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## del\_after\_upgrade\_pfile

(オプション)アップグレード後にパラメータを削除するPFILEのパスおよびファイル名を指定します。

例

```
sales3.del_after_upgrade_pfile=/path/to/my/pfile_del.ora
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## del\_during\_upgrade\_pfile

(オプション)アップグレード時にパラメータを削除するPFILEのパスおよびファイル名を指定します。

例

```
sales3.del_during_upgrade_pfile=/path/to/my/oldpfile.ora
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## env

(オプション) ORACLE\_SID、ORACLE\_HOME、ORACLE\_BASEおよびTNS\_ADMINを除く、オペレーティング・システムに設定されているカスタム・オペレーティング・システム環境変数を指定します。

使用上のノート

このパラメータを使用して、データベースsqlnet.oraファイルに示されている環境設定(Oracleウォレットに使用されるセキュア・ソケット・レイヤー暗号スイートなど)を指定します。複数設定するには、カンマで区切ります。

構文:

```
prefix=VARIABLE1=value1 [, VARIABLE2=value2, ...]
```

例

PDB sales2の場合、WALLET\_LOCATIONの値はカスタム環境変数を使用して設定されるとします。

```
WALLET_LOCATION=
(SOURCE=
(METHOD=file)
(METHOD_DATA=(DIRECTORY=/databases/wallets/$CUSTOM_ENV1/$CUSTOM_ENV2))
```

その場合、AutoUpgradeでそれらのカスタム環境変数の値を認識するには、envパラメータ(dir1は環境変数CUSTOM\_ENV1によって指定されるパス、dir2はCUSTOM\_ENV2によって指定されるパス)を使用して指定する必要があります。

```
sales2.env=CUSTOM_ENV1=dir1,CUSTOM_ENV2=dir2
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## keep\_source\_pdb

(オプション)切断/接続アップグレード操作のソースPDBが、ソースCDBから削除されるのではなく、閉じた状態に保持されるかどうかを指定します。

使用上のノート

デフォルトでは、ソースPDBはアップグレード・プロセス中にソースCDBから削除されます。keep\_source\_pdbがYESに設定されている場合、ソースPDBは以前のリリース・システムから削除されません。パラメータtarget\_pdb\_copy\_optionでコピー・オプションが指定されている場合のみ、パラメータをYESに設定できます。コピー・オプションを使用しない場合、PDBを削除する必要がありますため、このパラメータは無視されます。コピーがない場合、既存のデータファイルは単一のCDBでのみ使用できます。

オプション

[yes | no]

デフォルト値はnoです。

例

```
sales1. keep_source_pdb=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## log\_dir

(オプション)パラメータの接頭辞で識別されるアップグレード・ジョブに含まれるデータベース・セットにあるデータベース・アップグレード用に生成されるログ・ファイルの場所を設定します。

使用上のノート

設定すると、AutoUpgradeは、指定したローカル・ログ・ファイル・パスに基づいて階層ディレクトリを作成します。たとえば、ジョブ識別子の接頭辞はsalesで、log\_dirはupgrade-jobsとして識別され、stage1、stage2およびstagenはアップグレードのステージを表します。

```
/u01/app/oracle/upgrade-jobs
                               /temp/
                               /sales/
                               /sales/stage1
                               /sales/stage2
                               /sales/stagen
```

チルダ(~)など、パスにワイルド・カードを使用することはできません。完全なパスを使用する必要があります。

例

```
salesdb. log_dir=/u01/app/oracle/upgrade-jobs
```

デフォルトでは、グローバル構成ファイル・パラメータglobal.autoupg\_log\_dirが指定され、log\_dirを指定しない場合、global.autoupg\_log\_dirで指定されたパスがデフォルトになります。

global.autoupg\_log\_dirもlog\_dirも指定されていない場合、デフォルトでは、ログ・ファイルは構成ファイルに含めるデータベースのorabaseユーティリティで示される場所に配置されます。その場合、デフォルトのログ・ディレクトリはパスORACLE\_BASE/cfgtoollogs/autoupgradeにあります。

構成ファイルに含まれるすべてのデータベースについてorabaseユーティリティが失敗した場合、ログ・ファイルの場所は、AutoUpgradeを実行しているユーザーのtempディレクトリに基づきます。

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## manage\_standbys\_clause

(オプション) DB\_UNIQUE\_NAMEで特定したスタンバイOracle Data Guardスタンバイ・データベースをAutoUpgradeプラグインのアップグレードから除外して、スタンバイ・データベース・ファイルを再利用できるようにするかどうかを指定します。

## 使用上のノート

デフォルトでは、非CDBおよびPDBの切断/接続のシナリオでは、ソース・データベースがOracle Data Guard構成の一部である場合、デフォルトの動作はソース・データベースのスタンバイ・ステータスを無視することです。このシナリオでは、AutoUpgradeでは、すべてのスタンバイ・データベースのファイルが、プラグイン・アップグレード時のプライマリ・データベース上のファイルと同じ状態である必要があり、REDO適用によってスタンバイ上のファイルが検索されます。

スタンバイ・データ・ファイルが同じ状態ではなく、非CDBまたはPDBソース・データベースがOracle Data Guard構成のスタンバイ・データベースであるスタンバイ・データベースのデータファイルを保持する場合、このパラメータを使用して、すべてのスタンバイ・データベース、または指定したデータベース(除外リスト)を除くすべてのデータベース、または指定したデータベースのみ(包含リスト)に対してREDO適用を遅延できます。

### ノート:



スタンバイ・データベースを使用してデータベース構成をアップグレードする前に、潜在的な問題を削減するために、スタンバイ・データベースで AutoUpgrade を分析モードで実行することをお勧めします。

### オプション

次の構文では、pdb-nameは、切断/接続アップグレードでターゲットCDBにアップグレードするソースPDBのDB\_UNIQUE\_NAMEです。

```
manage_standbys_clause=STANDBYS=[STANDBYS=NONE|STANDBYS=ALL|STANDBYS=ALL EXCEPT ('pdb-name', 'pdb-name', ...) |STANDBYS=('pdb-name', 'pdb-name', ...)]
```

デフォルト値はNONEです。

### 例

次の例では、Oracle Data Guardスタンバイのメンバーである非CDBまたはプラグブル・データベースはAutoUpgradeプラグインのアップデートから除外されません。

```
upg2. sid=cdb1
upg2. pdba=*
upg2. target_cdb=cdb21x
upg2. source_home=/source/18x
upg2. target_home=/target/21x
upg2. manage_standbys_clause=standbys=none
```

次の例では、すべてのスタンバイ・データベースでのデータ・ファイルへのREDOの適用が、すべてのAutoUpgradeプラグイン・アップグレードで遅延されます。

```
upg3. sid=cdb2
upg3. pdba=*
upg3. target_cdb=cdb21x
upg3. source_home=/source/18x
upg3. target_home=/target/21x
upg3. manage_standbys_clause=standbys=all
```

次の例では、AutoUpgradeプラグインのアップグレード中に、PDB cdb3\_stby\_1およびcdb3\_stby\_2を除くすべてのスタンバイPDBでデータ・ファイルへのREDOの適用が遅延されます。

```
upg4. sid=cdb3
upg4. pdba=*
upg4. target_cdb=cdb21x
upg4. source_home=/source/12.2x
upg4. target_home=/target/21x
upg4. manage_standbys_clause=standbys=all except (' cdb3_stby_1', ' cdb3_stby_2')
```

次の例では、AutoUpgradeプラグインのアップグレード中に、データ・ファイルへのREDOの適用がスタンバイPDB cdb4\_stby1でのみ遅延されます。

```
upg4. sid=cdb4
upg4. pdba=*
upg4. target_cdb=cdb21x
upg4. source_home=/source/12.2x
upg4. target_home=/target/21x
upg4. manage_standbys_clause=standbys=(' cdb4_stby_1')
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## pdba

(オプション)アップグレードを実行するPDBのリストを設定します。このパラメータは、マルチテナント・アーキテクチャ(CDB)データベースのアップグレードにのみ適用されます。非CDBデータベースを接続およびアップグレードしている場合、このパラメータは無視されます。

使用上のノート

PDBリストはカンマ区切りです。リストには、PDB名またはアスタリスク(\*)のいずれかを含めることができます。これは、AutoUpgradeの実行時にCDBでオープンしているすべてのPDBをアップグレードすることを示します。パラメータが指定されていない場合、デフォルト値は\*です。

ANALYZEモードで実行している場合、AutoUpgradeはマウント状態のPDBを無視します。

FIXUPS、DEPLOYまたはUPGRADEモードで実行している場合、AutoUpgradeは実行モードに応じて、マウント状態のPDBを読み取り/書き込みモード、アップグレード・モードまたはその両方で開きます。

例

```
sales1.pdba=pdb1, pdb2, pdbn
upgr1.pdba=*
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## raise\_compatible

(オプション)アップグレードが正常に完了した後、Oracle DatabaseのCOMPATIBLE初期化パラメータをターゲット・リリースのデ

フォルト値まで増やします。

使用上のノート

オプション:

- Y: COMPATIBLEパラメータ設定をターゲット・リリースまで増やします
- N: COMPATIBLEパラメータ設定をターゲット・リリースまで増やしません

デフォルトはNです。

注意:



- COMPATIBLE パラメータを増やした後、データベースのダウングレードはできません。
- COMPATIBLE パラメータを現在のリリース・レベルに設定するのは、アップグレードしたデータベースを完全にテストした後に行うことをお勧めします。
- autoupgrade コマンドライン・パラメータ restore に使用する値に関係なく、構成ファイル・パラメータ raise\_compatible の値を yes に設定した場合は、アップグレードを開始する前に、作成した保証付きリストア・ポイントを手動で削除する必要があります。アップグレードが正常に完了すると、アップグレードを開始する前に AutoUpgrade によって作成された保証付きリストア・ポイントは自動的に削除されます。AutoUpgrade によって POSTUPGRADE ステージが開始されると、データベースをリストアする方法はありません。

例

```
sales1. raise_compatible=yes
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## remove\_underscore\_parameters

(オプション)構成ファイル内のすべてのOracle Databaseについて、アップグレード中およびアップグレード後にPFILFILEファイルからアンダースコア(非表示)パラメータを削除します。

使用上のノート

アンダースコアのパラメータは、Oracleサポートの指示がある場合のみ使用できます。

オプション

[yes | no]

デフォルト値はnoです。

例

```
sales1. remove_underscore_parameters=yes
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## restoration

(オプション)データベースのリストアに保証付きリストア・ポイント(GRP)を生成します。

使用上のノート

restoration=noを設定する場合は、データベースのバックアップおよびリストアを手動で実行する必要があります。このオプションは、NOARCHIVELOGモードで動作するデータベース、およびOracle Flashback Technology機能のフラッシュバック・データベースをサポートしていないStandard EditionおよびStandard Edition 2データベースに使用します。パラメータを指定しない場合、デフォルト値(yes)が使用され、保証付きリストア・ポイントが作成されます。

オプション

[yes | no]

デフォルト値はyes。

例

```
sales1.restoration=no
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## revert\_after\_action

(オプション)接頭辞によって指定された特定のデータベース・ジョブに対してシステムのリストアが完了し、データベースが稼働した後にオペレーティング・システムで実行するカスタム・アクションを指定します。

使用上のノート

revert\_after\_actionで指定するアクションは、リストア・プロセスが完了し、データベースが稼働した後、ターゲットのOracleホーム・バイナリで実行されます。

実行するように指定するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)
- Oracle SQLスクリプト(.sql)。revert\_before\_actionパラメータ接頭辞で指定されたデータベースのローカル操作の場合。

オプション

失敗時に停止します: [Y|N]。デフォルトはNです。

デフォルトでは、指定したスクリプトが失敗すると、AutoUpgradeは引き続き実行されます(N)。スクリプトが失敗した場合にAutoUpgradeが停止するように指定するには、Yフラグを使用します。スクリプトが0以外のステータスでオペレーティング・システムで実行を終了した場合、AutoUpgradeはスクリプトを失敗として識別します。

例

AutoUpgradeがリストアの処理を完了した後に、オペレーティング・システムで指定したスクリプトを実行します。スクリプトが失敗した場合にAutoUpgradeを停止するには、Yフラグを設定します。

```
sales3.revert_after_action =/user/path/script.sh Y
```

AutoUpgradeがリストアの処理を完了した後に、オペレーティング・システムで指定したスクリプトを実行します。フラグを指定しないと、デフォルトの失敗時に停止するオプションはNであるため、スクリプトが失敗した場合、AutoUpgradeは引き続き実行されます。

```
sales3.revert_after_action =/user/path/script.sh
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## revert\_before\_action

(オプション)接頭辞によって指定された特定のデータベース・ジョブに対してシステムのリストアが完了し、データベースが稼働する前にオペレーティング・システムで実行するカスタム・アクションを指定します。

使用上のノート

revert\_before\_actionで指定するアクションは、データベースのリストアが開始されてデータベースが稼働する前に、ターゲットのOracleホーム・バイナリで実行されます。

実行するように指定するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)
- Oracle SQLスクリプト(.sql)。revert\_before\_actionパラメータ接頭辞で指定されたデータベースのローカル操作の場合。

オプション

失敗時に停止します: [Y|N]。デフォルトはNです。

デフォルトでは、指定したスクリプトが失敗すると、AutoUpgradeは引き続き実行されます(N)。スクリプトが失敗した場合にAutoUpgradeが停止するように指定するには、Yフラグを使用します。スクリプトが0以外のステータスでオペレーティング・システムで実行を終了した場合、AutoUpgradeはスクリプトを失敗として識別します。

例

AutoUpgradeがリストアを開始する前に、オペレーティング・システムで指定するスクリプトを実行します。スクリプトが失敗した場合にAutoUpgradeを停止するには、Yフラグを設定します。

```
sales3.revert_before_action =/user/path/script.sh Y
```

AutoUpgradeがリストアを開始する前に、オペレーティング・システムで指定したスクリプトを実行します。フラグを指定しないと、デフォルトの失敗時に停止するオプションはNであるため、スクリプトが失敗した場合、AutoUpgradeは引き続き実行されます。

```
sales3.revert_before_action =/user/path/script.sh
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## run\_utlrb

(オプション)アップグレードの一部としてutlrbを有効または無効にします。

使用上のノート

utlrbユーティリティは、データベース・アップグレード時に無効になるすべてのデータ・ディクショナリ・オブジェクトを再コンパイルします。Oracle Databaseのすべてのアップグレード後にこのユーティリティを実行することをお勧めします。

オプション

[yes | no]

デフォルト値はyes。

例

```
prefix.run_utlrb=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## sid

(必須)アップグレードするデータベースのOracleシステム識別子(SID)を示します。

例

```
sales1.sid=salesdb
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## skip\_tde\_key\_import

(オプション) yesに設定すると、アップグレードは実行されますが、ソース・データベースKeyStoreのターゲット・データベースへのインポートはスキップされ、エラーは発生しません。

使用上のノート

ノート:



このパラメータは不要になったため、非推奨です。今後のリリースのAutoUpgradeで削除される可能性があります。このパラメータを使用するかわりに、`-load_password` コマンドライン・オプションを使用して、AutoUpgradeの

キーストアに TDE パスワードを追加するか、セキュアな外部パスワード・ストア(SEPS)に TDE パスワードを追加することをお勧めします。

このオプションは、非CDBからPDBへの操作および切断/接続操作に使用できます。ソース・データベースKeyStoreをターゲット・データベースへのインポートをスキップする場合、AutoUpgradeではPDBをアップグレード・モードでオープンしたままにするため、キーを手動でインポートできます。キーをインポートした後、通常モードでデータベースを再起動する必要があります。

オプション

[yes | no]

デフォルト値はnoです。

例

```
sales1.skip_tde_key_import=yes
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## source\_base

(オプション)ソースOracleホームのソースORACLE\_BASEパスを指定します。

例

```
source_base=/u01/app/oracle  
sales4.source_base=/u04/app/oracle4
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## source\_dblink

(オプション)切断/接続再配置(ホット・クローン)アップグレード用に設定されたデータベース・リンクを指定します。

使用上のノート

非CDBまたはPDBに対して切断/接続再配置アップグレードを設定するには、まずソース・データベースとターゲット・データベースの場所間にデータベース・リンクを設定する必要があります。次に、source\_dblinkパラメータを使用して、そのデータベース・リンクをAutoUpgradeに渡します。データベース・リンクに関連付けられているソース・データベース名は、source\_dblinkパラメータの接尾辞として識別します。データベースがデータベース・リンクからリフレッシュされる時間値を秒単位で指定することもできます。



ノート:

このオプションは、Oracle Database 12.2 以降のソース・データベース・リリースで使用できます。

target\_pdb\_copy\_optionパラメータを使用すると、source\_dblinkパラメータがアクティブになります。source\_dblinkを使用する場合は、file\_name\_convertパラメータの値も指定して、ファイル名を変換するように指定するか、ファイル名を変換し

ないように指定する必要があります。file\_name\_convertがnoneに設定されている場合、データベース・ファイルを配置する場所を指定するためにdb\_create\_file\_destも設定する必要があります。

また、ソース・データベースのデータベース・リンクでターゲット・データベースが更新される頻度を指定するリフレッシュ間隔を秒単位で設定することもできます。start\_timeパラメータとともにリフレッシュ間隔を使用すると、ターゲットの場所に対してソース・データベースをリフレッシュするように維持できます。リフレッシュ率が指定されていない場合、ソース・データベースは1回のみクローニングされ、リフレッシュは行われません。リフレッシュ率が指定されているが、start\_timeパラメータを使用して将来の開始時間を指定しない場合、リフレッシュ間隔値は無視され、データベースは1回のみクローニングされます。

## オプション

- (必須)接尾辞として指定されたソース・データベース名。
- (必須)作成したデータベース・リンクの名前。
- (オプション)ソース・データベースからのターゲット・データベースへのリフレッシュ率(秒)。リフレッシュ率を指定する場合は、通常、start\_timeパラメータを使用して将来の開始時間も指定します。

## 例

次の例では、2つのデータベース・リンクが作成されます。

- pdbxcdb18x\_link。pdbxという名前のPDBソース・データベースに作成されます。

```
CREATE DATABASE LINK pdbxcdb18x_link CONNECT TO remote-user IDENTIFIED BY password
USING' (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST
GRANT CREATE SESSION, CREATE PLUGGABLE DATABASE, SELECT_CATALOG_ROLE TO remote-user;
GRANT READ ON sys. enc$ TO remote-user;
```

- db18x\_link。db18xという名前の非CDBソース・データベースに作成されます。

```
CREATE DATABASE LINK db18x_link CONNECT TO remote-user IDENTIFIED BY password
USING' (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = db-node1) (PORT = 1521))
(CONNECT_DATA = (SERVICE_NAME = db18x)))';
```

AutoUpgrade構成ファイルでは、データベース・リンクに関連付けられたデータベース名は、その名前をsource\_dblinkの接尾辞として使用して指定します(PDBソース・データベースの場合、接尾辞はpdbxで、非CDBソース・データベースの場合、接尾辞はdb18x)。

次の例では、source\_dblinkを使用して、ソース・データベースpdbxのdblinkを指定します。時間間隔が指定されていないため、AutoUpgradeの起動直後にPDBアップグレード・デプロイメントが開始されます。

```
upg1.source_dblink.pdbx=pdbxcdb18x
```

同じ構成ファイルを使用して、AutoUpgradeは、AutoUpgradeがコマンドラインから起動された1時間40分後に、db18xという名前のデータベースのアップグレードを開始します。AutoUpgradeが起動されてからstart\_timeで指定されたデプロイメント時間までの間、クローニングされたターゲット・データベースはソースから20秒ごとにリフレッシュされます。

```
upg1.source_dblink.db18x=db18x_link 20
upg1.start_time=+1h40m
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## source\_home

(分析モード、修正モードおよびデプロイ・モードの場合は必須です。アップグレード・モードの場合はオプションです。)アップグレードするデータベースの現在のOracleホーム(ORACLE\_HOME)。

使用上のノート

upgradeモードの場合、ソース・ホームとターゲット・ホームの値は同じパスにできます。

例

```
sales2. source_home=/path/to/my/source/oracle/home
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## source\_ldap\_admin\_dir

(オプション)ソース・データベース・ホームのLDAP\_ADMINディレクトリへのパスを指定します。

使用上のノート

WindowsではLDAP\_ADMIN環境変数がレジストリ内で設定されているため、このパラメータはMicrosoft Windowsには影響しません。

例

```
sales1. source_ldap_admin_dir=/u01/app/oracle/12.2/dbhome01/ldap/admin
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## source\_tns\_admin\_dir

(オプション)ソース・データベース・ホームのTNS\_ADMINディレクトリへのパスを指定します。

使用上のノート

WindowsではTNS\_ADMIN環境変数がレジストリ内で設定されているため、このパラメータはMicrosoft Windowsには影響しません。

例

```
sales1. source_tns_admin_dir=/u01/app/oracle/12.2/dbhome01/network/admin
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## start\_time

(オプション)アップグレード・ジョブを実行する将来の起動時間を設定します。このパラメータは、アップグレード・ジョブをスケジュールしてサーバーの負荷を分散し、複数のジョブが即時に起動されないようにするために使用します。

## 使用上のノート

値は、now (すぐに開始)の形式にするか、英語日付書式の形式DD/MM/YYYYまたはMM/DD/YYYYにする必要があります。ここで、MMは月、DDは日、YYYYは年です。値を設定しない場合、デフォルトはnowです。

設定できる値:

```
now
30/12/2019 15:30:00
01/11/2020 01:30:15
2/5/2020 3:30:50
```

start\_time値をnowに設定して複数のジョブが起動されると、AutoUpgradeではシステムで使用可能なリソースに基づいて起動時間をスケジュールします。ジョブの起動時間は数分単位でずれる場合があります。

日付要素または時刻要素に間違ったデリミネータを使用している値、または次のような間違った日付書式または時間書式を使用している値は無効です。

```
30-12-2019 15:30:00
01/11/2020 3:30:15pm
2020/06/01 01:30:15
```

例

```
sales1.start_time=now
sales2.start_time=07/11/2020 01:30:15
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_base

(オプション)ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。

例

```
target_base=/u01/app/oracle
sales4.target_base=/u04/app/oracle4
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_cdb

(オプション)非CDB Oracle Databaseが接続されるターゲットCDBのSIDを指定します。このパラメータは、非CDBのOracleデータベースをアップグレードおよび変換する場合に必須です。

例

```
emp.target_cdb=salescdb
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_pdb\_copy\_option=file\_name\_convert

(オプション)非CDBデータベースをPDBに変換する場合、または別のソースCDBの既存のPDBを指定のターゲットCDBのPDBに変換する場合にAutoUpgradeが実行する、プラグブル・データベースを作成する文で使用されるfile\_name\_convertオプションを指定します。

使用上のノート

注意:



target\_pdb\_copy\_option を指定すると、AutoUpgrade は必要に応じてリカバリを管理できます。target\_pdb\_copy\_option が設定されておらず、デフォルトの nocopy オプションが使用されている場合、デフォルトの PDB はリカバリされません。必ず、ソース PDB をバックアップしてください。

このオプションは、ターゲットCDB内にプラグブル・データベースを作成する場合にのみ使用します。このパラメータを指定しない場合、パラメータのデフォルト値はNOCOPYであり、ソース・データベースの既存のデータ・ファイルが再利用されます。このパラメータを指定する場合は、ソース・データベース名またはPDB名を指定するパラメータに接尾辞を追加し

(target\_pdb\_copy\_option.suffix)、file\_name\_convert=を次のいずれかのオプションとともに指定する必要があります。

- ソース・ファイル名(f)およびターゲット置換ファイル名(r)を指定するか('f', 'r')、NONEを指定します
- リフレッシュ可能クローン・データベースを作成する場合は、ソース・データベース名またはPDB名を指定するパラメータに接尾辞を追加します(target\_pdb\_copy\_option.suffix)

ターゲットCDBで、パラメータDB\_CREATE\_FILE\_DESTまたはPDB\_FILE\_NAME\_CONVERTを設定し、ターゲットCDBでこれらのパラメータを置換ファイル名に対して有効にする場合は、値prefix.target\_pdb\_copy\_option.source-db-name-or-pdb=file\_name\_convert=NONEを設定します。

ターゲットCDBでの変換時に1つ以上のデータ・ファイル名を変更する場合、構文

prefix.target\_pdb\_copy\_option.source-db-name-or-pdb=('f1', 'r1', 'f2', 'r2', ...)を使用してパラメータの値を入力し、ソース・データベース名またはPDBを接尾辞として、そして変更するソース・ファイル名および既存のファイルのコピー先となるターゲット・ファイル名を指定します(f1は、ソースの最初のファイル名パターン、r1はターゲットCDBの最初の置換ファイル名パターン、f2はソースの2番目のファイル名パターン、r2はターゲットCDBの2番目の置換ファイル名パターン、となっていくます)。

構文

```
prefix.target_pdb_copy_option.source-db-name-or-pdb=file_name_convert=('f1', 'r1', 'f2', 'r2', 'f3', 'r3'...)
```

例

この例では、AutoUpgradeは、接頭辞文字列upg1および接尾辞salesを付けて、指定されたデータベースの変換時に既存のデータファイルをコピーし、ファイルパス文字列およびファイル名/old/path/pdb\_2をファイルパス文字列およびファイル名/new/path/depsalesで置換します。

```
upg1.target_pdb_copy_option.sales=file_name_convert=('/old/path/pdb_2', '/new/path/depsales')
```

target\_pdb\_copy\_optionsource-db-name-or-pdb=file\_name\_convertを使用してOMFファイルを変換するには、ターゲットOracleホームがOracle Database 19cリリース更新6以降(19.6.0)またはOracle Database 18cリリース更新10以降(18.10.0)である必要があります。

この例では、Oracle ASMに格納されているデータ・ファイルの内、Oracle管理ファイルとして格納されていないデータ・ファイルを+DATA/dbname/salesから+DATA/dbname/depsalesにコピーするように、パラメータが構成されています。

```
upg1.target_pdb_copy_option.sales=file_name_convert=(' +DATA/dbname/sales', '+DATA/dbname/depsales')
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_pdb\_name

(オプション)ターゲットCDBに接続された後に、非CDBのソースOracle Databaseに割り当てる名前を指定します。

使用上のノート

このパラメータは、非CDB Oracle DatabaseをアップグレードしてPDBに変換する場合にのみ使用されます。

非CDBをPDBに変換する場合のターゲットPDBのデフォルト名は、非CDB Oracle Databaseのデータベースの一意の名前を使用します。CDBに接続するときに非CDBの既存の名前とは異なる名前を指定する場合は、このパラメータを設定する必要があります。さらに、リフレッシュ可能クローン・データベースを作成する場合は、ソース・データベース名またはPDB名を指定するパラメータに接尾辞を追加します(target\_name.suffix)

例

```
upg.target_cdb=target_pdb_name-emp1pdb
```

リフレッシュ可能クローンの場合は、接尾辞を追加します。

```
upg1.target_cdb.emp=target_pdb_name-emp1pdb
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_ldap\_admin\_dir

(オプション)ターゲット・データベース・ホームのLDAP\_ADMINディレクトリへのパスを指定します。

例

```
sales1.target_ldap_admin_dir=/u01/app/oracle/19/dbhome01/ldap/admin
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## target\_tns\_admin\_dir

(オプション)ターゲット・データベース・ホームのTNS\_ADMINディレクトリへのパスを指定します。

例

```
sales1.target_tns_admin_dir=/u01/app/oracle/19/dbhome01/network/admin
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## timezone\_upg

(オプション)AutoUpgradeプロセスの一部としてタイムゾーンのアップグレードの実行を有効または無効にします。

使用上のノート

データ整合性を維持するには、データベースのアップグレード時にタイムゾーン設定をアップグレードすることをお勧めします。特に、timestamp with time zone表の列など、タイムゾーンに依存するデータがある場合、タイムゾーンをアップグレードします。この設定は、チェックリスト・ファイルで修正を上書きすることで無効にできることに注意してください。

オプション

[yes | no]

デフォルト値はyes。

例

```
sales1.timezone_upg=yes
```

親トピック: [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## tune\_setting

(オプション)指定したワークフロー・オプションに応じて、実行時にAutoUpgradeの動作を変更する特別なワークフローを有効にします。

使用上のノート

tune\_settingパラメータを使用すると、アップグレード・ステップ、またはAutoUpgrade構成ファイルのパラメータ接頭辞で指定されたコンテナ・データベースまたはプラグブル・データベース(CDBまたはPDB)によって指定されたアップグレードの処理に割り当てられたリソースを微調整できます。この機能は、デフォルトのAutoUpgrade値がシステム要件に十分でないか、デフォルト以外のAutoUpgradeオプションを有効にする場合に、一部のアップグレードに役立ちます。

構文

```
prefix.tune_setting=option[, option, option, ...]
```

次のリストから、必要なAutoUpgradeランタイム・チューニングを提供するtune\_settingオプションを選択します。複数のチューニング・オプションをtune\_settingパラメータと組み合わせるには、カンマ区切りを使用します。例:

```
sales3.tune_setting=proactive_fixups=true,query_hint_parallel=8,utltp_threads_per_pdb=8
```



ノート:

オプション	説明
active_nodes_limit	<p>Oracle Real Application Clusters データベースの分散アップグレード中に使用するアクティブなクラスタ・メンバー・ノードの新しい合計を設定します。デフォルトは 2 です。指定した数がクラスタ・メンバー・ノードの最大数以上である場合、すべてのノードが取得されます。</p> <p>sales3. tune_setting=active_nodes_limit=4</p>
distributed_upgrade	<p>AutoUpgrade が<b>分散アップグレード</b>を実行することを指定します。分散アップグレードでは、Oracle Clusterware クラスタ・メンバー・ノードのリソースを利用して、クラスタ上の PDB のアップグレードをより迅速に実行します。2 つ以上のノードの Oracle RAC クラスタ内の CDB をアップグレードする場合、このオプションを使用します。このオプションを選択すると、proactive_fixups オプションもデフォルトで有効になります。</p> <p>例:</p> <p>sales3. tune_setting=distributed_upgrade=true</p>
make_pdbs_available	<p><b>事前修正</b>モードを使用するときに、事後修正の完了後、接頭辞で指定された PDB を読取り/書込みモードおよび非制限モードでオープンします。このオプションを使用すると、アップグレードの完了直後に接頭辞で指定された PDB がサービスで使用可能になり、他の PDB が引き続きアップグレードされます。これは大規模なフリート・アップグレード・デプロイメントに役立ちます。</p> <p>注意:</p> <p>このオプションを選択すると、指定した PDB はユーザーからのサービス・リクエストを受け入れ、他の PDB は引き続きアップグレードされます。サービス・リクエストに対する PDB のレスポンス時間、および進行中の PDB アップグレードに必要な時間は、それぞれ影響を受ける可能性があります。</p> <p>例:</p> <p>sales3. tune_setting=make_pdbs_available=true</p>
proactive_fixups	<p>アップグレードの最後のステージとして PDB がアップグレードされる、事前修正モードを有効にします。PDB の数がデータ</p>

オプション	説明
	<p>ベースで定義されている CPU 数を 2 で割った数より多い場合、このチューニング・オプションを選択すると、アップグレードが速くなります。例：</p> <pre>sales3. tune_setting=proactive_fixups=true</pre> <p>注意：</p> <p>CPU の数が PDB の数より多い場合、この設定を変更すると、パフォーマンスが向上しない可能性があります。</p>
query_hint_parallel	<p>接頭辞で指定された PDB の問合せ中に表領域からデータを収集するコードにパラレル・スレッド指定を指定します。これにより、接頭辞で指定された PDB にパラレル・スレッド数を増減して割り当てることができます。例：</p> <pre>sales3. tune_setting=query_hint_parallel=8</pre> <p>このオプションを選択すると、AutoUpgrade によってシステム・リソースの消費量が増える可能性があります。</p>
utlrp_threads_per_pdb	<p>CDB 内の無効なオブジェクトの再コンパイルによって生成されるデフォルトの最大スレッド数を上書きし、指定したスレッド数を使用します。例：</p> <pre>sales3. tune_setting=utlrp_threads_per_pdb=8</pre> <p>注意：</p> <p>指定されたスレッド数がシステムで使用可能なスレッド数を超えると、パフォーマンスが低下する可能性があります。</p>
utlrp_pdb_in_parallel	<p>デフォルトの最大同時再コンパイル・スレッド数を、指定した数に上書きします。このオプションを使用して、無効なオブジェクトの再コンパイルのデフォルトの最大同時プロセス数を上書きします。例：</p> <pre>sales3. tune_setting=utlrp_pdb_in_parallel=2</pre> <p>注意：</p> <p>各 PDB プロセスでは、utlrp_threads_per_pdb で指定された数と同じ数のスレッドがシステムで必要です。</p>

例

次の例では、接頭辞sales3で指定されたデータベースのアップグレードは、Oracle Real Application ClustersのOracle Databaseインスタンスです。tune\_settingパラメータを使用して、Oracle Grid Infrastructureクラスタ内の複数のCDB間でアップグレード・ロードを分散する設定distributed\_upgradeを使用するように、これらのデータベース・インスタンスを設定します。

```
sales3.tune_setting=distributed_upgrade=true
```

次の例では、接頭辞sales3で指定されたデータベースのアップグレードは、複数のtune\_settingパラメータ・オプションでチューニングされます。

```
sales3.tune_setting=proactive_fixups=true,query_hint_parallel=8,utltp_threads_per_pdb=8
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

## upgrade\_node

(オプション)現在のユーザー構成が有効なノードを指定します。デフォルト値はlocalhostです。

使用上のノート

このパラメータの目的は、AutoUpgradeで使用する構成ファイルにリストされているデータベースがAutoUpgradeで処理されないようにすることです。ここで、upgrade\_nodeパラメータの値は現在のホスト名に対応していません。AutoUpgradeをリモートで実行することはできません。キーワードlocalhostをワイルド・カードとして使用して、ローカル・ホストのデータベースを処理するように指定できます。

ユースケース:

構成ファイルconfig.cfgには、10個のデータベースが含まれています。そのデータベースのうちの5つは、upgrade\_nodeの値がdenver01に設定されています。残りの5つは、upgrade\_nodeの値がdenver02に設定されています。AutoUpgradeが構成ファイルconfig.cfgを使用してサーバーdenver01で実行される場合、AutoUpgradeはupgrade\_nodeがdenver01に設定されているデータベースのみを処理します。upgrade\_nodeがdenver02に設定されているデータベースは無視されます。

hostnameユーティリティは、アップグレード・ノードを解決するために使用される値を識別します

例

```
hostname
denver02
sales1.upgrade_node=denver01
```

**親トピック:** [AutoUpgrade構成ファイルのローカル・パラメータ](#)

# AutoUpgradeユーザー構成ファイルのグローバル・パラメータ

構成ファイルで扱われるすべてのOracle Databaseアップグレードのパラメータに対するデフォルトの動作を指定するには、オプションのAutoUpgradeグローバル・パラメータを使用できます。

使用上のノート

グローバル・パラメータは、アップグレードまたはデプロイ・モードを使用する場合のtarget\_homeを除き、すべてオプションです。すべてのグローバル・パラメータは接頭辞globalを使用します。

add\_after\_upgrade\_pfileおよびdel\_during\_upgrade\_pfileグローバルおよびローカルのPFILEパラメータ操作は、次の階層順序で実行されます。

1. グローバル・アクション
    - a. グローバルの削除
    - b. グローバルの追加
  2. ローカル・アクション
    - a. ローカルの削除
    - b. ローカルの追加
- [add\\_after\\_upgrade\\_pfile](#)  
(オプション) PFILEをアップグレードした後にパラメータを追加するPFILEのパスおよびファイル名を指定します。
  - [add\\_during\\_upgrade\\_pfile](#)  
(オプション) PFILEのアップグレード中にパラメータを追加するPFILEのパスおよびファイル名を指定します。
  - [after\\_action](#)  
(オプション)すべてのアップグレード・ジョブが正常に終了した後に実行するカスタム・ユーザー・スクリプトのパスおよびファイル名を指定します。
  - [autoupg\\_log\\_dir](#)  
(オプション) ログ・ファイルと、AutoUpgradeが使用するグローバル・モジュールに属する一時ファイルの場所を設定します。
  - [before\\_action](#)  
(オプション)アップグレード・ジョブを開始する前にすべてのアップグレードに対して実行するカスタム・ユーザー・スクリプトを指定します。
  - [catctl\\_options](#)  
(オプション) AutoUpgradeがcatctl.plに送信してデフォルトの動作をオーバーライドするために選択できるcatctl.plオプションのセットを1つ以上指定します。
  - [del\\_after\\_upgrade\\_pfile](#)  
(オプション) PFILEのアップグレード後にパラメータを削除するPFILEのパスおよびファイル名を指定します。
  - [del\\_during\\_upgrade\\_pfile](#)  
(オプション) PFILEのアップグレード中にパラメータを削除するPFILEのパスおよびファイル名を指定します。
  - [drop\\_grp\\_after\\_upgrade](#)  
(オプション)データベースのアップグレード後に、保証付きリストア・ポイント(GRP)を削除します。
  - [keystore](#)  
(オプション) AutoUpgradeがパスワードやその他の機密情報を格納するために排他的に使用する専用ソフトウェア・

キーストアの場所を指定します。

- [raise\\_compatible](#)  
(オプション)アップグレードが正常に完了した後、compatibleパラメータをターゲット・リリースのデフォルト値まで増やします。
- [target\\_base](#)  
(オプション)ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。
- [target\\_home](#)  
(分析および修正モードの場合はオプションです。アップグレードおよびデプロイ・モードの場合は必須です。)構成ファイルで指定されたすべてのデータベースのグローバル・ターゲット・ホームを設定します。
- [target\\_version](#)  
(オプション) AutoUpgradeでアップグレードを実行するターゲット・リリース・バージョンを指定します。
- [upgradexml](#)  
(オプション) upgrade.xml ファイルを生成します。

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

## add\_after\_upgrade\_pfile

(オプション) PFILEをアップグレードした後にパラメータを追加するPFILEのパスおよびファイル名を指定します。

使用上のノート

この指定は、ユーザー構成ファイル内のすべてのデータベースに適用されます。

例

```
global.add_after_upgrade_pfile=/path/to/my/add_after.ora
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## add\_during\_upgrade\_pfile

(オプション) PFILEのアップグレード中にパラメータを追加するPFILEのパスおよびファイル名を指定します。

使用上のノート

この指定は、ユーザー構成ファイル内のすべてのデータベースに適用されます。

例

```
global.add_during_upgrade_pfile=/path/to/my/add_during.ora
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## after\_action

(オプション)すべてのアップグレード・ジョブが正常に終了した後に実行するカスタム・ユーザー・スクリプトのパスおよびファイル名を指定します。

## 使用上のノート

使用するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)

デフォルトでは、スクリプトが失敗すると、AutoUpgradeが引き続き実行されます。オペレーティング・システムがスクリプトの失敗を検出した場合、Yフラグを使用してAutoUpgradeが停止するように指定します。スクリプトが0とは異なるステータスで終了すると、失敗して完了したとみなされます。

### 例

スクリプトが失敗した場合、AutoUpgradeを停止します。

```
global.after_action=/path/to/my/script.sh Y
```

スクリプトが失敗した場合、AutoUpgradeを続行します。

```
global.after_action=/path/to/my/script.sh
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## autoupg\_log\_dir

(オプション) AutoUpgradeが使用するログ・ファイルとグローバル・モジュールに属する一時ファイルの場所を設定します。

### 使用上のノート

特定の接頭辞のログ・ディレクトリにあるuserconfigファイルに、異なるログ・ディレクトリ・パスを構成できます

このパラメータをパスに設定しない場合、デフォルトでは、ログ・ファイルは構成ファイルに含めるデータベースのorabaseユーティリティで示される場所に配置されます。その場合、デフォルトのログ・ディレクトリはパス

ORACLE\_BASE/cfgtoollogs/autoupgradeにあります。

構成ファイルに含まれるすべてのデータベースについてorabaseユーティリティが失敗した場合、ログ・ファイルの場所は、AutoUpgradeを実行しているユーザーのtempディレクトリに基づきます。

### 例

```
global.autoupg_log_dir=/path/to/my/global/log/dir
```

特定の接頭辞のログ・ディレクトリにあるuserconfigファイルに、異なるログ・ディレクトリ・パスを構成します

```
global.autoupg_log_dir=/path/to/my/global/log/dir  
myprefix_log_dir=global.auto_log_dir:different/path
```

この構文が使用されると、ログ・ファイルおよび一時ファイルは、接頭辞myprefixで識別されるデータベースの次のパスに配置さ

れます。

```
/path/to/my/global/log/dir/different/path
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## before\_action

(オプション) アップグレード・ジョブを開始する前にすべてのアップグレードに対して実行するカスタム・ユーザー・スクリプトを指定します。

使用上のノート

使用するスクリプトはname.ext (例: myscript.sh)の形式である必要があります。これにより、AutoUpgradeは実行するスクリプトのタイプを識別できます。特定のアップグレード・ジョブの前にスクリプトを実行する場合は、ローカル・パラメータ(local.before\_action)を使用してそのスクリプトを指定します

許可される拡張子オプションは、次のとおりです。

- UNIXシェル(.sh)
- Microsoft Windowsバッチ(.bat、.cmd)
- Microsoft Windows PowerShell (.ps1)

デフォルトでは、スクリプトが失敗すると、AutoUpgradeが引き続き実行されます。オペレーティング・システムがスクリプトの失敗を検出した場合、Yフラグを使用してAutoUpgradeが停止するように指定します。スクリプトが0とは異なるステータスで終了すると、失敗して完了したとみなされます。

例

スクリプトが失敗した場合、AutoUpgradeを停止します。

```
global.before_action=/path/to/my/script.sh Y
```

スクリプトが失敗した場合、AutoUpgradeを続行します。

```
global.before_action=/path/to/my/script.sh
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## catctl\_options

(オプション) AutoUpgradeがcatctl.plに送信してデフォルトの動作をオーバーライドするために選択できるcatctl.plオプションのセットを1つ以上指定します。

使用上のノート

オプションの詳細は、「[パラレル・アップグレード・ユーティリティ\(catctl.pl\)のパラメータ](#)」を参照してください。

オプション

使用可能なcatctl.plオプション:

- -n 並列操作に使用するプロセス数。リプレイ・アップグレードの場合、アップグレードに使用される並列・プロセスの数は、CPU\_COUNTを4で割った値にデフォルト設定されます。クラシック・アップグレードの場合、CDB\$ROOTのデフォルトは8です。
- -N PDBのアップグレード時に使用するプロセッサの数。リプレイ・アップグレードの場合、アップグレードに使用される並列・プロセスの数は、CPU\_COUNTを4で割った値にデフォルト設定されます。クラシック・アップグレードの場合、デフォルトは2です
- -t クラシック・アップグレードでSQLを実行し、デフォルトのリプレイ・アップグレード方法を上書きします
- -T オフラインのユーザー・スキーマベースの表領域を取得します。
- -z catcon.pmの本番デバッグ情報を有効にします。

例

```
global.catctl_options=-t -n 24 -N 4
```

## 関連項目

- [並列・アップグレード・ユーティリティ\(catctl.pl\)のパラメータ](#)
- [catcon.plの構文およびパラメータ](#)

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## del\_after\_upgrade\_pfile

(オプション) PFILEのアップグレード後にパラメータを削除するPFILEのパスおよびファイル名を指定します。

使用上のノート

この指定は、ユーザー構成ファイル内のすべてのデータベースに適用されます。

例

```
global.del_after_upgrade_pfile=/path/to/my/del_after.ora
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## del\_during\_upgrade\_pfile

(オプション) PFILEのアップグレード中にパラメータを削除するPFILEのパスおよびファイル名を指定します。

使用上のノート

この指定は、ユーザー構成ファイル内のすべてのデータベースに適用されます。

例

```
global.del_during_upgrade_pfile=/path/to/my/del_during.ora
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## drop\_grp\_after\_upgrade

(オプション)データベースのアップグレード後に、保証付きリストア・ポイント(GRP)を削除します。

使用上のノート

このオプションを選択した場合、アップグレードが正常に完了した後にGRPが削除されます。raise\_compatibleをyesに設定する場合は、パラメータdrop\_grp\_after\_upgradeもyesに設定する必要があります。

オプション

[yes | no]

デフォルト値はnoです。

例

```
global.drop_grp_after_upgrade=yes
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## keystore

(オプション) AutoUpgradeがパスワードやその他の機密情報を格納するために排他的に使用する専用ソフトウェア・キーストアの場所を指定します。

使用上のノート

キーストア・パラメータを使用して、AutoUpgradeが排他的に使用する専用ソフトウェア・キーストアを作成する場所を指定できます。

AutoUpgradeキーストアには、ファイルwallet.p12が含まれます(データベースで使用される他の種類のキーストアと同様)。ファイルは、TDEプロンプトでsaveコマンドを使用したときに作成されます。自動ログイン・キーストアの生成を選択すると、ファイルcwallet.ssoも作成されます。自動ログイン・キーストアがある場合、AutoUpgradeはAutoUpgradeの起動時にキーストア・パスワードの入力を求めません。

AutoUpgradeによって生成されたキーストアには機密情報が含まれ、キーストアを初めて使用するときに選択したパスワードによって保護されます。キーストアに変更が加えられるたびに、パスワードを指定する必要があります。AutoUpgradeの自動ログイン・キーストアを作成しない場合、AutoUpgradeを起動したときにAutoUpgradeでキーストアからの情報が求められ、キーストア・パスワードを指定する必要があります。

注意:



global.keystore で指定するディレクトリにはソフトウェア・キーストアが含まれているため、他のすべての高度にセキュアなキーストア・ファイルと同じセキュリティのベスト・プラクティスを使用して保護する必要があります。

例

次の例では、ORACLE\_SIDを、キーストアを使用するデータベースのシステム識別子に置き換えます。

親トピック: [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## raise\_compatible

(オプション)アップグレードが正常に完了した後、compatibleパラメータをターゲット・リリースのデフォルト値まで増やします。

使用上のノート

このオプションを選択した場合、アップグレードが正常に完了した後にGRPが削除されます。raise\_compatibleをyesに設定する場合は、パラメータdrop\_grp\_after\_upgradeもyesに設定する必要があります。



注意:

- COMPATIBLEパラメータを増やした後、データベースのダウングレードはできません。
- COMPATIBLEパラメータを現在のリリース・レベルに設定するのは、アップグレードしたデータベースを完全にテストした後のみ行うことをお勧めします。
- autoupgradeコマンドライン・パラメータrestoreに使用する値に関係なく、構成ファイル・パラメータraise\_compatibleの値をyesに設定した場合は、アップグレードを開始する前に、作成した保証付きリストア・ポイントを手動で削除する必要があります。アップグレードが正常に完了すると、アップグレードを開始する前にAutoUpgradeによって作成された保証付きリストア・ポイントは自動的に削除されます。AutoUpgradeによってPOSTUPGRADEステージが開始されると、データベースをリストアする方法はありません。
- raise\_compatibleをyesに設定する場合は、パラメータdrop\_grp\_after\_upgradeもyesに設定する必要があります。

オプション

[yes | no]

デフォルト値はnoです。

例

```
global.raise_compatible=yes
```

親トピック: [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## target\_base

(オプション)ターゲットOracleホームのターゲットORACLE\_BASEパスを指定します。

使用上のノート

このパラメータを使用する必要があるのは、まれな場合のみです。

例

```
global.target_base=/u01/app/oracle
sales4.target_base=/u04/app/oracle4
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## target\_home

(分析および修正モードの場合はオプションです。アップグレードおよびデプロイ・モードの場合は必須です。)構成ファイルで指定されたすべてのデータベースのグローバル・ターゲット・ホームを設定します。

使用上のノート

同じtarget\_homeを複数回指定しないようにするには、このオプションを使用します。このパラメータはローカルで上書きできます。

例

```
global.target_home=/target/Oracle/home
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## target\_version

(オプション)AutoUpgradeでアップグレードを実行するターゲット・リリース・バージョンを指定します。

使用上のノート

AutoUpgradeでは、このパラメータで指定したリリース・バージョン情報を使用して、アップグレード対象のターゲットのOracle Databaseリリースに対して正しいチェックおよび修正が使用されるようにします。このパラメータの形式は、有効なOracleバージョンのピリオド区切りの値です。

有効な値

- 12.2
- 18
- 19
- 21

このオプションは、ターゲット・ホームがシステムに存在しないか、ターゲット・ホームが12.2リリースの場合のみ必要です。それ以外の場合、AutoUpgradeはターゲット・リリース値を導出できます。

例

```
global.target_version=19
employees.target_version=12.2
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

## upgradexml

(オプション) upgrade.xml ファイルを生成します。

使用上のノート

生成されたupgrade.xmlは、XMLパラメータを指定したときにアップグレード前パッケージが生成した以前のリリースのファイルと同等のものです。このファイルは、分析モード(mode -analyze)中に作成されます。これは、AutoUpgradeログ・ファイル用に定義された事前チェック・ディレクトリに生成されます。

#### オプション

[yes | no]

デフォルト値はnoです。

#### 例

```
global.upgradexml=yes
```

**親トピック:** [AutoUpgradeユーザー構成ファイルのグローバル・パラメータ](#)

# AutoUpgradeを使用した非CDBからPDBへのアップグレードの理解

1回の操作で非CDBを新しいCDBのPDBにアップグレードおよび変換するか、非CDBデータベースを既存のCDBのPDBにアップグレードしてから変換できます。

Oracle Database 21cへのすべてのアップグレードで、マルチテナント・アーキテクチャを使用する必要があります。非CDB Oracle Databaseアーキテクチャの使用はサポートされなくなりました。非CDBアーキテクチャからPDBにデータベースを移行する場合、マルチテナント・ライセンスを必要とせずに、コンテナ・データベース(CDB)で最大3つのユーザー構成可能なPDBを取得します。4つ以上のPDBを構成することを選択する場合は、マルチテナント・ライセンスが必要です。

AutoUpgradeユーティリティの非CDBからPDBへの機能には、マルチテナント・アーキテクチャへのアップグレードおよび変換時に以前のリリースの非CDB Oracle Databaseをアップグレードする方法を制御する柔軟なオプションが用意されています。Oracle Database 21c以降、既存のターゲット・リリースのCDBがある場合、AutoUpgradeを使用して、アップグレード中に非CDBのOracle Databaseをターゲット・リリースのCDBのPDBに変換できるようになりました。非CDBからPDBへのアップグレードおよび変換を実行するには、非CDBに関する情報をAutoUpgrade構成ファイルに指定します。必要に応じて、非CDBのOracle Databaseをソース・リリースのPDBに変換し、PDBをターゲット・リリースのCDBに接続することもできます。この場合、PDBを接続すると、アップグレードが実行されます。

アップグレード後、データベース・リスナーおよびローカル・ネーミング・パラメータ(tnsnames.oraファイル)を構成する必要があります。

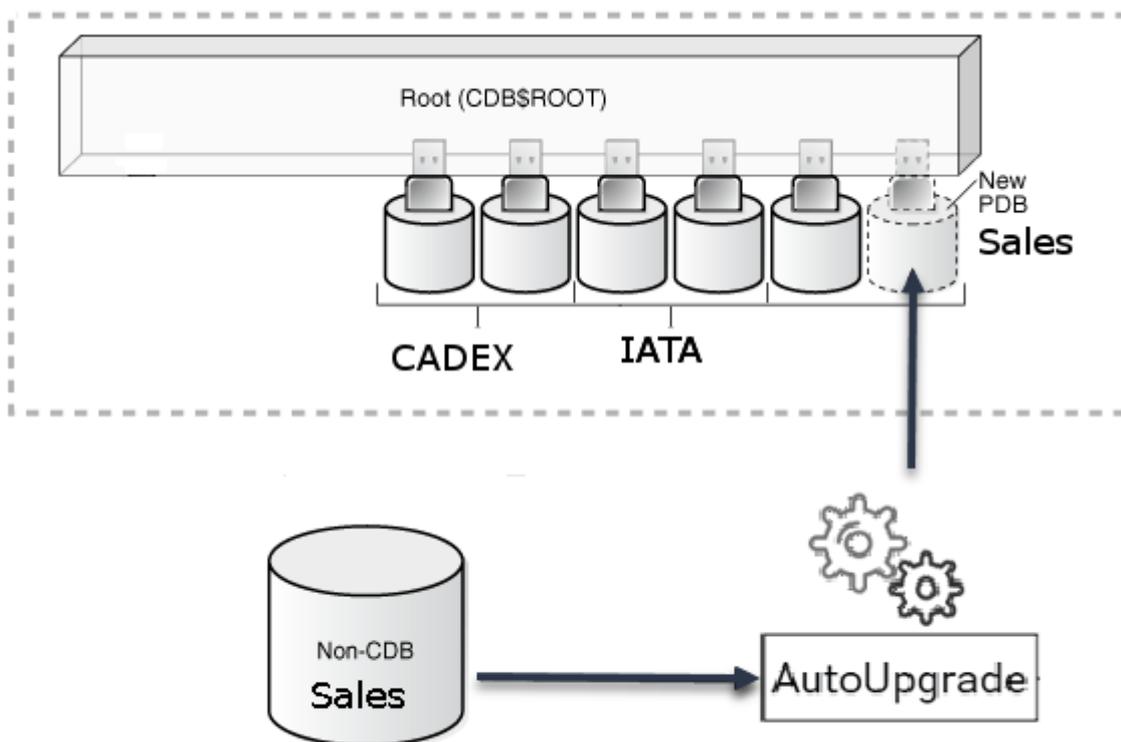
## 注意:



変換およびアップグレードを完了するために AutoUpgrade を実行する前に、ソース・データベースの完全なバックアップを作成し、アップグレードの完全なテストを完了することをお勧めします。AutoUpgrade がこのプロセスを開始した後、非 CDB の Oracle Database 状態にロールバックするオプションはありません。

図2-2 AutoUpgradeを使用した非CDBの変換およびアップグレード

この図では、Salesという非CDB Oracle Databaseが変換され、SalesというPDBにアップグレードされます。



#### ソース非CDBおよびターゲットCDBの要件

PDBへのアップグレードおよび変換を実行するためのソース非CDBおよびターゲットCDBの要件は、次のとおりです。

- AutoUpgradeを使用してアップグレードを実行する前に、ターゲットCDBを作成する必要があります。
- 非CDBから作成されたPDBは、ソース非CDB名を引き続き使用する必要があります。データベースの名前は変更できません。
- ソースとターゲットの両方に同じOracle Databaseオプションのセットが構成されていること。
- ソースCDBとターゲットCDBのエンディアン形式が同じであること。
- ソースCDBとターゲットCDBに、互換性のある文字セットと各国語文字セットがあること。
- ソース非CDBのOracle Databaseリリースおよびオペレーティング・システム・プラットフォームでは、ターゲットCDBリリースへの直接のアップグレードがサポートされている必要があります。
- ソースCDBおよびターゲットCDBに対してオペレーティング・システム認証が有効になっていること

ソース・データベースの最小COMPATIBLEパラメータ設定は12.2.0以上である必要があります。COMPATIBLE設定が低いバージョンの場合、変換およびアップグレード・プロセス中にCOMPATIBLEが12.2.0に設定されます。変換中、元のデータファイルは保持されます。これらは、新しいPDBを作成するためにコピーされません。AutoUpgradeでアップグレードを実行できるようにするには、AutoUpgrade構成ファイルを編集して、AutoUpgradeパラメータtarget\_versionをターゲットCDBリリースに設定し、target\_cdbを使用して、アップグレードしたデータベースが配置されるCDBを識別します。変換およびアップグレード・プロセスでは、AutoUpgradeはその情報を使用してターゲットCDBへのアップグレードを完了します。

#### 例2-5 非CDBからPDBへの変換のためのAutoUpgrade構成ファイル

非CDBからPDBオプションを使用するには、AutoUpgrade構成ファイルでパラメータtarget\_cdbを設定する必要があります。target\_cdbパラメータの値は、非CDBのOracleデータベースを接続するコンテナ・データベースのOracleシステム識別子

(SID)を定義します。次に例を示します。

```
global.autoupg_log_dir=/home/oracle/autoupg
upg1.sid=s12201
upg1.source_home=/u01/product/12.2.0/dbhome_1
upg1.log_dir=/home/oracle/autoupg
upg1.target_home=/u01/product/20.1.0/dbhome_1
upg1.target_base=/u01
upg1.target_version=19.1.0
upg1.target_cdb=cdb19x
```

## 関連項目

- [Oracle Database製品で許可される機能、オプションおよび管理パック](#)

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# 非CDBからPDBへのアップグレードのガイドラインおよび例

変換前に、データファイルおよびデータベースをバックアップし、ソースのOracle Databaseリリースに関するガイドラインに従います。

変換中にデータが失われないようにするために、AutoUpgradeを使用して非CDBのアップグレードおよび変換を実行する前に、アップグレード計画に時間をかけてバックアップ戦略を実装することをお勧めします

## アップグレード計画のガイドライン

非CDBからPDBへの変換とアップグレードのプロセスはリカバリできません。適切なアップグレードと変換を保証し、予期しない停止時間を減らすために、分析フェーズ中に見つかったエラー状態に対処することをお勧めします。

構成ファイルで`target_pdb_copy_option`を使用してデータ・ファイルのコピーを作成すると、既存のデータベースをバックアップとして使用できます。これは安全なオプションですが、追加の時間とディスク領域が必要になります。AutoUpgrade構成ファイルの`target_pdb_copy_option`を設定していない場合、データベース変換は、既存のデータベース・ファイルで使用されているファイルの場所およびファイル名と同じものを使用します。潜在的なデータ損失を防ぐには、データをバックアップし、AutoUpgradeを起動する前にファイル配置計画を検討するようにします。

## 非CDBからマルチテナント・アーキテクチャへのGRPとアップグレード

- アップグレード中に、AutoUpgradeは、AutoUpgradeデプロイ・ワークフローのアップグレード・ステージのコンテキストでのみ使用可能な保証付きリストア・ポイント(GRP)を作成します。潜在的なデータ損失に備えて、AutoUpgradeを起動する前にバックアップ計画を実装する必要があります。
- 非CDBからマルチテナント・アーキテクチャへのデータベース変換は、AutoUpgradeの排出ステージで実行されます。このステージが完了した後、AutoUpgradeによって作成されたGRPは削除され、AutoUpgradeの`restore`コマンドを使用してデータベースをリストアすることはできません。以前の非CDBのOracleデータベース・リリースへのリカバリが必要な場合は、データベースを手動でリカバリする準備が必要です。

## 例2-6 マルチテナント・アーキテクチャを使用した非CDBからOracle Database 19cへのアップグレードおよび変換

デプロイ変換およびアップグレード・ワークフロー中に、AutoUpgradeはGRPを作成し、事前修正ステージを実行します事前修正ステージ完了までのデプロイ・ワークフローのいずれかの部分が失敗した場合、AutoUpgradeはデプロイメントの開始時に作成されたGRPにデータベースをリストアできます。

ただし、事前修正ステージが完了すると、アップグレードされたデータベースはターゲット・リリースのOracle Databaseコンテナ・データベース(CDB)に接続されて変換が完了します。非CDBがCDBに接続されるとすぐに、GRPは有効でなくなり削除されます。

接続中になんらかの問題が発生し、構成ファイルで`target_pdb_copy_option`を使用してデータファイルのコピーを作成しなかった場合、AutoUpgradeはデータベースをリカバリおよびリストアできないことに注意してください。その場合は、データベースを手動でリストアする必要があります。

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# AutoUpgradeの非CDBからPDBの構成ファイルの例

これらの例を使用して、AutoUpgradeの独自のOracle Databaseアップグレード構成ファイルを変更する方法を理解します。

これらの例は、Oracle Database 12cリリース2 (12.2)のDB12という名前の非CDBからターゲットOracle Database 19cのCDB2という名前のCDBにあるOracle Database 19cのPDB3という名前のPDBにアップグレードするケースです。グローバル・パラメータおよびローカル・パラメータの使用の詳細は、パラメータ参照を参照してください。

## 注意:



このアップグレードは非CDBからPDBへの変換であるため、AutoUpgradeでは非CDBを19cにリストアできる保証付きリストア・ポイントを作成できません。問題からリカバリできるようにするには、以前のリリース・データベースをバックアップするか、以前のリリースのOracle DatabaseでCDBをPDBに変換してから、以前のリリースPDBをアップグレードして以降のリリースに変換します。

## 例2-7 ソース・データベースの個別のバックアップ・ソリューションによるアップグレードおよび変換のためのAutoUpgrade構成ファイル

この例では、構成ファイルはAutoUpgradeに、DB12という名前の非CDB Oracle Database 12cをCDB2という名前のOracle Database 19c CDBにあるPDB3という名前のPDBにアップグレードおよび変換するように指示します。

```
global.autoupg_log_dir=/home/oracle/logs
upg1.dbname=DB12
upg1.start_time=NOW
upg1.source_home=/u01/app/oracle/product/12
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=DB12
upg1.log_dir=/home/oracle/logs
upg1.upgrade_node=localhost
upg1.target_version=19
upg1.restoration=no
upg1.target_cdb=CDB2
upg1.target_pdb_name=PDB3
```

## 例2-8 target\_pdb\_copy\_optionを使用するAutoUpgrade

この例では、パラメータupg1.target\_pdb\_copy\_optionを使用して、AutoUpgradeにより、Oracle Database 12c (12.2.0.1)リリースのコピーがOracle Database 19c CDB1に接続したPDB3という名前のPDBに作成されます。

AutoUpgradeは、PDB3を/u02/oradata/CDB1/pdb3から/u02/oradata/CDB2/pdb3に移動します。

```
global.autoupg_log_dir=/home/oracle/logs
upg1.source_home=/u01/app/oracle/product/12.2.0.1
upg1.target_home=/u01/app/oracle/product/19
upg1.sid=CDB1
upg1.pdb=PDB3
upg1.target_cdb=CDB2
upg1.target_pdb_copy_option=file_name_convert=('CDB1', 'CDB2')
upg1.log_dir=/home/oracle/logs
```

**親トピック:** [Oracle Databaseのアップグレードの準備](#)

# 3 AutoUpgradeによる非CDBからPDBへのアップグレードおよび変換

AutoUpgradeユーティリティは、マルチテナント・アーキテクチャを使用して、以前のリリースのOracle Databaseをそれ以降のOracle Databaseのリリースにアップグレードおよび変換するタスクを簡略化できます。

- [同一サーバー上にソースおよびターゲットのデータベース・ホームがあるAutoUpgrade \(通常\)](#)  
Oracle DatabaseのソースおよびターゲットのOracleホームが同じ物理サーバーにインストールされている場合、この例を使用します。
- [異なるサーバー上にソースおよびターゲット・データベース・ホームがあるAutoUpgrade](#)  
Oracle DatabaseのソースおよびターゲットのOracleホームが異なる物理サーバーにあるときは、両方のサーバーでタスクを完了する必要があります。

## 同じサーバー上にソースおよびターゲットのデータベース・ホームがあるAutoUpgrade(通常)

Oracle DatabaseのソースおよびターゲットのOracleホームが同じ物理サーバーにインストールされている場合、この例を使用します。

コンテキスト: ソース・ホームとターゲット・ホームが同じサーバー上にあります。

分析を開始するには、次のコマンドを入力します。

```
java -jar autoupgrade.jar -config config.txt -mode analyze
```

このコマンドは、レポートを生成し、コマンドが検出したエラー状態を示します。エラー状態を確認してください。

アップグレードのデプロイメントを開始するには、次のコマンドを入力します。

```
java -jar autoupgrade.jar -config config.txt -mode deploy
```

**親トピック:** [AutoUpgradeによる非CDBからPDBへのアップグレードおよび変換](#)

## 異なるサーバー上にソースおよびターゲット・データベース・ホームがあるAutoUpgrade

Oracle DatabaseのソースおよびターゲットのOracleホームが異なる物理サーバーにあるときは、両方のサーバーでタスクを完了する必要があります。

コンテキスト: ソースおよびターゲットのOracleホームが異なる物理サーバー上にあります。

分析を開始するには、次のコマンドを入力します。

```
java -jar autoupgrade.jar -config config.txt -mode analyze
```

このコマンドは、レポートを生成し、コマンドが検出したエラー状態を示します。エラー状態を確認してください。

ソースとターゲットのOracle Database Oracleホームは異なるサーバー上にあるため、ソース・サーバーで修正を実行し、ターゲット・サーバーでアップグレードを実行します。

1. ソース・サーバーで修正を実行します。

```
java -jar autoupgrade.jar -config config.txt -mode fixups
```

2. タスクを実行してソースOracle Databaseをソース・サーバーからターゲット・サーバーに移動します。

3. ターゲット・サーバー上で、アップグレード・モードでデータベースを起動し、upgradeモードでAutoUpgradeを実行します。

```
java -jar autoupgrade.jar -config config.txt -mode upgrade
```

**親トピック:** [AutoUpgradeによる非CDBからPDBへのアップグレードおよび変換](#)

## 4 Oracle Databaseのアップグレード後の作業

Oracle Databaseのアップグレードを終了した後に、必要なアップグレード後のタスクを完了し、新しいリリースの推奨事項を検討してください。

- [アップグレード後の状態ツールを使用したアップグレードの確認](#)  
アップグレードのスパール・ログ・ファイルを確認し、アップグレード後の状態ツール(`utlusts.sql`)を使用します。
- [Oracle Databaseのアップグレード後に必要な作業](#)  
アップグレードの完了後、現在の環境に対して指定されたこれらの必要な作業を確認して完了します。
- [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)  
Oracle Databaseを更新する場合、これらの適切なプラクティス・ガイドラインを完了することをお勧めします。特に明記されている場合を除き、これらの方法はすべてのタイプのアップグレードにお勧めします。

# アップグレード後の状態ツールを使用したアップグレードの確認

アップグレードのスプール・ログ・ファイルを確認し、アップグレード後の状態ツール(`utlusts.sql`)を使用します。

アップグレード後の状態ツールは、パス`$ORACLE_HOME/rdbms/admin`にあります。このツールは、Oracle Databaseに付属するSQLスクリプトです。アップグレード後の状態ツールは、新しいリリースの環境で実行します。アップグレード後の状態ツールは、データベースのアップグレード後にいつでも実行できます。

**親トピック:** [Oracle Databaseのアップグレード後の作業](#)

# Oracle Databaseのアップグレード後に必要な作業

アップグレードの完了後、現在の環境に対して指定されたこれらの必要な作業を確認して完了します。

Oracle Databaseをアップグレードした後で、次のアップグレード後タスクを実行する必要があります。これらのタスクは、リプレイ・アップグレード(デフォルト)またはAutoUpgradeで(指定されている場合を除く)でアップグレードを実行する場合に、実行する必要があります。

- [手動アップグレード後のLinuxおよびUNIXシステム上での環境変数の設定](#)  
必要とされるオペレーティング・システムの環境変数が、新しいOracle Databaseリリースのディレクトリを指していることを確認します。
- [PL/SQLパッケージおよび依存プロシージャの確認](#)  
以前のリリースのOracle Databaseにインストールしたパッケージは新しいリリースでは使用できない可能性があります。これはアプリケーションに影響する場合があります。
- [Oracle Databaseのアップグレード後のDBMS\\_STATSパッケージで作成された統計表のアップグレード](#)  
DBMS\_STATS.CREATE\_STAT\_TABLEプロシージャを使用して統計表を作成した場合、DBMS\_STATS.UPGRADE\_STAT\_TABLEを実行してそれらの表をアップグレードします。
- [Oracle XML DBに対するFTPとHTTPのポートおよびHTTP認証の構成](#)  
Oracle Database Configuration Assistant (DBCA)では、Oracle Database 12c以降のリリースのOracle XML DBのポートは構成されません。アップグレードではダイジェスト認証が使用されます。
- [Oracle Databaseのアップグレード後のOracle Textが提供するナレッジ・ベースのインストール](#)  
Oracle Databaseのアップグレード後に、Oracle Textが提供するナレッジ・ベースに対するユーザー拡張をすべて再生成する必要があります。
- [読取り専用OracleホームでのDEMOディレクトリの置換](#)  
読取り専用Oracleホームをアップグレードした後、以前のリリースのOracle Databaseのdemoディレクトリをコピーし、読取り専用Oracleホームのdemoディレクトリを新しいリリースのdemoディレクトリに置き換えます。
- [外部ネットワーク・サービスへのアクセス制御リスト\(ACL\)の構成](#)  
Oracle Database 12c以降のリリースには、UTL\_TCP、UTL\_SMTP、UTL\_MAIL、UTL\_HTTPまたはUTL\_INADDRパッケージに対するファイナグレイン・アクセス制御が含まれています。
- [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)  
ターゲットのデータベース・リリースによっては、Oracle Database Vaultを無効にしないとOracle Databaseのアップグレードを完了できない場合があります。
- [SQLNET.ALLOWED\\_LOGON\\_VERSIONパラメータの動作の確認](#)  
10gより前のリリースのクライアントからのOracle Databaseに対する接続は、ORA-28040: 「一致する認証プロトコルがありません」というエラーによって失敗します。

親トピック: [Oracle Databaseのアップグレード後の作業](#)

## 手動アップグレード後のLinuxおよびUnixシステム上での環境変数の設定

必要とされるオペレーティング・システムの環境変数が、新しいOracle Databaseリリースのディレクトリを指していることを確認します。

通常、オペレーティング・システムの環境変数は、プロファイルおよびシェル・スクリプトで設定されます。次のOracleユーザー環境

変数が新しいOracleホームのディレクトリを指していることを確認します。

- ORACLE\_HOME
- PATH

以前のリリースのOracleホームを参照する他の環境変数(LD\_LIBRARY\_PATHなど)を探します。通常、環境変数内のすべての古いOracleホームを新しいOracleホーム・パスに置き換える必要があります。

#### 関連項目

- [ステップ2: 必要な環境変数が設定されていることの確認](#)

親トピック: [Oracle Databaseのアップグレード後に必要な作業](#)

## PL/SQLパッケージおよび依存プロシージャの確認

以前のリリースのOracle Databaseにインストールしたパッケージは新しいリリースでは使用できない可能性があります。これはアプリケーションに影響する場合があります。

アップグレード後、AutoUpgradeを使用する場合は、無効なオブジェクトに関するAutoUpgradeレポートを確認します。リプレイ・アップグレードを使用する場合は、独自のスクリプトで使用していたパッケージまたは独自のスクリプトからコールしていたパッケージがすべて新しいリリースで使用できることを確認してください。パッケージに依存するプロシージャのテストは、アップグレード計画に含まれる必要があります。

データベース・アプリケーションのコードは、接続先データベースのオブジェクトを参照できます。たとえば、Oracle Call Interface(OCI)およびプリコンパイラ・アプリケーションは無名PL/SQLブロックを発行できます。Oracle Formsアプリケーションのトリガーは、スキーマ・オブジェクトを参照できます。これらのアプリケーションは、参照しているスキーマ・オブジェクトに依存しています。依存性管理の方法は開発環境によって異なります。Oracle Databaseでは、アプリケーションの依存性が自動的に追跡されることはありません。

#### 関連項目

- [Oracle Database管理者ガイド](#)

親トピック: [Oracle Databaseのアップグレード後に必要な作業](#)

## Oracle Databaseのアップグレード後のDBMS\_STATSパッケージで作成された統計表のアップグレード

DBMS\_STATS.CREATE\_STAT\_TABLEプロシージャを使用して統計表を作成した場合、DBMS\_STATS.UPGRADE\_STAT\_TABLEを実行してそれらの表をアップグレードします。

次の例で、greenは統計表の所有者で、STAT\_TABLEは統計表の名前です。

```
EXECUTE DBMS_STATS.UPGRADE_STAT_TABLE(' green', ' stat_table');
```

各統計表にこのプロシージャを実行します。

参照:

DBMS\_STATSパッケージの詳細は、[『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。

**親トピック:** [Oracle Databaseのアップグレード後に必要な作業](#)

## Oracle XML DBに対するFTPとHTTPのポートおよびHTTP認証の構成

Oracle Database Configuration Assistant (DBCA)は、Oracle Database 12c以降のリリースでOracle XML DB用のポートを構成しません。アップグレードではダイジェスト認証が使用されます。

ポートを構成する場合、改善されたセキュリティ機能を利用するために、Oracle XML DB Repositoryへのアクセス用にHTTPの認証も構成することをお勧めします。

Oracle Database 12c以降では、ダイジェスト認証のサポートを提供することによって、データベースのセキュリティが向上しました。ダイジェスト認証は、HTTPプロトコルで一般に使用され、ほとんどのHTTPクライアントでサポートされている業界標準のプロトコルです。ほとんどのHTTPクライアントではこれがサポートされています。ダイジェスト認証により、暗号化(HTTPS)接続が使用されない場合でも、パスワードが常にセキュアな方法で送信されます。ダイジェスト認証をサポートすることによって、組織では、パスワードの漏えいを心配することなくOracle XML DB HTTPを使用するアプリケーションをデプロイできます。Oracle XML DBでのダイジェスト認証のサポートでは、Oracle XML DB HTTPサーバーとMicrosoft Web Folders WebDAVクライアントとの互換性も引き続き維持されます。

新しいリリースのインストールまたはアップグレード後に、次のようにOracle XML DBのFTPおよびHTTPポートを手動で構成する必要があります。

1. DBMS\_XDB\_CONFIG.setHTTPPort(HTTP\_port\_number)を使用して、Oracle XML DBのHTTPポートを設定します。

```
SQL> exec DBMS_XDB_CONFIG.setHTTPPort(port_number);
```

2. DBMS\_XDB\_CONFIG.setFTPPort(FTP\_port\_number)を使用して、Oracle XML DBのFTPポートを設定します。

```
SQL> exec DBMS_XDB_CONFIG.setFTPPort(FTP_port_number);
```



ノート:

手順内の FTP および HTTP で使用するポート番号は、DBMS\_XDB\_CONFIG.getFTPPort および DBMS\_XDB\_CONFIG.getHTTPPort をそれぞれ使用することによって問い合わせることができます。

3. 使用されているすべてのポート番号を確認するには、DBMS\_XDB\_CONFIG.usedportを問い合わせます。

**親トピック:** [Oracle Databaseのアップグレード後に必要な作業](#)

## Oracle Databaseのアップグレード後のOracle Textが提供するナレッジ・ベースのインストール

Oracle Databaseのアップグレード後に、Oracle Textが提供するナレッジ・ベースに対するユーザー拡張をすべて再生成す

る必要があります。

ユーザー拡張の再生成は、指定されたOracleホームにインストールされているすべてのデータベースに影響します。

アップグレード後、新しいOracle Databaseの付属製品の一部であるOracle Textナレッジ・ベースは、すぐに使用できるようにはなっていません。アップグレード前には使用可能であったナレッジ・ベースに依存するOracle Textの機能は、アップグレード後には機能しなくなります。これらの機能を再度使用可能にするには、Oracle Textが提供するナレッジ・ベースを新しいOracle Databaseリリースのインストール・メディアからインストールする必要があります。

参照:

- Oracle Textナレッジ・ベースの詳細は、[Oracle Textアプリケーション開発者ガイド](#)を参照してください。
- 付属製品については、[Oracle Databaseインストール・ガイド](#)を参照してください

親トピック: [Oracle Databaseのアップグレード後に必要な作業](#)

## 読取り専用OracleホームでのDEMOディレクトリの置換

読取り専用Oracleホームをアップグレードした後、以前のリリースのOracle Databaseのdemoディレクトリをコピーし、読取り専用Oracleホームのdemoディレクトリを新しいリリースのdemoディレクトリに置き換えます。

Oracle Database 18c以上のリリースでは、ファイル・パス`oracle_home/rdbms/demo`に製品デモンストレーション・ディレクトリが含まれています。これらのディレクトリには、各Oracle Databaseリリースのオプションと機能に固有の例と製品デモンストレーションが含まれています。これらの一部は、Oracle Database Examplesをインストールすると、アップグレード後に追加できます。以前のリリースでは、以前のリリースのデモンストレーション・ファイルを使用してダウンロードして作業した場合、2つの問題があります。新しいリリースでレビューおよびテストするために以前のリリースの作業を保存する必要があること、および新しいリリースに固有のデモンストレーションのリフレッシュを取得する必要があることです。

Oracleホームをアップグレードした後、新しいデモンストレーション・ファイルで実行する他の作業をダウンロードして実行したら、古いデモンストレーション・ファイルをリフレッシュできます。

例4-1 以前のリリースのDemoディレクトリのコピーおよび読取り専用Oracleホームでのデモンストレーションのリフレッシュ

アップグレード後に、この手順を使用して、以前のdemoディレクトリの作業を読取り専用Oracleホームに保存し、以前のリリースのdemoディレクトリを新しいリリースのdemoディレクトリで置き換えます。

1. Oracleソフトウェア所有者ユーザー(`oracle`)としてログインします。
2. `rdbms/demo`ディレクトリが読取り専用Oracleホームにコピーされているかどうかを確認します。

この例では、環境変数`ORACLE_BASE_HOME`が読取り専用Oracleホームのパスとして定義されています。

LinuxおよびUNIXプラットフォーム:

```
$ ls -l -d $ORACLE_BASE_HOME/rdbms/demo  
/u01/app/oracle/product/19.0.0/dbhome_1/rdbms/demo
```

Microsoft Windowsプラットフォーム

```
ls -l -d %ORACLE_BASE_HOME%\rdbms\demo
%ORACLE_BASE_HOME%\rdbms\demo
```

3. ディレクトリを読み取り専用Oracleホームに変更し、コピーを作成します。ここで、demo.old\_release18は以前のリリースのデモンストレーション・ファイルに指定する名前です。

```
cd $ORACLE_BASE_HOME/rdbms
mv demo demo.old_release18
```

4. 新しいdemoディレクトリをアップグレードされたOracleホームから読み取り専用Oracleホームにコピーします。  
この例では、環境変数ORACLE\_HOMEが新しいリリースのOracleホームとして定義されています。

LinuxおよびUNIX:

```
cp -r $ORACLE_HOME/rdbms/demo demo
```

Microsoft Windows

```
xcopy c:%ORACLE_HOME%\rdbms\demo c:%ORACLE_BASE_HOME%\rdbms\demo /E
```

**親トピック:** [Oracle Databaseのアップグレード後に必要な作業](#)

## 外部ネットワーク・サービスへのアクセス制御リスト(ACL)の構成

Oracle Database 12c以降のリリースには、UTL\_TCP、UTL\_SMTP、UTL\_MAIL、UTL\_HTTPまたはUTL\_INADDRパッケージに対するファイナグレイン・アクセス制御が含まれています。

これらのパッケージを使用するアプリケーションがある場合、影響を受けるパッケージを以前のリリースの場合と同様に動作させるには、Oracle Databaseのアップグレード後にデータベースのネットワーク・アクセス制御リスト(ACL)を構成する必要があります。ACLがない場合、エラー「ORA-24247: アクセス制御リスト(ACL)によりネットワーク・アクセスが拒否されました」でアプリケーションが失敗する可能性があります。

参照:

一部のユーザーはホストAに接続し、別のユーザーはホストBに接続するなど、より複雑な状況については、[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:** [Oracle Databaseのアップグレード後に必要な作業](#)

## Oracle Databaseのアップグレード後のOracle Database Vaultの有効化

ターゲットのデータベース・リリースによっては、Oracle Database Vaultを無効にしないとOracle Databaseのアップグレードを完了できない場合があります。

- [Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード](#)  
ターゲットのOracle Databaseリリースが12.2以上である場合は、Oracle Database Vaultを無効にせずにアップ

グレードできます。

- [Oracle Database Vaultが関連する一般的なアップグレード・シナリオ](#)

アップグレード後にOracle Database Vaultを有効にする必要があるかどうかは、ソースのOracle Databaseリリースによって異なります。

**親トピック:** [Oracle Databaseのアップグレード後に必要な作業](#)

## Oracle Database Vaultの無効化なしでのOracle Databaseのアップグレード

ターゲットのOracle Databaseリリースが12.2以上である場合は、Oracle Database Vaultを無効にせずにアップグレードできます。

ソースのOracle DatabaseリリースでOracle Database Vaultを有効にしていた場合は、Oracle Database Vaultを事前に無効にしなくても、Oracle DatabaseをOracle Database 18c以降のリリースにアップグレードできます。ソースのOracle DatabaseリリースがOracle Database 12cリリース1 (12.1)以上である場合は、アップグレード後に、アップグレード前に用意したものと同一強化設定を使用してOracle Database Vaultが有効になります。たとえば、ソース・データベースがOracle Databaseリリース12.1であり、そのリリースでOracle Database Vaultを無効にしていた場合は、アップグレード後も無効になります。アップグレード前にソースのOracle Databaseリリース12.1データベースでOracle Database Vaultを有効にしていた場合は、アップグレード後もOracle Database Vaultは有効になります。

アップグレード前に手動でOracle Database Vaultを無効にした場合は、アップグレード後に手動でOracle Database Vaultを有効にする必要があります。

アップグレード前にOracle Database Vaultを有効にしていなかった場合は、アップグレード後に手動で有効にできます。

プロシージャ`dvsys.dbms_macadm.enable_dv()`を使用して、アップグレードしたデータベースでOracle Database Vaultを有効にします。このプロシージャは、`DV_OWNER`を付与されたユーザー・アカウントを使用して実行します。プロシージャを実行したら、そのプロシージャを有効にするためにデータベース・インスタンスを再起動します。

### 関連項目

- [Oracle Database Vaultの管理者ガイド](#)

**親トピック:** [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)

## Oracle Database Vaultが関連する一般的なアップグレード・シナリオ

アップグレード後にOracle Database Vaultを有効にする必要があるかどうかは、ソースのOracle Databaseリリースによって異なります。

- Oracle Database 11gリリース2 (11.2)以下からのアップグレード: アップグレード後、Oracle Database Vaultはデフォルトで無効になります。
- Oracle Database 12cリリース1 (12.1)以上からのアップグレード: アップグレード後、Oracle Database Vaultはアップグレード前に用意したものと同一強化ステータスになります。

表4-1 Oracle Database Vaultの一般的なアップグレード・シナリオおよびアップグレード準備作業

ソース・データベース・リリース	ターゲット・データベース・リリース	アップグレード前にDatabase Vaultを無効にする必要があるかどうか	アップグレード後のDatabase Vaultのステータス
11.2 以前	12.1	必要	無効。アップグレード後に Database Vault を手動で有効にする必要があります。
11.2 以前	12.2、18.1 以降	不要	無効。アップグレード後に Database Vault を手動で有効にする必要があります。
12.1、12.2、18.1 以降	12.2、18.1 以降	不要	Database Vault はアップグレード前に用意したものと同一強化ステータスになります。

**親トピック:** [Oracle Databaseのアップグレード後のOracle Database Vaultの有効化](#)

## SQLNET.ALLOWED\_LOGON\_VERSIONパラメータの動作の確認

10gより前のリリースのクライアントからのOracle Databaseに対する接続は、ORA-28040: 「一致する認証プロトコルがありません」というエラーによって失敗します。

Oracle Database 18c以降では、SQLNET.ALLOWED\_LOGON\_VERSIONパラメータのデフォルト値が、11 (Oracle Database 12c (12.2))から12 (Oracle Database 18c以降のリリース)に変更されています。このパラメータの使用は非推奨になりました。

SQLNET.ALLOWED\_LOGON\_VERSIONは、現在、SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERおよびSQLNET.ALLOWED\_LOGON\_VERSION\_CLIENTパラメータに置き換えられています。アップグレードしたデータベースでSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータを明示的に設定していない場合、10gより前のリリースのクライアントからの接続は、ORA-28040: 「一致する認証プロトコルがありません」というエラーによって失敗します。セキュリティ強化のため、データベース・ユーザーのパスワード検証機能を確認し、SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERおよびSQLNET.ALLOWED\_LOGON\_VERSION\_CLIENTパラメータを設定して正しいパスワード検証機能を使用するようにデータベースを構成します。

既存のデータベースにパスワードで保護されたロール(セキュア・ロール)があり、デフォルトのSQLNET.ALLOWED\_LOGON\_VERSION\_SERVER設定である12を使用してOracle Database 18c以降のリリースにアップグレードする場合、そのセキュア・ロールに含まれるのはリリース10gの検証機能のみであるため、アップグレード後もセキュア・ロールが使用可能な状態になるように、管理者は各セキュア・ロールのパスワードをリセットする必要があります。

参照:

- パスワードのセキュリティの脅威から守る方法の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。
- ユーザーのパスワード・バージョンの設定の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)[『Oracle Databaseセ](#)

[セキュリティ・ガイド](#)』を参照してください。

**親トピック:** [Oracle Databaseのアップグレード後に必要な作業](#)

# Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス

Oracle Databaseを更新する場合、これらの適切なプラクティス・ガイドラインを完了することをお勧めします。特に明記されている場合を除き、これらの方法はすべてのタイプのアップグレードにお勧めします。

- [Databaseのバックアップ](#)  
少なくともレベル1のバックアップを実行するか、時間が許せばレベル0のバックアップを実行することをお勧めします。
- [AutoUpgradeのアップグレード後のチェックの実行](#)  
AutoUpgradeをdeployモードで実行しなかった場合は、preupgradeパラメータを指定してAutoUpgradeをpostfixupsモードで実行します。
- [DBMS\\_STATSを使用した固定オブジェクトの統計の再収集](#)  
アップグレードの後に、または他のデータベース構成の変更後に、Oracle Databaseに対して代表的なワークロードを実行してから固定オブジェクトの統計を再収集することをお勧めします。
- [パスワードのリセットによる大/小文字区別の強制](#)  
アップグレードしたデータベースで、デフォルト・ユーザー・アカウントおよびユーザー・アカウントの大/小文字区別のあるパスワードを使用してセキュリティを強化します。
- [10Gパスワード・バージョンを使用するユーザーのパスワードの確認と再設定](#)  
セキュリティを向上するため、10Gバージョンのパスワードを使用しているユーザー・アカウントを確認しパスワードをリセットして、より安全なバージョンのパスワードが今後使用されるようにします。
- [Oracle Grid Infrastructure、Oracle ASMおよびOracle Clusterwareの理解](#)  
Oracle ClusterwareとOracle Automatic Storage Management (Oracle ASM)は、両方ともOracle Grid Infrastructureインストールの一部です。
- [Oracle Grid InfrastructureのインストールとアップグレードおよびOracle ASM](#)  
Oracle ASMは、Oracle Grid Infrastructureとともにインストールされます。
- [新機能の適宜追加](#)  
データベース・アップグレード計画の一環として新機能を確認します。
- [必要な新しい管理手順の作成](#)  
スクリプトおよびプロシージャの再確認を計画し、必要に応じて変更します。
- [ロールバック・セグメントから自動UNDOモードへの移行](#)  
データベース・リリースがOracle Database 11gより前の場合は、ロールバック・セグメント(手動UNDO管理)を使用してアップグレードしているデータベースを自動UNDO管理に移行する必要があります。
- [LONGデータ型からLOBデータ型への表の移行](#)  
ALTER TABLE文を使用して、LONGデータ型の列をCLOBに、LONG RAWデータ型の列をBLOBに変更できます。
- [アップグレードしたOracle Databaseの統合監査の使用への移行](#)  
統合監査の全機能を使用するには、手動で統合監査に移行する必要があります。
- [再構築対象のOracle Text索引の識別](#)  
新しいOracle Databaseリリースへのアップグレード後に、再構築が推奨されるトークン表付きOracle Text索引の識別に役立つスクリプトを実行できます。
- [DBMS\\_SCHEDULERジョブの削除および再作成](#)  
以前のリリースからのアップグレード後にDBMS\_SCHEDULERジョブが機能しなくなった場合は、ジョブを削除して再作成します。

- [アップグレード後の統合監査レコードの転送](#)  
アップグレードし、統合監査に移行した後で高いパフォーマンスを得る方法を理解するためには、これらのトピックを確認します
- [Oracle Databaseのアップグレード後のリカバリ・カタログ・アップグレードについて](#)  
RMANクライアントで要求されるバージョンより古いリカバリ・カタログ・スキーマを使用している場合、それをアップグレードする必要があります。
- [アップグレードされたデータベースでの無効なリリース更新バグ修正の有効化](#)  
実行計画の変更の原因となる可能性があるリリース更新のバグ修正は無効になっているため、使用する無効なバグ修正を有効にすることをお勧めします。
- [アップグレードした本番Oracle Databaseのテストについて](#)  
アプリケーションが期待どおり動作することを確認するため、テスト・データベースに実行したテストを本番データベースに対して繰り返します。
- [Oracle Databaseのアップグレード後のタイムゾーン・ファイルのバージョンのアップグレード](#)  
データベースのアップグレードを完了した後、upgrade.xmlのAutoUpgradeアップグレード前レポートによってタイムゾーン・ファイルをアップグレードするよう指示され、AutoUpgradeを設定してこのタスクを完了しない場合は、DBMS\_DST PL/SQLパッケージを使用してタイムゾーン・ファイルをアップグレードします。

親トピック: [Oracle Databaseのアップグレード後の作業](#)

## データベースのバックアップ

少なくともレベル1のバックアップを実行するか、時間が許せばレベル0のバックアップを実行することをお勧めします。

### 関連項目

- [データベースのバックアップ](#)

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## AutoUpgradeアップグレード後チェックの実行

AutoUpgradeをdeployモードで実行しなかった場合は、preupgradeパラメータを指定してAutoUpgradeをpostfixupsモードで実行します。

ノート:



AutoUpgrade を deploy モードで実行した場合、このステップはすでに完了しているため、ここで完了する必要はありません。

アップグレード後にデータベースを確認する方法を確認するには、次の例を使用します。

### 例4-2 アップグレード後修正モードを使用したAutoUpgradeの実行

Oracle Database 12cリリース2 (12.2)のOracle Databaseをアップグレードしたとします。アップグレード後チェックを実行するには、次の手順を実行します。

1. Oracleホーム環境をソースOracle Databaseホームに設定します。

```
setenv ORACLE_HOME /u01/app/oracle/product/12.2.0/dbhome_1
```

2. Oracleシステム識別子(SID)をソースOracle Database SIDに設定します。

```
setenv ORACLE_SID db122
```

3. postfixupsモードでpreupgradeパラメータを使用してAutoUpgradeを実行し、ターゲット・ホームをターゲットのOracle Database Oracleホームに設定します。次に例を示します。

```
java -jar autoupgrade.jar -preupgrade  
"target_home=/u01/app/oracle/product/19.0.0/dbhome_1, dir=/autoupgrade/test/log" -mode  
postfixups
```

4. ディレクトリ/autoupgrade/test/log/db122/102/postfixupsの下にあるファイルpostfixups.xmlで、修正後スクリプト・チェックの結果を確認します。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## DBMS\_STATSを使用した固定オブジェクトの統計の再収集

アップグレードの後に、または他のデータベース構成の変更後に、Oracle Databaseに対して代表的なワークロードを実行してから固定オブジェクトの統計を再収集することを強くお勧めします。

ノート:



パフォーマンス・チューニング用に非常に正確な固定オブジェクト統計を提供するには、システムが代表的なワークロードで実行されている時点でベースライン統計を収集することをお勧めします。有用な結果を得るには、アップグレードの直後に DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATS を実行しないでください。

固定オブジェクトは、X\$表とその索引です。V\$パフォーマンス・ビューは、X\$表を通じて定義されます。固定オブジェクトの統計の収集は、データベース・パフォーマンスにとって有益です。これは、最適化が適切な実行計画を生成する際に役立つため、データベース・パフォーマンスが向上する可能性があります。代表的な統計を取得しないと、実行計画が最適ではなくなる可能性があり、深刻なパフォーマンス問題が発生する場合があります。

データベースで代表的なワークロードが実行されていることを確認し、DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATS PL/SQL プロシージャを使用して固定オブジェクト統計を収集します。DBMS\_STATS.GATHER\_FIXED\_OBJECTS\_STATSでは、INIT.ORAまたはSPFILEからすべての非表示またはアンダースコアのパラメータおよびイベントを削除するための推奨事項も表示されます。

X\$表は一時的な性質を持つため、システムで代表的なワークロードが発生したときに固定オブジェクトの統計を収集する必要があります。ピーク負荷時に固定オブジェクトの統計を収集できない場合、システムが実行時の状態で、最も重要なタイプの固定オブジェクト表が移入された後に収集することをお勧めします。

固定オブジェクトの統計を収集するには、次のPL/SQLプロシージャを実行します。

```
SQL> execute dbms_stats.gather_fixed_objects_stats;
```

## 関連項目

- [データベース統計の収集](#)

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## パスワードのリセットによる大/小文字区別の強制

アップグレードしたデータベースで、デフォルト・ユーザー・アカウントおよびユーザー・アカウントの大/小文字区別のあるパスワードを使用してセキュリティを強化します。

セキュリティを強化するために、パスワードで大/小文字の区別を有効にすることをお勧めします。大/小文字を区別すると、ユーザーは、正しいパスワード文字列を入力し、さらにその文字列の各文字の大/小文字も正しく区別する必要があるため、パスワードのセキュリティを強化できます。たとえば、パスワードがhPP5620qrの場合、hpp5620QRまたはhPp5620Qrと入力すると失敗します。

データベースを保護するため、パスワードはセキュアな方法で作成します。データベースのデフォルト・パスワードを使用している場合、それらのパスワードは変更してください。デフォルトでは、パスワードを変更すると、大/小文字区別が強制されます。事前定義されたユーザー・アカウントのパスワードを含め、すべてのパスワードは、Oracle推奨のパスワード要件を満たしている必要があります。

アップグレード後に新しく作成されたデータベースでは、追加の作業や追加の管理要件はありません。

パスワード変更に関する既存のデータベース要件およびガイドライン

- Oracle Database 12cリリース1 (12.1)以上のデフォルトのセキュリティ設定が適用されている場合、パスワードは8文字以上にする必要があり、welcomeやoracleなどのパスワードは使用できません。
- IGNORECASEパラメータは非推奨です。このパラメータは使用しないでください。
- 既存のデータベースでパスワードの大/小文字区別を利用するには、データベースのアップグレード作業中に既存のユーザーのパスワードをリセットする必要があります。既存のデータベース・ユーザーごとに、ALTER USER文を使用してパスワードをリセットします。
- DBA\_USERSのPASSWORD\_VERSIONS列を問い合わせ、10Gのパスワード・バージョンのみを持ち、11Gや12Gのパスワード・バージョンを持たないアカウントのUSERNAMEを見つけます。10Gのパスワード・バージョンのみを持つすべてのアカウントのパスワードをリセットします。

参照:

- パスワードの大/小文字区別の詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください
- パスワードの強度の詳細は、[『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: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## 10Gパスワード・バージョンを使用するユーザーのパスワードの確認と再設定

セキュリティを向上するため、10Gバージョンのパスワードを使用しているユーザー・アカウントを確認しパスワードをリセットして、より

安全なバージョンのパスワードが今後使用されるようにします。

現行ユーザーのすべてのパスワード・バージョンの確認

DBA\_USERSデータ・ディクショナリ・ビューに問い合せて、ユーザー・アカウント用に構成されたパスワード・バージョンすべてのリストを出すことができます。

次に例を示します。

```
SELECT USERNAME, PASSWORD_VERSIONS FROM DBA_USERS;
USERNAME                                PASSWORD_VERSIONS
-----                                -
JONES                                    10G 11G 12C
ADAMS                                    10G 11G
CLARK                                    10G 11G
PRESTON                                  11G
BLAKE                                    10G
```

PASSWORD\_VERSIONS列は、アカウントに存在するパスワード・バージョンのリストを示しています。10Gは、以前の、大/小文字を区別しないOracleパスワード・バージョン、11GはSHA-1ベースのパスワード・バージョン、12CはSHA-2ベースのSHA-512パスワード・バージョンを表します。

- ユーザー-jones: このユーザーのパスワードは、SQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータ設定が8のときに、Oracle Database 12cリリース12.1で再設定されました。これにより、3つのすべてのパスワード・バージョンを作成できます。
- ユーザー-adamsおよびclark: これらのアカウントのパスワードは当初Oracle Database 10gで作成され、Oracle Database 11gで再設定されました。Oracle Database 11gソフトウェアは、その時点でデフォルトのSQLNET.ALLOWED\_LOGON\_VERSION設定の8を使用していました。大/小文字を区別しないことがデフォルトで有効になっているため、これらのパスワードは、prestonのパスワードと同様に大/小文字が区別されるようになりました。
- ユーザー-preston: このアカウントは排他モード(SQLNET.ALLOWED\_LOGON\_VERSION = 12)で実行されていたOracle Database 11gデータベースからインポートされたものです。
- ユーザー-blake: このアカウントはまだOracle Database 10gパスワード・バージョンを使用しています。この段階では、ユーザー-blakeはログインできません。

#### 10Gパスワード・バージョンを使用するユーザーの再設定

すべてのユーザーのアカウントから10Gパスワード・バージョンを削除する必要があります。次の手順で、10Gパスワード・バージョンを持つユーザーのパスワードを再設定するには、ログインが許可される前のクライアントの機能レベル要件を制御するSQLNET.ALLOWED\_LOGON\_VERSION\_SERVER設定を、一時的に緩和する必要があります。設定を緩和することで、それらのユーザーがログインしてパスワードを変更し、10Gパスワード・バージョンに加えてより新しいパスワード・バージョンを生成できるようになります。後で、排他モードを使用するようにデータベースを設定し、クライアントが05L\_NP機能を使用できることを確認できます。そうすれば、パスワードにもう10Gが含まれず、より安全な11Gおよび12Cパスワード・バージョンだけを持つように、ユーザーはもう一度パスワードを再設定できます。

1. DBA\_USERSビューに問い合せて、10Gパスワード・バージョンだけを使用するユーザーを見つけます。

```
SELECT USERNAME FROM DBA_USERS
WHERE ( PASSWORD_VERSIONS = ' 10G ' )
```

```
OR PASSWORD_VERSIONS = '10G HTTP ' )  
AND USERNAME <> ' ANONYMOUS' ;
```

2. 次のようにして、データベースが排他モードで実行されないように構成します。

- a. sqlnet.oraファイルのSQLNET.ALLOWED\_LOGON\_VERSION\_SERVER設定を、デフォルトよりも許容度が高くなるように編集します。例:

```
SQLNET.ALLOWED_LOGON_VERSION_SERVER=11
```

- b. CDBルートにいる場合は、データベースを再起動します(たとえば、SHUTDOWN IMMEDIATEに続いてSTARTUP)。PDBにいる場合は、SYSDBA管理権限を使用してrootに接続し、次の文を入力します。

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

3. DBA\_USERSビューに問い合わせて、10Gパスワード・バージョンのみを使用するユーザーを期限切れにします。

11Gまたは12C、あるいはその両方のパスワード・バージョンを持つユーザーではなく、10Gパスワード・バージョンだけを持つユーザーを期限切れにする必要があります。

次に例を示します。

```
ALTER USER username PASSWORD EXPIRE;
```

4. 期限切れにしたユーザーにログインするよう、依頼します。

ユーザーがログインすると、パスワードを変更するよう求められます。データベースは10Gパスワード・バージョンに加えて、欠落している11Gおよび12Cパスワード・バージョンを生成します。データベースは許容モードで実行中のため、10Gパスワード・バージョンは引き続き存在します。

5. ユーザーが接続しているクライアント・ソフトウェアに05L\_NP機能があることを確認します。

Oracle Databaseリリース11.2.0.3以上のすべてのクライアントに05L\_NP機能があります。以前のOracle Databaseクライアントがある場合は、CPU0ct2012パッチをインストールする必要があります。

6. すべてのクライアントに05L\_NP機能が導入されたら、次のようにしてサーバーのセキュリティを元の排他モードに設定します。

- a. サーバーのsqlnet.oraファイルからSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERパラメータを削除するか、サーバーのsqlnet.oraファイルでSQLNET.ALLOWED\_LOGON\_VERSION\_SERVERの値を12に設定しなおして、排他モードに設定します。

```
SQLNET.ALLOWED_LOGON_VERSION_SERVER = 12
```

- b. CDBルートにいる場合は、データベースを再起動します(たとえば、SHUTDOWN IMMEDIATEに続いてSTARTUP)。PDBにいる場合は、SYSDBA管理権限を使用してrootに接続し、次の文を入力します。

```
ALTER PLUGGABLE DATABASE pdb_name CLOSE IMMEDIATE;  
ALTER PLUGGABLE DATABASE pdb_name OPEN;
```

7. まだ10Gパスワード・バージョンを持っているアカウントを見つけます。

```
SELECT USERNAME FROM DBA_USERS  
WHERE PASSWORD_VERSIONS LIKE '%10G%'  
AND USERNAME <> ' ANONYMOUS' ;
```

8. まだ10Gパスワード・バージョンを持っているアカウントを期限切れにします。

```
ALTER USER username PASSWORD EXPIRE;
```

9. これらのユーザーにアカウントにログインするよう、依頼します。

ユーザーはログインするときに、パスワードを再設定するようプロンプトが表示されます。そうすると、データベースはそのアカウントに11Gおよび12Cパスワード・バージョンだけを生成します。データベースは排他モードで実行されているため、10Gパスワード・バージョンはもう生成されません。

10. 次の問合せを再実行します。

```
SELECT USERNAME FROM DBA_USERS  
WHERE PASSWORD_VERSIONS LIKE '%10G%'  
AND USERNAME <> 'ANONYMOUS';
```

この問合せで返された結果が0件だった場合は、10Gパスワード・バージョンを持つユーザー・アカウントがないことを意味します。したがって、データベースは以前のリリースよりもよりセキュアなモードで実行されています。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## Oracle Grid Infrastructure、Oracle ASMおよびOracle Clusterwareの理解

Oracle ClusterwareとOracle Automatic Storage Management (Oracle ASM)は、両方ともOracle Grid Infrastructureインストールの一部です。

Oracle Grid Infrastructureは、単一サーバーにインストールした場合、Oracle ASMとともにOracle Restartインストールとしてデプロイされます。クラスタにインストールした場合は、Oracle ASMとともにOracle Clusterwareインストールとしてデプロイされます。

Oracle Restartは、単一インスタンス環境におけるOracle Databaseの可用性を向上します。Oracle Restartをインストールし、データベース、リスナー、Oracle ASMインスタンスなどのOracle Databaseソフトウェア・スタックの一部が一時的に失敗した場合、失敗したコンポーネントはOracle Restartによって自動的に再起動されます。また、これらのすべてのコンポーネントは、データベース・ホスト・コンピュータが再起動されるときに、Oracle Restartによって起動されます。コンポーネントは、コンポーネント間の依存性を考慮して適切な順序で起動されます。

Oracle Clusterwareは、単一サーバーを単一システムとして連携するようにクラスタ化できるポータブル・クラスタ・ソフトウェアです。Oracle RACに必須なインフラストラクチャも提供します。また、Oracle Clusterwareでは、クラスタ内の任意のOracleアプリケーションまたはその他のアプリケーションを保護できます。いずれの場合でも、Oracle Clusterwareはそれらのシステムにおけるインテリジェント機能としてクラスタ・ノード間の必要な連携を確実にします。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## Oracle Grid InfrastructureのインストールとアップグレードおよびOracle ASM

Oracle ASMは、Oracle Grid Infrastructureとともにインストールされます。

以前のリリースまでは、Oracle ASMは、Oracle Databaseインストールの一部としてインストールされていました。Oracle Databaseリリース11.2以上では、Oracle ASMは、Grid Infrastructureコンポーネントのインストール時にインストールされます。Oracle ASMは、Oracle ClusterwareとOracleホームを共有します。

参照:

Oracleホーム、ロール割当て済のシステム権限グループ、様々なインストール・ソフトウェア所有者ユーザー、および他の変更の詳細は、使用しているプラットフォームの『[Oracle Grid Infrastructureインストール・ガイド](#)』を参照してください。

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## 新機能の適宜追加

データベース・アップグレード計画の一環として新機能を確認してください。

『Oracle Database新機能ガイド』では、新しいOracle Databaseリリースで使用可能な多くの新機能について説明されています。これらのどの新機能がデータベースおよびアプリケーションに有効かを判断してください。その後、これらの機能を使用する計画を立てることができます。

新しいOracle Databaseソフトウェアを使用するためにすぐに変更する必要はありません。データベースおよびアプリケーションに、新機能の拡張機能を徐々に取り入れることもできます。

参照:

『[Oracle Database新機能ガイド](#)』

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## 必要な新しい管理手順の作成

スクリプトおよびプロシージャの再確認を計画し、必要に応じて変更します。

新しいOracle Databaseリリースの機能をよく理解したうえで、データベース管理用のスクリプトおよびプロシージャを再確認し、変更が必要かどうかを判断します。

それぞれのアプリケーションに必要な変更を、データベースにも行う必要があります。たとえば、データベースで整合性制約を使用可能にした場合、アプリケーションでのデータ・チェックの一部を削除できます。

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## ロールバック・セグメントから自動UNDOモードへの移行

データベース・リリースがOracle Database 11gより前の場合は、ロールバック・セグメント(手動UNDO管理)を使用してアップグレードしたデータベースを、自動UNDO管理に移行する必要があります。

自動UNDO管理は、デフォルトのUNDO領域管理モードです。システムで使用するUNDO領域管理モードは、UNDO\_MANAGEMENT初期化パラメータで指定します。

- UNDO\_MANAGEMENTがAUTOに設定されている場合(またはUNDO\_MANAGEMENTが設定されていない場合)、データベース・インスタンスは自動UNDO管理モードで起動します。

UNDO\_MANAGEMENT初期化パラメータがNULLの場合のデフォルトは、Oracle Database 11gリリース1 (11.1)以降では自動UNDO管理モードです。以前のリリースでは、手動UNDO管理モードがデフォルトです。以前のリリースをアップグレードするときには注意してください。

- UNDO\_MANAGEMENTが MANUALに設定されている場合、UNDO領域はロールバック・セグメントとして外部に割り当てられます。

1. UNDO\_MANAGEMENTパラメータを UNDO\_MANAGEMENT=MANUALに設定します。
2. インスタンスを再起動して標準的なビジネス・サイクルを一通り実行し、代表的なワークロードを取得します。ワークロードを評価し、自動UNDO管理に必要なUNDO表領域のサイズを計算します。
3. 標準的なビジネス・サイクルを完了したら、次のファンクションを実行してUNDO表領域のサイズを収集すると、UNDO表領域のサイズ変更に関与します。このファンクションを実行するにはSYSDBA権限が必要です。

```
DECLARE
  utbsiz_in_MB NUMBER;
BEGIN
  utbsiz_in_MB := DBMS_UNDO_ADV.RBU_MIGRATION;
end;
/
```

このファンクションではPL/SQLプロシージャが実行され、システム構成およびシステムのロールバック・セグメントの使用状況を基にして、新しいUNDO表領域のサイズを求める方法に関する情報が提供されます。このファンクションはサイズ変更に関する情報を直接戻します。

4. 必要なサイズのUNDO表領域を作成し、UNDO\_MANAGEMENT=AUTOに設定するかパラメータを削除して、自動UNDO管理を有効にします。
5. Oracle RAC構成の場合は、すべてのインスタンスでこれらのステップを繰り返します。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## LONGデータ型からLOBデータ型への表の移行

ALTER TABLE文を使用して、LONGデータ型の列をCLOBに、LONG RAWデータ型の列をBLOBに変更できます。

LOBデータ型(BFILE、BLOB、CLOBおよびNCLOB)には、LONGデータ型よりも多くのメリットがあります。

次の例では、long\_tab表のlong\_colというLONG列が、CLOBデータ型に変更されます。

```
SQL> ALTER TABLE Long_tab MODIFY ( long_col CLOB );
```

この方法でLONG列をLOBに変換した後も、表に設定されている既存の制約およびトリガーはすべて使用できます。ただし、表のすべての列で、ドメイン索引およびファンクション索引を含むすべての索引が使用不可となるため、ALTER INDEX ... REBUILD文を使用してすべての索引を再構築する必要があります。また、LONG列上のドメイン索引は、LONG列をLOBに変更する前に削除する必要があります。

参照:

LOBデータを使用するためのアプリケーションの変更の詳細は、[『Oracle Database SecureFilesおよびラージ・オブジェクト開発者ガイド』](#)を参照してください。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## アップグレードしたOracle Databaseの統合監査の使用への移行

統合監査の全機能を使用するには、手動で統合監査に移行する必要があります。

統合監査では、すべてのOracle Database監査証跡(データベース監査証跡ではSYS. AUD\$, ファイングレイン監査ではSYS. FGA\_LOG\$, Database VaultではDVYS. AUDIT\_TRAIL\$など)は、1つの監査証跡にまとめられますが、これは、単一インスタンスのインストールではUNIFIED\_AUDIT\_TRAILデータ・ディクショナリ・ビューを、Oracle Real Application Clusters環境ではGV\$UNIFIED\_AUDIT\_TRAILを問い合わせることによって表示できます。

- [Oracle Databaseでの統合監査の移行プロセスの理解](#)  
アップグレードしたデータベースで使用する監査ポリシーを決定します。
- [Oracle Databaseでの統合監査への移行](#)  
マルチテナント・コンテナ(CDB)データベースでこの手順を使用して、統合監査に移行します。
- [統合監査への移行後の古い監査レコードの管理について](#)  
統合監査証跡を使用する準備として古い監査証跡を確認、アーカイブおよび削除します。
- [統合監査機能の削除](#)  
この手順を使用して、統合監査を削除し、混合モードの監査を使用します。
- [統合監査を使用しない場合のドキュメント参照の取得](#)  
ここにリストされているドキュメントにアクセスして、非統合監査の使用方法に関する構成情報を取得できます。

参照:

このリリースでの監査機能の変更の詳細は、[『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:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## Oracle Databaseでの統合監査の移行プロセスの理解

アップグレードしたデータベースで使用する監査ポリシーを決定します。

デフォルトでは、アップグレードされたデータベースでは、統合監査は有効化されていません。以前のリリースからOracle Database 12cアップグレードした場合は、データベースでは以前のリリースで使用していたものと同じ監査機能が使用されています。新しく作成されたデータベースでは、統合監査の混合モードの方式がデフォルトで有効化されます。統合監査への移行が完了すると、従来の監査は無効化され、新しい監査レコードが統合監査証跡に書き込まれます。

監査ポリシーを有効化して使用方法を構成するには、次のように1つの方法を選択します。

- 純統合監査機能を使用。

統合監査に移行して、統合監査の全機能を使用します。統合監査への移行手順が完了すると、新しい監査ポリシーを作成および有効化でき、事前定義された監査ポリシーも使用できます。これらのポリシーの監査レコードは、統合監査証跡に書き込まれます。以前の監査証跡および監査レコードは保持されますが、新しい監査レコードは以前の監査証跡には書き込まれません。

ノート:



以前のリリースの監査構成は、統合監査システムでは効果がありません。統合監査証跡内では、統合監査ポリシーのみが監査レコードを生成します。

- 混合モードの監査機能を使用。

混合モードの監査機能は、従来の監査機能と統合監査機能の両方を同時に実行でき、新しいデータベースとアップグレードされたデータベースの両方に適用されます。混合モードの統合監査機能は、1つ以上の事前定義された統合監査ポリシーを有効化すると使用可能になります。これらのポリシーの監査レコードは、統合監査証跡に書き込まれます。Oracle Databaseの以前のリリースでの監査構成も使用可能で、この構成の監査レコードは以前の監査証跡に書き込まれます。純統合監査機能を使用することに決定した場合、それに移行できます。

ノート:



データベースが書き込み不可な場合、監査レコードは、`$ORACLE_BASE/audit/$ORACLE_SID` ディレクトリにある新しい形式のオペレーティング・システム・ファイルに書き込まれます。

参照:

- 事前定義された監査ポリシーの詳細は、[『Oracle Databaseセキュリティ・ガイド』](#)を参照してください。
- ora\_SecureConfig監査ポリシーの詳細は、[『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:** [アップグレードしたOracle Databaseの統合監査の使用への移行](#)

## Oracle Databaseでの統合監査への移行

マルチテナント・コンテナ・データベース(CDB)でこの手順を使用して、統合監査に移行します。

rootで次の手順を実行します。この手順によって、root CDBとそれに関連するPDBの両方が統合監査に移行されます。

ノート:



コンテナ・データベース(CDB)の root からの統合監査のみは無効にできますが、個々のプラグブル・データベース(PDB)に対しては無効にできません。

ただし、統合監査が無効になっている場合、その PDB でローカル監査ポリシーが有効かどうかに応じて、個々の PDB で混合モードの監査を使用できます。CDB 共通監査ポリシーが有効になっている場合、すべての PDB で

混合モード監査が使用されます。

1. SYSDBA権限を持つユーザーSYSとしてSQL\*Plusにログインします。

```
sqlplus sys as sysdba  
Enter password: password
```

マルチテナント環境では、このログインによってrootに接続されます。

2. 次の問合せを使用して、使用しているOracle Databaseが統合監査に移行したかどうかを確認します。

```
SQL> SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
```

VALUE列の出力がTRUEの場合、データベースで統合監査がすでに有効化されています。以前の監査レコードの管理に進むことができます。出力がFALSEの場合は、この手順の残りのステップを完了してください。

3. データベースを停止します。単一インスタンス環境では、SQL\*Plusから次のコマンドを入力します。

```
SQL> SHUTDOWN IMMEDIATE  
SQL> EXIT
```

Windowsシステムでは、Oracleサービスを停止します。

```
net stop OracleService%ORACLE_SID%
```

Oracle Real Application Clusters (Oracle RAC)インストールの場合、次のようにして各データベース・インスタンスを停止します。

```
srvctl stop database -db db_name
```

4. リスナーを停止します。(Oracle RACおよびOracle Grid Infrastructureリスナーでは、リスナーを停止する必要はありません。)

```
lsnrctl stop listener_name
```

lsnrctl statusコマンドを実行すると、リスナーの名前を検索できます。名前はAlias設定で示されます。

5. ディレクトリ\$ORACLE\_HOME/rdbms/libに移動します。

6. Oracleユーザーの統合監査を有効にします。

- LinuxおよびUnix

```
make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME
```

- Microsoft Windows

ファイル名を%ORACLE\_HOME%/bin/orauniau12.dll.dblから%ORACLE\_HOME%/bin/orauniau12.dllに変更します。

ノート:



Oracle ホームを共有していない Oracle RAC データベースの場合、各クラスタ・メンバー・ノードでこのステップを繰り返して、各クラスタ・ノードのローカル ORACLE\_HOME 内のバイナリが更新されるようにする

必要があります。

- リスナーを再起動します。

```
lsnrctl start listener_name
```

- データベースを再起動します。

SQL\*Plusにログインしてから、STARTUPコマンドを入力します。

```
sqlplus sys as sysoper
Enter password: password
SQL> STARTUP
```

Microsoft Windowsシステムの場合、Oracleサービスを開始します。

```
net start OracleService%ORACLE_SID%
```

Oracle RACインストールの場合は、各データベース・インスタンスを起動します。

```
srvctl start database -db db_name
```

**親トピック:** [アップグレードしたOracle Databaseの統合監査の使用への移行](#)

## 統合監査への移行後の古い監査レコードの管理について

統合監査証跡を使用する準備として古い監査証跡を確認、アーカイブおよび削除します。

統合監査を使用するためのOracle Databaseの移行手順が完了すると、データベースで使用していた以前の監査レコードは以前の監査証跡に保持されます。これらの監査レコードをアーカイブして、監査証跡を削除できます。統合監査が使用可能な場合、新しい監査レコードは統合監査証跡に書き込まれます。

参照:

- 『[Oracle Databaseセキュリティ・ガイド](#)』の監査証跡のアーカイブに関する説明
- 『[Oracle Databaseセキュリティ・ガイド](#)』の監査証跡レコードの削除に関する説明

**親トピック:** [アップグレードしたOracle Databaseの統合監査の使用への移行](#)

## 統合監査機能の削除

この手順を使用して、統合監査を削除し、混合モードの監査を使用します。

データベースで統合監査を使用できるようにした後で、統合監査を使用しないことに決定した場合、この手順を使用して、統合監査機能を削除できます。この場合、データベースでは混合モードの監査機能が使用されます。

- データベースを停止します。

```
sqlplus sys as sysoper
Enter password: password
SQL> SHUTDOWN IMMEDIATE
SQL> EXIT
```

Windowsシステムでは、Oracleサービスを停止します。

```
net stop OracleService%ORACLE_SID%
```

Oracle RACインストールの場合は、次のようにして、各データベース・インスタンスを停止します。

```
srvctl stop database -db db_name
```

2. \$ORACLE\_HOME/rdbms/libディレクトリに移動します。

3. 統合監査実行可能ファイルを無効にします。

- Unix: 次のコマンドを実行します。

```
make -f ins_rdbms.mk uniaud_off ioracle ORACLE_HOME=$ORACLE_HOME
```

- Microsoft Windows: %ORACLE\_HOME%/bin/oraunaud12.dllファイルの名前を%ORACLE\_HOME%/bin/oraunaud12.dll.dblに変更します。

4. データベースを再起動します。

```
sqlplus sys as sysoper  
Enter password: password  
SQL> STARTUP  
SQL> EXIT
```

Microsoft Windowsシステムでは、Oracleサービスを再度起動します。

```
net start OracleService%ORACLE_SID%
```

Oracle RACインストールの場合は、次の構文を使用して、各データベース・インスタンスを起動します。

```
srvctl start database -db db_name
```

**親トピック:** [アップグレードしたOracle Databaseの統合監査の使用への移行](#)

## 統合監査を使用しない場合のドキュメント参照の取得

ここにリストされているドキュメントにアクセスして、非統合監査の使用方法に関する構成情報を取得できます。

新しいOracle Databaseリリースへのアップグレード後に、統合監査に変更しないことを選択した場合、OracleドキュメントおよびOracle Technology Networkで従来の非統合監査に関する情報が提供されています。

- *Oracle Database*セキュリティ・ガイド: このガイドは、監査を構成する際の主要な情報源です。このマニュアルのOracle Databaseリリース11gバージョンを使用する必要があります。このガイドにアクセスするには:
  1. 次のOracle Technology Networkのdocs.oracle.comサイトで、データベース・ページにアクセスします。  
<https://docs.oracle.com/en/database/index.html>
  2. 「Oracle Database」を選択します。
  3. 「ダウンロード」ページで、「ドキュメント」タブを選択します。
  4. リリース・リスト・フィールドで「Earlier Releases」を選択し、「Oracle Database 11g Release 2 (11.2)」を選択します。
  5. Oracle Database 11gリリース2 (11.2)の「Documentation」ページから、「All Books」リンクを選択し

て、ドキュメント・セットの刊行物を表示します。

6. 「セキュリティ・ガイド」を検索します。

7. このガイドの「HTML」または「PDF」リンクのいずれかを選択します。

**親トピック:** [アップグレードしたOracle Databaseの統合監査の使用への移行](#)

## 再構築対象のOracle Text索引の識別

新しいOracle Databaseリリースへのアップグレード後にトークン表の再構築が推奨されるOracle Text索引の識別に役立つスクリプトを実行できます。

Oracle Database 12cリリース1 (12.2.0.1)をOracle Database 18c以降のリリースにアップグレードすると、Oracle Textトークン表(\$I、\$Pなど)が64バイトから255バイトに拡張されます。しかし、索引で使用されている既存のトークン表のサイズ範囲が小さいと、拡張後のトークン列範囲をOracle Text索引で利用できなくなります。255バイトのサイズ範囲を使用するように索引を再構築する必要があります。Oracleは、再構築が推奨される索引の識別に役立つスクリプトを提供しています。

My Oracle Supportからスクリプトを取得してください。

<https://support.oracle.com/rs?type=doc&id=2287094.1>

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## DBMS\_SCHEDULERジョブの削除および再作成

以前のリリースからのアップグレード後にDBMS\_SCHEDULERジョブが機能しなくなった場合は、ジョブを削除して再作成します。

アップグレード後にDBMS\_SCHEDULERジョブが機能しないことに気づいたら、それらのジョブを削除して再作成してください。この問題は、アップグレード処理からの問題報告がなく、システム・オブジェクトが有効な場合でも発生する可能性があります。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## アップグレード後の統合監査レコードの転送

統合監査をアップグレードし、移行した後で高いパフォーマンスを得る方法を理解するためには、これらのトピックを確認します。

- [アップグレード後の監査レコードの転送について](#)  
統合監査レコードをOracle Database 12cリリース12.1から、新しいOracle DatabaseリリースのAUDSYSスキーマ下の新しいリレーショナル表に転送すると、統合監査証跡の読取りパフォーマンスが向上します。
- [アップグレード後の統合監査レコードの転送](#)  
DBMS\_AUDIT\_MGMT.TRANSFER\_UNIFIED\_AUDIT\_RECORDS PL/SQLプロシージャを使用して、統合監査レコードをAUDSYSの新しいリレーショナル表に転送できます。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## アップグレード後の監査レコードの転送について

統合監査レコードをOracle Database 12cリリース12.1から、新しいOracle DatabaseリリースのAUDSYSスキーマ下の新しいリレーショナル表に転送すると、統合監査証跡の読取りパフォーマンスが向上します。

Oracle Database 12cリリース2以降、統合監査レコードは直接、AUDSYSスキーマにある新しい内部リレーショナル表に書き込まれます。Oracle Database 12cリリース12.1では、統合監査レコードは共通ロギング・インフラストラクチャ(CLI) SGAキューに書き込まれていました。そのリリースの統合監査に移行した場合は、より高い読取りパフォーマンスを得るために、そのリリースの統合監査レコードを新しいOracle Databaseリリースの内部表に転送できます。この転送を実行することは必須ではありませんが、統合監査証跡の読取りパフォーマンスを向上させるため、転送をお勧めします。これは1回かぎりの操作です。アップグレード後に生成されたすべての新しい統合監査レコードはこの新しい表に書き込まれます。この表は、読取り専用の表です。この表のメタデータまたはデータを変更しようとすると、それらはすべて強制的に監査されます。

新しいOracle Databaseリリースにアップグレードした後に、以前のリリースのUNIFIED\_AUDIT\_TRAILに統合監査レコードが存在している場合は、この転送プロシージャを使用してそれらを新しい内部リレーショナル表に転送し、統合監査証跡の読取りパフォーマンスを高めることを検討してください。

SYSスキーマの場合と同様、SELECT ANY TABLEシステム権限がある場合はAUDSYSスキーマの問合せはできません。さらに、この表はユーザーがSELECT ANY DICTIONARYシステム権限か、この内部表に対する明示的なSELECT権限を持っていないかぎり、ALL\_TABLESデータ・ビューでスキーマ・オブジェクトとしてリストされません。データベースが読取り/書込みモードでオープンするまで、監査レコードはオペレーティング・システムの過剰ファイルに書き込まれます(.bin形式)。ただし、データベースが読取り/書込みモードでオープンした後に、DBMS\_AUDIT\_MGMT.LOAD\_UNIFIED\_AUDIT\_FILESプロシージャを使用してこれらのオペレーティング・システム・ファイルの監査レコードを内部リレーショナル表に転送することはできません。

**親トピック:** [アップグレード後の統合監査レコードの転送](#)

## アップグレード後の監査レコードの転送

DBMS\_AUDIT\_MGMT.TRANSFER\_UNIFIED\_AUDIT\_RECORDS PL/SQLプロシージャを使用して、統合監査レコードをAUDSYSの新しいリレーショナル表に転送できます。

1. AUDIT\_ADMINロールを付与されたユーザーとして、データベース・インスタンスにログインします。

たとえば、非マルチテナント環境では、次のとおりです。

```
sqlplus sec_admin
Enter password: password
```

マルチテナント環境では、rootとして接続します。

```
sqlplus c##sec_admin@root
Enter password: password
```

このプロシージャの実行はPDBだけではなくrootでも実行できます。これは、UNIFIED\_AUDIT\_TRAILビューがテナント別であるからです。さらに、転送手順もテナント別です。つまり、rootからの転送を実行してもPDBの統合監査証跡にある統合監査レコードには影響がありません。

2. マルチテナント環境ではDBA\_PDB\_HISTORYビューに問い合せて、監査レコードの転送元のテナント固有のCLI表

に関連付けられた正しいGUIDを見つめます。

次に例を示します。

```
SQL> SELECT PDB_NAME, PDB_GUID FROM DBA_PDB_HISTORY;
PDB_NAME  PDB_GUID
-----
HR_PDB    33D96CA7862D53DFE0534DC0E40A7C9B
...
```

- マルチテナント環境の、監査レコードの転送を行うコンテナに接続します。

現在接続しているコンテナとは別のコンテナでは転送操作を実行できません。

- DBMS\_AUDIT\_MGMT.TRANSFER\_UNIFIED\_AUDIT\_RECORDSプロシージャを実行します。

次に例を示します。

```
SQL> EXEC DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS;
PL/SQL procedure successfully completed.
```

または、PDB GUIDを指定するために次を実行します。

```
SQL> EXEC DBMS_AUDIT_MGMT.TRANSFER_UNIFIED_AUDIT_RECORDS ('33D96CA7862D53DFE0534DC0E40A7C9B');
PL/SQL procedure successfully completed.
```

- データベースが読み取り/書き込みモードでオープンしている場合は、

DBMS\_AUDIT\_MGMT.LOAD\_UNIFIED\_AUDIT\_FILESプロシージャを実行します。

データベースが読み取り/書き込みモードでオープンするまで、監査レコードはオペレーティング・システム(OS)のファイルに書き込まれます。DBMS\_AUDIT\_MGMT.LOAD\_UNIFIED\_AUDIT\_FILESプロシージャは、そのファイルに存在する統合監査レコードをデータベース表に移動します。OS過剰ファイルに存在する統合監査レコードは、V\$UNIFIED\_AUDIT\_TRAIL動的ビューの問合せにより見つけることができます。

たとえば、このプロシージャをHR\_PDBコンテナの監査レコードに実行する場合は、まず、そのPDBに接続する必要があります。

```
SQL> CONNECT sec_admin@HR_PDB
Enter password: password
SQL> EXEC DBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILES;
PL/SQL procedure successfully completed.
```

- UNIFIED\_AUDIT\_TRAILデータ・ディクショナリに問い合せて、レコードが正しく転送されたことを確認します。

問合せUNIFIED\_AUDIT\_TRAILを行うことを強くお勧めします。監査レコードが正常に転送されたら、問合せUNIFIED\_AUDIT\_TRAILを行う必要があります。これは、V\$UNIFIED\_AUDIT\_TRAIL動的ビューの問合せではOSの過剰ファイルにある監査レコードしか表示されないからです。

**親トピック:** [アップグレード後の統合監査レコードの転送](#)

## Oracle Databaseのアップグレード後のリカバリ・カタログ・アップグレードについて

RMANクライアントで要求されるバージョンより古いリカバリ・カタログ・スキーマを使用している場合、そのカタログをアップグレードする必要があります。

参照:

- RMANリカバリ・カタログの管理の詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください。
- リカバリ・カタログのアップグレードおよびUPGRADE CATALOGコマンドの詳細は、[『Oracle Databaseバックアップおよびリカバリ・ユーザーズ・ガイド』](#)を参照してください

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## アップグレードされたデータベースでの無効なリリース更新バグ修正の有効化

実行計画の変更の原因となる可能性があるリリース更新のバグ修正は無効になっているため、使用する無効なバグ修正を有効にすることをお勧めします。

データベースをアップグレードした後、リリース更新に含まれる実行計画の変更の原因となるバグ修正パッチがデフォルトで無効になっている状態でインストールされます。これらのバグ修正は、修正を有効にするまで有効になりません。これらの修正は、PF FILEコマンドまたはALTER SYSTEMコマンドで手動で有効にすることも、DBMS\_OPTIM\_BUNDLEパッケージを使用することもできます。AutoUpgrade 19.12以降、DBMS\_OPTIM\_BUNDLEパッケージには58個の標準修正が含まれています。これで、DBMS\_OPTIM\_BUNDLEを使用して修正を追加できるようになりました。修正を追加すると、追加した修正はデフォルトの修正に加えて実行されます。

本番システムで使用するこれらの無効になっているパッチを有効にし、アップグレード・テスト計画の一部としてこれらのパッチを使用して完全なワークロード・パフォーマンス・テストを実行することをお勧めします。

DBMS\_OPTIM\_BUNDLEを使用して、実行計画を変更する可能性があるために無効にされたパッチを有効にする方法の詳細は、*Oracle Database PL/SQL*パッケージおよびタイプ・リファレンス、およびMy Oracle Supportノート2147007.1を参照してください。

### 関連項目

- [DBMS\\_OPTIM\\_BUNDLE](#)
- [My Oracle SupportドキュメントID 2147007.1 DBMS\\_OPTIM\\_BUNDLEによるデータベース・リリース更新での「インストールされているが無効になっている」バグ修正の管理](#)

親トピック: [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## アップグレードした本番Oracle Databaseのテストについて

アプリケーションが期待どおり動作することを確認するため、テスト・データベースに実行したテストを本番データベースに対して繰り返します。

テスト・データベースを新しいOracle Databaseリリースにアップグレードしてテストした場合、新しいOracle Databaseリリースにアップグレードした本番データベースでも同じテストを繰り返すことができます。結果を比較し、相違点を記録します。必要に応じて、アップグレードのテストを繰り返します。

新しいOracle Databaseリリースでアプリケーションが適切に動作することを確認するには、新しくアップグレードされた本番データベースを既存のアプリケーションでテストします。また、使用可能なOracle Database機能を追加してテストし、拡張機能をテストすることもできます。ただし、アプリケーションがアップグレードの前と同様に動作するかどうかを最初に確認してください。

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)

## Oracle Databaseのアップグレード後のタイムゾーン・ファイルのバージョンのアップグレード

データベースのアップグレードを完了した後、upgrade.xmlのAutoUpgradeアップグレード前レポートによってタイムゾーン・ファイルをアップグレードするよう指示され、このタスクを完了するようにAutoUpgradeを設定しない場合は、DBMS\_DST PL/SQLパッケージを使用してタイムゾーン・ファイルをアップグレードします。

Oracle Databaseでは、複数のバージョンのタイムゾーン・ファイルを提供しています。各タイムゾーン・ファイルに関連付けられた2つのタイプのファイルがあり、1つはデータベースに定義されたすべてのタイムゾーンを含む大きいファイルで、1つは最も一般的に使用されるタイムゾーンのみを含む小さいファイルです。大きいバージョンは、timezlg\_version\_number.datという名前です。小さいバージョンは、timezone\_version\_number.datという名前です。ファイルは、Oracle Databaseホーム・ディレクトリ下のoracore/zoneinfoサブディレクトリにあります。

### 関連項目

- [DBMS\\_DSTパッケージを使用したタイム・ゾーン・データのアップグレード](#)
- <https://support.oracle.com/rs?type=doc&id=1585343.1>

**親トピック:** [Oracle Databaseのアップグレード後の推奨作業およびベスト・プラクティス](#)