

Oracle Maximum
Availability Architecture

Oracle Databaseローリング・アップグレード Oracle Data Guardフィジカル・スタンバイ・データベースの使用

Oracleホワイト・ペーパー | 2016年8月



ORACLE®

目次

Data Guardを使ったローリング・データベース・アップグレード	2
データベース高速スイッチオーバー	4
要件	4
サービス移行トリガー	4
前提条件および制限	7
ローリング・アップグレード・プロセス	10
アップグレード前の作業	10
エラーからのリカバリ	11
詳細なアップグレード手順	12
実行後の考慮事項	16
複数のスタンバイ・データベースがある場合の考慮事項	16
フォールバックのベスト・プラクティス	17
フラッシュバックの考慮事項	18
まとめ	19
付録A：一時ロジカル・スタンバイでのEDSの使用	20
付録B：physruスクリプトの最初の実行のサンプル出力	22
付録C：physruスクリプトの2回目の実行のサンプル出力	25
付録D：physruスクリプトの最後の実行のサンプル出力	27

概要

Oracle Maximum Availability Architecture (MAA) は、オラクルの高可用性テクノロジーを実装するためのベスト・プラクティス構想です。Oracle Database 11g Release 1 以降、ローリング・データベース・アップグレード（新しいパッチ・セットまたは新しい Oracle Database リリースへのアップグレードなど）の実行に関する Oracle MAA 推奨ベスト・プラクティスでは、Oracle Data Guard で使用可能な一時ロジカル・プロセスを使用します。そのため、既存の Data Guard フィジカル・スタンバイを使ってデータベース・ローリング・アップグレードを実行することができます。ローリング・アップグレード操作の大半を自動化する `physru` という Bourne シェル・スクリプトを My Oracle Support Note [949322.1](#) からダウンロードできます。

一時ロジカル・スタンバイ・ローリング・アップグレードは次のような複数の理由から魅力的です。

- » データベース・アップグレードに伴う停止時間が、数時間から数秒に短縮される可能性があります。
- » データベース・ローリング・アップグレードの実行には既存のフィジカル・スタンバイ・データベースを採用します。ローリング・アップグレードのためだけにロジカル・スタンバイ・データベースを別途デプロイするための追加のストレージや作業は必要ありません。
- » 1 つのカタログをアップグレードするだけで、プライマリ・データベースとスタンバイ・データベースの両方を新しい Oracle リリースに移行できます。
- » アプリケーションとクライアントを新しい環境に切り替える前に、アップグレードとシステムをさらに検証できるため、新しいデータベース・バージョンへのアップグレードのリスクが大幅に減ります。
- » アップグレード・プロセスが完了すると、プライマリ・データベースとフィジカル・スタンバイ・データベースの両方で新しい Oracle リリースが実行されます。
- » プロセスが自動化されているため、一時ロジカル・スタンバイを使用するローリング・データベース・アップグレード・プロセスは比較的デプロイと操作が簡単です。

Oracle MAA ベスト・プラクティスでは、データベースの 1 回限りのパッチに適したオンライン・パッチ、データベース・パッチ/パッチ・セット更新 (PSU) /Critical Patch Update (CPU) に適した Oracle Real Application Clusters (Oracle RAC) ローリング・アップグレード、Cluster Ready Services (CRS) ローリング・アップグレード、Oracle Automatic Storage Management (Oracle ASM) ローリング・アップグレード、および Data Guard Standby-First Patch 適用 (My Oracle Support Note [1265700.1](#) を参照) をできるだけ使用して、他のタイプの計画メンテナンスの可用性を最大化することを推奨します。ただし、このような場合でも、このホワイト・ペーパーに記載のデータベース・ローリング・アップグレード・プロセスには、本番アプリケーションとクライアントをアップグレード済みシステムに切り替える前に、完全に別のシステムとデータベースで変更を適用し、テストできるメリットがあります。

データベース・ローリング・アップグレード・プロセスでは、短時間ながら一時停止して、データベースのロールを変える必要があります。絶対に停止できない場合は、Data Guard データベース・ローリング・アップグレード・プロセスの代わりに、Oracle GoldenGate 双方向レプリケーションを使用する必要があります。Oracle GoldenGate は柔軟性を向上させ、計画メンテナンスの停止時間を完全に排除できますが、Oracle Data Guard を使用する場合より大きな運用投資も必要になります。さまざまな使用例に応じた各種ローリング・アップグレード戦略については、[Oracle Database 高可用性のドキュメント](#)を参照してください。

Data Guardを使ったローリング・データベース・アップグレード

Data Guard一時ロジカル・スタンバイは、新しいパッチ・セットやOracle Databaseの主要リリースにアップグレードする際、ほとんど停止時間なしでデータベース・ローリング・アップグレードを実行するための、Oracle MAA ベスト・プラクティスです。一時ロジカル・ローリング・アップグレード・プロセスはOracle Database 11g Release 1 (11.1) で初めて導入され、既存のフィジカル・スタンバイ・データベースを使うことでデータベース・ローリング・アップグレードの実行をより簡素化しました。アップグレードはフィジカル・スタンバイ・データベースで始まりますが、一時ロジカル・スタンバイ・プロセスではSQL Applyを使って、旧Oracleリリースを実行するデータベースで生成されたREDOを取得し、新しいOracleリリースを実行するスタンバイ・データベースにそのREDOを適用します。アップグレード・プロセスが完了すると、プライマリ・データベースとそのフィジカル・スタンバイ・データベースの両方が新しいOracle Databaseリリースで稼働します。

以下のおおまかな手順は、図1に示す一時ロジカル・データベース・ローリング・アップグレードを説明します。フェーズ2、4、6の自動化は、My Oracle Supportで入手可能なオラクルによって保守およびサポートされるスクリプトで実現します。自動化スクリプトと各手順の詳細については、このドキュメントで後述します。

フェーズ1：前提条件および準備（手動）：

- » 前提条件、制約、ベスト・プラクティスを確認します。
- » アウトオブプレース・アップグレードを準備するために、新しいOracleホームを全ノードにインストールします。
- » (オプション) データベース・サービスのロール変更トリガーとスクリプトを構成し、インストールします。

フェーズ2：physruの最初の実行

- » Data Guard Brokerが無効にされ、FRAが構成されていることを確認します。
- » 保証付きリストア・ポイントを作成します。
- » 既存のフィジカル・スタンバイをロジカル・スタンバイに変換します。

フェーズ3：アップグレード（手動）

- » オラクルのドキュメントに従って、ロジカル・スタンバイ上でアップグレードを手動で実行します。
- » テストを実行してアップグレードを検証します。

フェーズ4：スイッチオーバーするための2回目のphysruの実行（APPLICATION BROWNOUT）：

- » スイッチオーバーを実行して、アップグレード済みのスタンバイ・データベースをプライマリ・データベースにします。
- » 手順1の保証付きリストア・ポイントへ、元のプライマリ・データベースでフラッシュバックを実行し、シャットダウンします。

フェーズ5：新しいOracleホームをインストールで旧プライマリ・データベースをマウント（手動）

フェーズ6：最終となる3回目のphysru実行：

- » REDO Applyを起動します。
- » 元の構成に戻すかどうか、プロンプトで問われます。
- » 保証付きリストア・ポイントを削除します。

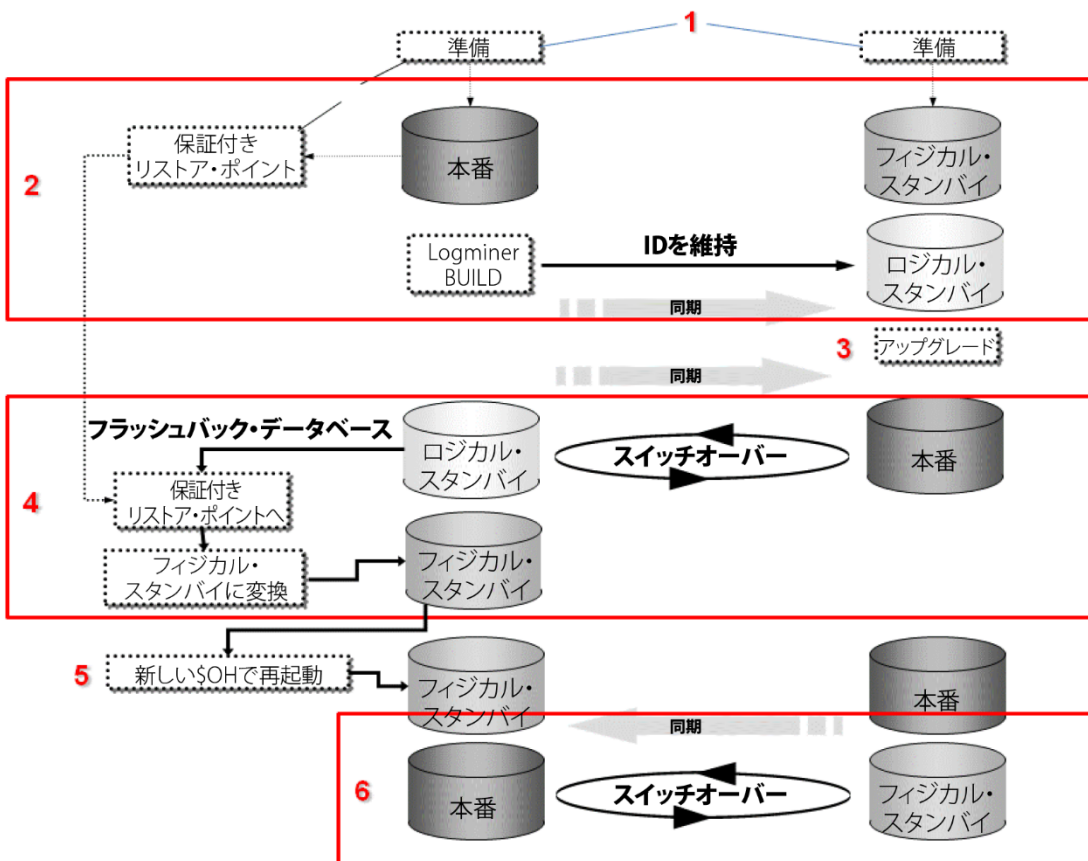


図1:一時ロジカル・ローリング・アップグレード・プロセス

一時ロジカル・ローリング・アップグレード・プロセスを使用すると、計画停止時間は常に、従来のデータベース・アップグレード手法よりも大幅に短縮されます。オラクルの大手顧客企業で一時ロジカル・スタンバイを用いたPoCを実施した結果、データベースの合計停止時間が1分未満で、プライマリ・データベースとスタンバイ・データベースを新しいリリースにアップグレードできることが証明されました。

Oracleデータベースの一般的なアップグレード手順は、『Oracle Databaseアップグレード・ガイド』（RDBMSバージョンごと）で詳述されています。一時ロジカル・スタンバイを使ってローリング・アップグレードを実行する前に、このドキュメントを確認してください。このホワイト・ペーパーに記載のベスト・プラクティスはこのガイドを補完するものですが、すでにガイドに記載された情報を重複して載せる意図はありません。

一時ロジカル・スタンバイ・ローリング・アップグレード・プロセスは、計画メンテナンス操作として考えるべきであり、非ピーク時間に実行する必要があります。アプリケーション停止時間は、`physru`の2回目の実行で始まり（図1のフェーズ4）、通常の操作は、`physru`の3回目の最後の実行後に再開できます（図1のフェーズ6）。

Oracle Database 12.1.0.2以降を実行中の本番データベースは、`physru` スクリプトの代わりに新しい `DBMS_ROLLING PL/SQL` パッケージを使って、ローリング・アップグレード・プロセスを自動化できます。Data Guardとともに使用できる`physru`スクリプトとは違って、`DBMS_ROLLING`を使用するにはOracle Active Data Guardライセンスが必要です。

データベース高速スイッチオーバー

多くのアプリケーションでは、データベースのアップグレードのために、従来のスイッチオーバー処理を使って約1分間の一時停止時間を設ける余裕がありますが、可用性要件がもっとも厳しいアプリケーションの場合、一時停止時間をさらに短縮する必要があります。なおスイッチオーバー・プロセスの最新の機能を利用すると、一時停止時間を10秒に短縮できることがテストで示されています。

要件

この機能を有効にするための要件は次のとおりです。

- ▶ スタンバイREDOログ（SRL）をスタンバイ・データベースで使用します。
- ▶ ロジカル・バイスタンダ・データベースを使用することはできません。
- ▶ ロジカル・スイッチオーバーのパフォーマンス改善に必要なその他の要件については、MOS [949322.1](#)を参照してください。

サービス移行トリガー

ローリング・アップグレード・プロセス中、Data Guardブローカを無効にする必要があります。そのためスイッチオーバー後、新しいプライマリ・データベース上のロールベース・サービスは、自動的に起動されません。なお、サービスを手動で停止および起動を実施すると、アプリケーションの一時停止時間が長くなる可能性があります。ユーザーの介入を最小限にし、かつ最短時間で利用可能な状態に戻るためには、各データベース上で適切なデータベース・サービスを起動および停止するスクリプトを準備して、ロール変更トリガーで呼び出す必要があります。このトリガーにより、データベースのロール変更後、新しいプライマリ・データベースがアプリケーション・サービスを起動し、新しいロジカル・スタンバイ・データベースがデータベース・サービスを停止します。以下に、テスト中に使われたスクリプトの例とスクリプトを呼び出すトリガーを示します。トリガーはプライマリ・データベースにインストールし、`physru`スクリプトの最初の実行前にREDO経由でスタンバイ・データベースへ適用する必要があります。

このプロセスの間はスタンバイ・データベースをActive Data Guard読取り専用ソースとして使用することはできないため、このプロセスの間、読取り専用サービスを移行しないでください。

インストール前に、スクリプト内のORACLE_HOMEとサービス名、トリガーDDL内におけるスクリプトの場所へのパスを編集します。

スクリプトの開始_service.sh

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.4/dbhome_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$ORACLE_HOME/bin/srvctl start service -d ${1} -s oltpru
```

スクリプトの停止_service.sh

```
#!/bin/sh
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.4/dbhome_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$ORACLE_HOME/bin/srvctl stop service -d ${1} -s oltpru
```

ロール変更トリガー

```
create or replace trigger MANAGE_SERVICE AFTER DB_ROLE_CHANGE ON DATABASE DECLARE
role VARCHAR(30);
db_u_nm VARCHAR2(30);
BEGIN
SELECT DATABASE_ROLE INTO role FROM V$DATABASE;
select SYS_CONTEXT('USERENV','DB_UNIQUE_NAME') INTO db_u_nm FROM DUAL;
IF role = 'PRIMARY' THEN
  dbms_scheduler.create_job(
    job_name=>'publish_start',
    job_type=>'executable',
    job_action=>'<path to script>/start_service.sh',
    number_of_arguments=>1,
    enabled=>FALSE
  );
dbms_scheduler.set_attribute(name=>'publish_start',attribute=>'DATABASE_ROLE',value=>'PRIMARY');
dbms_scheduler.set_job_argument_value('publish_start',1,db_u_nm);
dbms_scheduler.enable('publish_start');
dbms_scheduler.run_job('publish_start');
```

```
END IF;
IF role = 'LOGICAL STANDBY' THEN
  dbms_scheduler.create_job(
    job_name=>'publish_stop',
    job_type=>'executable',
    job_action=>'<path to script>/stop_service.sh',
    number_of_arguments=>1,
    enabled=>FALSE
  );
dbms_scheduler.set_attribute(name=>'publish_stop',attribute=>'DATABASE_ROLE',value=>'LOGICAL STANDBY');
dbms_scheduler.set_job_argument_value('publish_stop',1,db_u_nm);
dbms_scheduler.enable('publish_stop'); dbms_scheduler.run_job('publish_stop');
END IF;
EXCEPTION
  WHEN OTHERS THEN
    RAISE_APPLICATION_ERROR (-20000, 'ERROR WITH TRIGGER!!!!');
END;
/
```

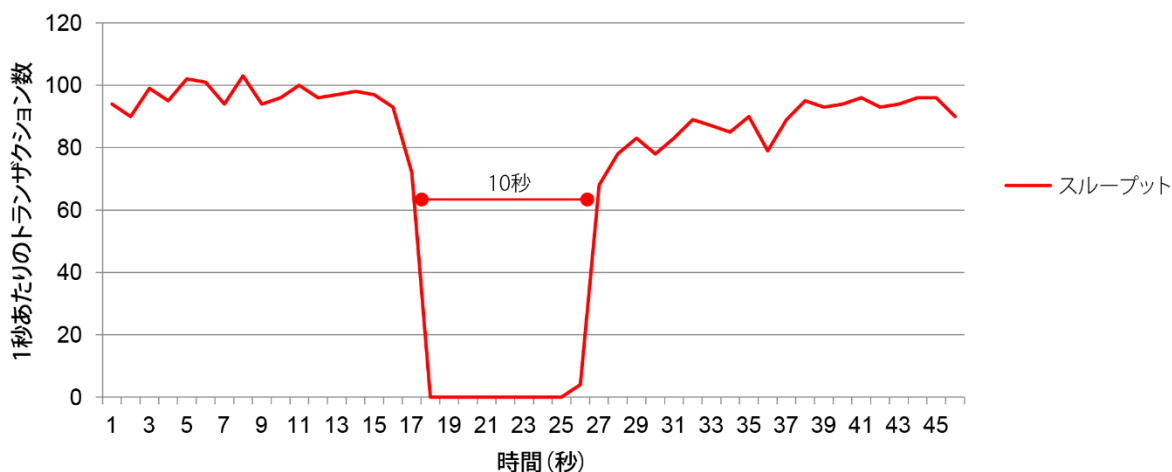
Oracle Clusterwareがない場合は、スクリプトの呼び出しをDBMS_SERVICE.START_SERVICEおよびDBMS_SERVICE.STOP_SERVICEに置き換えることができます。

トリガーおよびスクリプトによるパフォーマンス

スイッチオーバー処理を高速化する、MOS [949322.1](#)に含まれたトリガー、スクリプト、その他の情報により、Oracle MAAのテスト中、アプリケーションの一時停止が約10秒に短縮されました。Oracle RACまたは単一インスタンスのデータベース・タイプのどちらを使用しても、タイミングは同様です。

実際の一時停止時間は、スイッチオーバー時のデータベース・アクティビティやシステム・リソースなど、数多くの要因によって異なる場合があります。

サービス・トリガーによる、ファスト・ロジカル・スイッチオーバー時のアプリケーション・スループット



前提条件および制限

このプロセスでは、ロジカル・スタンバイ・データベースが利用されるため、ロジカル・スタンバイの制限が適用されます。もっとも一般的な制限の一覧を以下に示します。ロジカル・スタンバイの前提条件と制限のリストの詳細については、Oracle Data Guardのドキュメントを参照してください ([11.2|12.1](#))。

- » Oracle Data Guard Brokerを有効にする必要があります。
- » 最初のスタンバイ・データベース上の'processes'パラメータは、'parallel_max_servers'に100を足した値を上回る必要があります。
- » Data Guard保護モードをMAXIMUM PROTECTIONにすることはできません。
- » スタンバイ・データベースのLOG_ARCHIVE_DEST_nはOPTIONALにする必要があります。
- » COMPATIBLE初期化パラメータは、アップグレード前のソフトウェア・リリースと一致する必要があります。従って、リリースXから新しいリリースYへのローリング・アップグレードでは、COMPATIBLE初期化パラメータをプライマリ・データベースとスタンバイ・データベースの両方で、Xに設定する必要があります。アップグレード後、すべての信頼性テストに合格したら、COMPATIBLEパラメータを新しいターゲット・リリースのYに更新できます。COMPATIBLEは11.1.0以上に設定する必要がありますが、強化された機能を有効にするには、11.2に設定します。

COMPATIBLEパラメータをターゲット・データベースのリリースに変更したら、データベースをフラッシュバック・データベースまたはデータベース・ダウングレード・プロシージャで、前のリリースにダウングレードすることはできません。

- » プライマリ・データベースの表の行が一意に識別できることを確認します。バージョンに合った適切な『Oracle Data Guard概要および管理』のドキュメントの「ロジカル・スタンバイ・データベースの作成要件」を参照してください。
- » ロジカル・スタンバイ・データベースでは、Oracle Label Securityはサポートされません。
- » ロジカル・スタンバイ・データベースでは、Oracle E-Business Suite実装は完全にサポートされません。サポート対象外のデータ型を含む表があるためです。
- » データ型の制限 (11.2) : *
 - » BFILE
 - » コレクション (VARRAYおよびネストされた表を含む)
 - » マルチメディア・データ型 (Spatial、Image、Oracle Textを含む)
 - » ROWID、UROWID
 - » ユーザー定義型

拡張データ型サポートを利用すると、データ型の制限を軽減できます。My Oracle Support Note 949516.1を参照してください。

- » データ型の制限 (12.1) : *
 - » BFILE
 - » ROWID、UROWID
 - » コレクション (VARRAYおよびネストされた表を含む)
 - » ネストされた表およびREFを含むオブジェクト
 - » 以下の空間型はサポートされていません。
 - MDSYS.SDO_GEOASTER
 - MDSYS.SDO_TOPO_GEOMETRY
 - » ID列
- » 複数のスタンバイ・データベースがある場合は、ローリング・アップグレードを開始する前に「[複数のスタンバイ・データベースがある場合の考慮事項](#)」の項を確認してください。
- » アップグレードに関わるActive Data Guardスタンバイ・データベースは、このプロセスの実行中、読取り専用ソースとして使用することはできません。

また、このプロセスは以下の前提条件を満たして、確実に実行する必要があります。

- » 既存のフィジカル・スタンバイ・データベース。
- » 強制ロギングを有効にします。回復不能なブロックがないことを確認します。以下の問合せを実行しても、no rowsが返り、行は選択されません。NOLOGGING処理に関する詳細については、MOS [290161.1](#)を参照してください。

```
SQL> select NAME from V$DATAFILE where UNRECOVERABLE_CHANGE#>0;
no rows selected
```

注：データファイルでUNRECOVERABLE_CHANGE#が>0に設定されている場合は、プライマリとスタンバイの値を比較します。値が同じ場合、データファイルは回復不能な変更後にコピーされるので、対応は不要です。

- » ファスト・リカバリ領域（FRA）を構成します（保証付きリストア・ポイント（GRP）の使用のため）。
- » スイッチオーバーを容易にするため、ログ・アーカイブ先を各データベース上で構成する必要があります。ブローカが構成されている場合、ブローカが無効の際はスタンバイでログ・アーカイブ先が構成されないため、手動で構成する必要があります。
- » fal_serverパラメータをすべてのデータベースで設定します。
- » STANDBY_FILE_MANAGEMENT=AUTO
- » DB_FILE_NAME_CONVERTを適宜設定します。 **ファイルが上書きされないようにするため、ローカル・スタンバイ・データベースで特に重要です。**
- » physruスクリプトがデータベース・ローリング・メンテナンス全体を通じてデータベースを起動できるように、各データベースの1つのインスタンスに静的サービスが必要です。まだ静的サービスを設定していない場合は、[My Oracle Support Note 1387859.1の「Oracle Data Guard Broker and Static Service Registration¹」](#)を参照して設定します。
- » 静的サービスを使用するOracle Netサービス名。
- » CDBの場合、TNSサービスはルート・コンテナを参照する必要があります。
- » 既存のリストア・ポイントはこのプロセスでドロップされます。これがアプリケーションで問題にならないことを確認します。
- » アップグレードでは、Data Vaultはサポートされません。アップグレード中、Data Vaultを処理する手順については、MOS [1085051.1](#)を参照してください。

¹ <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1387859.1>

ローリング・アップグレード・プロセス

一時ロジカル・プロセスは、アップグレードのためにロジカル・スタンバイ・データベースに一時的に変換される際、フィジカル・スタンバイ・データベースで始まって終了することから、その名が付けられています。このプロセスは、表のパーティション化、または透過的データ暗号化によるデータの暗号化など、他のタイプの計画メンテナンスの停止時間を短縮する場合にも使用できます。physruというオラクル提供のスクリプトは、手動による手順の多く、およびプロセス全体を通じて自動的に行われるその他のレベルの検証を自動化します。

physrusクリプトには次の6つのパラメータが必要です。

```
$./physru <sysdba user> <primary TNS alias> <physical standby TNS alias> <primary db unique name> <physical standby db unique name> <target version>
```

例：\$./physru sys ru_static rustby_static ru rustby 12.1.0.2.0

スクリプトが実行されるたびに、最初にSYSDBAパスワードの入力を求められます。ローリング・アップグレードに関わるデータベースへのSQL*PlusとSQL*Net接続にアクセスできる限り、どのUnix/Linuxノードからでもスクリプトを実行できます。

アップグレード前の作業

静的リスナー

スクリプトを実行する前に、ローリング・アップグレードに関わるプライマリ・データベースとフィジカル・スタンバイ・データベースの両方に静的なLISTENER.ORAエントリが作成されていることを確認します。Oracle RACの場合、このエントリは1つのインスタンスに対してだけ作成されていけばよく、これらのインスタンスに接続するために作成された有効なTNSエイリアスがphysruの実行元ノードにあることを確認します。以下の例は、Oracle RACデータベースruのインスタンスru1用のプライマリ・データベースのLISTENER.ORAファイルを示します。

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = ru1)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/11.2.0.4/db)
    )
  )
```

以下の例は、Oracle RACデータベースrustbyのインスタンスrustby1用のフィジカル・スタンバイ・データベースのLISTENER.ORAファイルを示します。

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = rustby1)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/11.2.0.4/db)
    )
  )
```

これらの静的エントリにより、データベース・インスタンスのシャットダウン後、スクリプトはこれらのデータベース・インスタンスに接続し、再起動させることができます。LISTENER.ORAファイルの変更後、リスナー・プロセスをリロードします。

```
$lsnrctl reload <listener_name>
```

tnsnames.ora

Oracle RACデータベースごとに、Oracle RACデータベースの単一インスタンスに直接アクセスできるようにTNSエイリアスが定義されていることを確認します。これらのエイリアスは、physruの実行時に適切なパラメータで使用されるようにします。physruが実行中、何回かメディア・リカバリを起動、停止するため、フィジカル・スタンバイ・データベースがOracle RACデータベースの場合、このTNSエイリアスは、メディア・リカバリを実行しているインスタンスを参照する必要があります。メディア・リカバリが実行されているインスタンスに対してだけ、REDO Applyの停止を実行できます。次に例を示します。

```
RU_STATIC =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <host VIP>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = rustby1)
    )
  )
```

ソフトウェアのインストール

環境の全ノードの新しいホームにソフトウェアをインストールします。

前提条件および制約

上記の前提条件と制約をすべて確認し、適宜対策を講じてください。

サービス移行トリガー（オプション）

「Fast Logical Role Transitions」の項で説明されているサービス・トリガーとスクリプトを変更し、作成します。

エラーからのリカバリ

physruスクリプトは、障害発生時点から再開できるようになっています。スクリプトの実行中にエラーが発生した場合は、問題を解決し、physruを再開して続行することができます。スクリプトは、最後の完了したステージを判別し、その時点から実行を再開します。

最後に完了したステージが、計画された終了ポイントではない場合（実行フェーズの完了など）、スクリプトは再開時に、最後の実行ポイントから続行するか、最初から開始するかを選べるメニューを表示します。最初から開始することを選択した場合、必要なクリーンアップ手順がすべて手動で実行され、環境が元のプライマリおよびフィジカル・スタンバイ・データベース構成に戻されたものと見なされます。

詳細なアップグレード手順

1. “physru”アップグレード・スクリプトを最後まで実行します。この最初の実行により、次の処理が行われます。

```
$ physru.sh sys ru_static rustby_static ru rustby 12.1.0.2.0
```

- » スクリプトの残りを実行する前に、環境を検証します。
- » プライマリ・データベースとターゲット・フィジカル・スタンバイ・データベース双方の制御ファイルのバックアップを作成します。
- » 保証付きリストア・ポイント（GRP）をプライマリ・データベースとフィジカル・スタンバイ・データベース双方に作成することで、プロセスの開始時または途中の中間ステップへのフォールバックができるようにします。
- » フィジカル・スタンバイを一時ロジカル・スタンバイ・データベースに変換します。この変換では、スタンバイが単一のインスタンス・モードになっている必要があります。ユーザーはプロンプトにより、CLUSTER_DATABASE=FALSEと設定し、必要な場合はスタンバイ・データベースを再起動するように指示されます。

この最初の実行の間、データベース内にサポート対象外のデータ型がある場合、次の警告が表示されることがあります。

```
WARN:Objects have been identified on the primary database which will not be replicated on the transient logical standby.The complete list of objects and their associated unsupported datatypes can be found in the dba_logstdby_unsupported view.For convenience, this script has written the contents of this view to a file - physru_unsupported.log.
```

```
Various options exist to deal with these objects such as:
```

- disabling applications that modify these objects
- manually resolving these objects after the upgrade
- extending support to these objects (see My Oracle Support Note:559353.1)

```
If you need time to review these options, you should enter 'n' to exit the script. Otherwise, you should enter 'y' to continue with the rolling upgrade.
```

```
Are you ready to proceed with the rolling upgrade? (y/n):
```

このような場合は、実行ディレクトリのphysru_unsupported.logファイルを見て詳細を確認し、適宜問題を解決します。

スクリプトの処理が終了すると、コマンド・プロンプトが再び表示されます。

2. ロジカル・スタンバイからターゲット・リリースにアップグレードします。

アップグレード・スクリプトまたはDatabase Upgrade Assistant（DBUA）を使用して、アップグレード手順を実行できます。アップグレードの前に、そのリリースの『Oracle Databaseアップグレード・ガイド』([12.1|11.2](#))を参照するか、Database READMEを参照してください。

この時点では、`COMPATIBLE`初期化パラメータを変更しないでください。変更すると、ダウングレードすることができなくなります。

一時ロジカル・スタンバイ・データベースへのアップグレードが完了したら、アップグレードされたスタンバイ・データベースを使ってテストを実行できます。スイッチオーバーの前にテストを実施することで、アクティブ・プライマリ・データベースからテストを分離し、本番データに対してテストを実施するというメリットも得られます。この機会を使って、アップグレード後に重要なプロセスが引き続き機能するかを確認します。テスト前に保証付きリストア・ポイント (GRP) を作成し、テストが完了したらこのGRPにフラッシュバックします。これにより、すべてのテストが終わったときに、SQL Applyがこの適切な時点から確実に再開されます。

`physru`スクリプトの実行前に、一時ロジカル・スタンバイ・データベースをそのまま長時間にわたって実行する場合は、一時ロジカル・スタンバイ・データベースの性能特性が要件を満たせるようにSQL Applyを調整します。SQL Applyの推奨事項については、『[SQL Apply Best Practices - Data Guard 11g](#)』ホワイト・ペーパーにあるガイドラインを参照してください。

アップグレードとテストが完了したら、SQL Applyを開始します。SQL Applyを手動で開始するには、SQL*Plusを用い、`physru`スクリプトを使用している一時ロジカル・スタンバイ・データベースのインスタンスに、`SYSDBA`としてログインし、次のコマンドを実行します。

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

スイッチオーバーをできるだけ高速に行うには、この手順でSQL Applyを開始することが重要です。

- 次に、アップグレードした一時ロジカル・スタンバイ・データベースの`LISTENER.ORA`エントリを変更して新しい`ORACLE_HOME`を参照し、リスナーをリロードします。次に例を示します。

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = rustby1)
      (ORACLE_HOME =
        /u01/app/oracle/product/12.1.0/12.1.0.2/db)
      )
    )
  )

$ lsnrctl reload listener
```

Database Upgrade Assistant (DBUA) をアップグレードで使用しなかった場合は、データベース・インスタンスがシームレスに起動できるよう新しい`$ORACLE_HOME`が構成されていることを確認する必要があります。これには、適切な初期化ファイル (`init.ora`) やOracleパスワード・ファイルを`$ORACLE_HOME/dbs`ディレクトリ内に作成することや、現在の`TNSNAMES.ORA`ファイルにアクセス権を付与することなどが含まれます。また、アップグ

ロードされているデータベース・インスタンスの/etc/oratabエントリも更新する必要があります。クラスタウェアがインストールされている場合は、`srvctl upgrade database -d <dbname> -o <new Oracle home>`を実行し、前述の変更が完了していることも確認します。

4. (計画メンテナンスの開始) スタンバイ・データベースのアップグレードが完了し、必要なテストと検証を実施したら、`physru`スクリプトを再実行します (同じコマンドライン・パラメータを使用)。2回目の実行では、停止していたポイントからスクリプトが自動的に再開し、次のタスクが処理されます。

```
$ physru.sh sys ru_static rustby_static ru rustby 12.1.0.2.0
```

- ▶ 一時ロジカル・スタンバイ・データベースがターゲット・バージョンにアップグレードされていて、`OPEN MIGRATE`モードで開始していないことを確認します。この処理には時間がかかることがあるため、`DBA_LOGSTBY_LOG`ビューで進捗状況を監視してください。
- ▶ 一時ロジカル・スタンバイ・データベースがプライマリと同じ最新の状態になっていることを確認します。これには、一時ロジカル・スタンバイのアップグレード時にプライマリ・データベースに行われたすべての変更を適用することが含まれます。
- ▶ アップグレードされた一時ロジカル・スタンバイ・データベースへのスイッチオーバーを実行します。スタンバイ・データベース上のSQL Applyの遅延が、プライマリ・データベースから30秒以下となるまで、スイッチオーバーは実行されません。
 - ▶ スwitchオーバーの完了後、データベース・サービスを起動し、アプリケーションを再接続できます。「Fast Transient Logical Switchovers」で説明されたスクリプトとトリガーが準備されている場合、サービスが自動的に再起動されます。
- ▶ 手順1で作成した最初の保証付きリストア・ポイントに、元のプライマリ・データベースをフラッシュバックします。元のプライマリ・データベースがOracle RACデータベースの場合、フラッシュバック操作を実行する前に、元のプライマリ・データベースを除くすべてのインスタンスをシャットダウンする必要があります。`physru`スクリプトの実行が一時停止し、適切なときにシャットダウンを実行するように指示されます。
- ▶ 元のプライマリ・データベースをフィジカル・スタンバイ・データベースに変換します。
- ▶ 新しいフィジカル・スタンバイ・データベースをシャットダウンします。

スクリプトの処理が終了し、コマンド・プロンプトが再び表示されます。

5. 新しいフィジカル・スタンバイを準備します。

元のプライマリ・データベースの定義がOracle Cluster Registry (OCR) に保存されている場合は、OCRを更新してデータベースの新しいORACLE_HOMEを保存します (新しいホームへのパスワード・ファイルの手動コピーが必要です。また、`tnsnames.ora`と`listener.ora`がORACLE_HOMEから使用されていた場合、これらの更新が必要なことがあります)。

```
$ srvctl upgrade database -d ru -o /u01/app/oracle/product/12.1.0/12.1.0.2/db
```


次に、新しいスタンバイ・データベースをマウントします。

新しいスタンバイ・データベースをマウントしない場合は、手順1でインストールした、アップグレード済みのOracleホームを含むデータベースを手動でマウントします。これはアウトオブプレース・アップグレードであり、ソフトウェアがすでにインストールされているため、`oratab`を新しいOracleホームに変更し、`init.ora/spfile`とパスワード・ファイルを正しい場所にコピーし、環境をリセットし、データベースをマウントしてください。構成によっては、静的サービスの`listener.ora`と`tnsnames.ora`の変更も必要なことがあります。

- 最後に、`physru`の3回目の実行を行って、新しいプライマリによって生成されたすべてのREDOに新しいスタンバイ（元のプライマリ）を同期させます。これには、一時ロジカル・データベースのアップグレード中に元のプライマリ・データベースで発生したすべてのアプリケーション・アクティビティ、アップグレード自体に直接関連するすべてのREDO、初回スイッチオーバー後に新しいプライマリで発生したすべてのトランザクションが含まれます。

```
$ physru.sh sys ru_static rustby_static ru rustby 12.1.0.2.0
```

`physru`のこの最後の実行によって次の処理が行われます。

- 新しいフィジカル・スタンバイ・データベース（元のプライマリ・データベース）上でREDO Applyを起動し、このプロセスの間に生成されたすべてのREDOを、アップグレードの一部として一時ロジカル・スタンバイ上で実行されたすべてのSQL文を含めて適用します。
- 進捗状況の定期的なステータスを確認し、フィジカル・スタンバイ・データベースがプライマリ・データベースと同期するまで待機します。スクリプトがタイムアウトしたら再起動します。
- 同期されると、更新されたソフトウェア上の元のロールにデータベースを戻すための最終的なスイッチオーバーを実行するオプションを選べます。このスクリプトは、以下が完了するまでスイッチオーバーを実行しません。
- REDO Applyが正常に起動し、アップグレード・プロセス中のすべてのREDOを適用する。
- スタンバイ・データベース上のSQL Applyがプライマリ・データベースを30秒以下の間、遅延させる。
- `physru`によって作成されたすべての保証付きリストア・ポイントを削除する。

監視している場合は、次のメッセージがアラート・ログに表示されることがあります。

```
ORA-19906: recovery target incarnation changed during recovery Managed Standby Recovery not using Real Time Apply
```

```
Recovery Slave PR00 previously exited with exception 19906
```

これは、新しいプライマリでリモート・ログ・アーカイブ先がタイムアウトしたためです。このメッセージは無視してかまいません。最終的には再接続されます。または、新しいプライマリで`log_archive_dest_state_n=enable`を設定して、再接続を実行させます。

実行後の考慮事項

プロセスの完了後、環境で以下のいずれかを実行する必要があるかを確認します。

- » ロール変更トリガーをドロップする（作成されていた場合）。

```
SQL> drop trigger MANAGE_SERVICE;
```
- » トリガーによって作成されたジョブをドロップする。

```
SQL> execute DBMS_SCHEDULER.drop_job(job_name => 'PUBLISH_STOP',  
force => true);  
SQL> execute DBMS_SCHEDULER.drop_job(job_name => 'PUBLISH_START',  
force => true);
```
- » 必要に応じて、各データベースのクラスタウェアの起動オプションを確認する。
- » 必要に応じて、Oracle Data Guard Brokerの構成を再度有効にする
- » プロセスで追加された静的エントリをlistener.oraから削除する。
- » プロセスで追加されたtnsnames.oraのエントリを削除する。

複数のスタンバイ・データベースがある場合の考慮事項

データ保護の強化、HA、ディザスタ・リカバリ、またはActive Data GuardのReader Farmのために、環境に複数のスタンバイ・データベースが存在する場合があります。その場合でも、一時ロジカル・スタンバイ手順を使ってデータベース・ローリング・アップグレードを実行できます。ただし、追加の手順と考慮事項に従ってください。

バイスタンダ・フィジカル・スタンバイ・データベースの考慮事項

physruスクリプトは、アップグレード・プロセス中、バイスタンダ・フィジカル・スタンバイ・データベースを直接管理しません。ただし、次の手順を実行して、手動でアップグレードすることができます。

1. スクリプトを初めて実行する前に（手順1）、アップグレードの完了後にData Guard構成で保持する各バイスタンダ・フィジカル・スタンバイ・データベース上で保証付きリストア・ポイント(GRP)を作成します。次の手順に従って、GRPを作成します。
 - a. バイスタンダ・フィジカル・スタンバイ・データベース上のREDO Applyを停止します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```
 - b. SYSDBAユーザーとしてGRPを作成します。

```
SQL> CREATE RESTORE POINT PRE_UPGRADE GUARANTEE FLASHBACK DATABASE;
```
 - c. REDO Applyを起動します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT  
LOGFILE DISCONNECT;
```

Oracle Database 12cの場合、USING CURRENT LOGFILE句は不要です。

2. 各バイスタンダ・スタンバイで新しいOracleホームをインストールします。
3. 「詳細なアップグレード手順」の項に記載のphysruを使って、アップグレードを実行します。
4. physruスクリプトの最後の実行後（手順8）：
 - a. バイスタンダ・フィジカル・スタンバイ・データベース上のREDO Applyを停止します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```
 - b. バイスタンダ・データベースがOracle RACデータベースの場合、1つを除くすべてのデータベース・インスタンスをシャットダウンします。

```
$ srvctl stop instance -d <bystander database name> -i <bystander instance> -o abort
```
 - c. バイスタンダ・フィジカル・スタンバイ・データベースをPRE_UPGRADE_GRPにフラッシュバックします。

```
SQL> FLASHBACK DATABASE TO RESTORE POINT PRE_UPGRADE;
```
 - d. バイスタンダ・スタンバイ・データベースをシャットダウンします。
 - e. アップグレード済みのOracleホームを使って、バイスタンダ・フィジカル・スタンバイ・データベースをマウントします。
 - f. バイスタンダ・フィジカル・スタンバイ・データベース上のREDO Applyを起動します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT;
```
 - g. バイスタンダ・フィジカル・スタンバイ・データベースがOracle RACデータベースの場合、残りのデータベース・インスタンスを起動します。

```
$ srvctl start instance -d <bystander database name> -i <bystander instance> -o <startup option>
```
 - h. 保証付きリストア・ポイントをドロップします。

```
DROP RESTORE POINT PRE_UPGRADE;
```

フォールバックのベスト・プラクティス

バックアップ

アップグレード・プロセスの開始前に、プライマリ・データベースとスタンバイ・データベース上でデータベースとソフトウェアのバックアップをとっておいてください。ソフトウェアのバックアップには、oralnventoryディレクトリ・ツリーを含める必要があります。ソフトウェアのバックアップをとったことがなく、新たにインストールされた別個のORACLE_HOMEにパッチ・セットを適用するのではなく、既存のORACLE_HOMEツリーに直接適用する場合にだけ、ソフトウェアのバックアップをとる必要があります。

保証付きリストア・ポイント

何らかの理由で、physruスクリプトを2回目に実行する前にフラッシュバックを実行する必要がある場合は、一時ロジカル・スタンバイ・データベースをフィジカル・スタンバイに変換し直すために追加の操作を実行する必要があります。これは、将来のパッチで解決される予定の既知のバグがあるためです。

一時ロジカル・スタンバイをリストア・ポイントPRU_0000_0002にフラッシュバックします。

```
SQL> STARTUP MOUNT FORCE  
  
SQL> FLASHBACK DATABASE TO RESTORE POINT PRU_0000_0002;  
  
SQL> ALTER DATABASE CONVERT TO PHYSICAL STANDBY;  
  
SQL> SHUTDOWN ABORT
```

一時ロジカル・スタンバイ・データベースで新しいバージョンの\$ORACLE_HOMEが使われている場合は、環境をリセットし、元の\$ORACLE_HOMEを使用するデータベースをマウントします。

データベースのマウント後、SYSDBAとして次のコマンドを実行します。

```
SQL> ALTER SYSTEM SET _TRANSIENT_LOGICAL_CLEAR_HOLD_MRP_BIT=TRUE;
```

REDO Applyを起動します。

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE [ USING CURRENT LOGFILE ]  
DISCONNECT;
```

physruスクリプトによって作成された保証付きリストア・ポイントを後の時点で削除します。

ダウングレード

ダウングレード手順を実行する場合は、アップグレードしたバージョンの適切なアップグレード・ガイドに記載された、手動のダウングレード手順に従います。ダウングレードの実行は、一般にアップグレード手順と同じくらい時間がかかり、トランザクションの変更が維持される一方で、パッチ・セットの変更は取り消されます。アップグレード後にトランザクションの変更が生じ、データをリストアする方法がほかにない場合は、ダウングレード手順を実行する必要があります。ダウングレードの前提条件は、COMPATIBLEデータベース・パラメータをダウングレードするリリースの値のままにする必要があることです。

フラッシュバックの考慮事項

physruスクリプトの実行中、複数の保証付きリストア・ポイントが作成されますが、これらのGRPは主に、進捗ポイントを維持するためにスクリプトによって使用されます。physruによって作成されるGRPのうち、手動で使用する必要がある唯一のGRPは、最初に生成されたGRPです。このGRPは、処理が行われる前に作成され、フラッシュバックで安全に使用してすべての変更を取り消すことができます。physruによって作成された他のGRPをフラッシュバック・ポイントとして使用しようとすると、望ましくない結果になることがあります。

アップグレードを放棄する必要がある場合は、選択したフラッシュバック・オプションの実行後、スクリプトによって作成されたすべての保証付きリストア・ポイントがドロップされたことを確認してください。



まとめ

一時ロジカル・ローリング・アップグレード・プロセスを使用すると、アプリケーションの可用性への影響を大幅に軽減できます。Oracleによって保守およびサポートされるスクリプトを使って、既存のフィジカル・スタンバイ・データベースをロジカル・スタンバイ・データベースに変換すると、1つのカタログのアップグレードで両方のデータベースのアップグレードを容易に完了できます。このホワイト・ペーパーで説明したスイッチオーバー・プロセスのさらに進化した最新の拡張機能により、本番環境を新しいデータベース・バージョンにスイッチオーバーする際、アプリケーションの一時停止をさらに短縮して、もっとも厳しい品質保証契約も満たすことができます。

付録A：一時ロジカル・スタンバイでのEDSの使用

拡張データ型サポート（EDS）の概要については、Oracle MAAホワイト・ペーパー『[拡張データ型サポート](#)』を参照してください。下位データベース・バージョンの変更を上位データベース・バージョンにレプリケートするデータベース・ローリング・アップグレード・プロセスではSQL Applyが使用されますが、SQL Applyプロセスによってネイティブにサポートされないデータ型がデータベースに含まれている場合は、EDSが役立ちます。

SQL Applyによってネイティブにサポートされないデータ型を、EDSを使ってサポートする場合の詳細と使用例については、以下のSupport Noteを参照してください。

- » 11.2より以前 - [My Oracle Support Note 559353.1](#)
- » 11.2以降 - [My Oracle Support Note 949516.1](#)

スタンバイがロジカル・スタンバイのときに一時ロジカルでEDSを使用し、いったんスタンバイをフィジカル・スタンバイに変換したら、EDSは不要になります。

EDSの準備

フィジカル・スタンバイをロジカル・スタンバイに変換する前に、以下の手順を実行してEDSを設定します。これらの手順は、EDSサポートが必要な表ごとに、プライマリ・データベース上で実行します。

1. プライマリ・データベース上で、EDSサポートが必要な表を含むスキーマごとにDBMS_LOGSTDBYパッケージの実行権限を付与します。

```
e.g. (where OBJUSER is the schema)
SQL> connect system/<password>
SQL> grant execute on dbms_logstdby to OBJUSER;
```

2. プライマリ・データベース上に次のようなロギング表を作成します。

```
OBJUSER.PLAYERS_LOG
```

3. プライマリ・データベース上に次のような実表トリガーを作成します。

```
OBJUSER.PLAYERS_PRI_TRG
```

4. プライマリ・データベース上に次のようなロギング表トリガーを作成します。

```
OBJUSER.PLAYERS_STBY_TRG
```

開始リリースがOracle Database 11g Release 1 (11.1.0.6) の場合、すべてのロギング表トリガーの作成が無効にされます。次に例を示します。

```
SQL> create trigger PLAYERS_STBY_TRG disable ...
```

Oracle Database 11g Release 1 (11.1.0.6) の場合、ロギング表トリガーによって、コンパイル中に警告が生成され、無効にしないと実表の操作に影響が及びます。上位リリースにアップグレードされた後、無効なロギング表トリガーはロジカル・スタンバイ・データベース上で正常にコンパイルされ、有効にされます。

EDSロギング表トリガーの構成

フィジカル・スタンバイがロジカル・スタンバイになり、ロジカル・スタンバイがアップグレードされたら、ロギング表トリガーを構成します。これらのコマンドは、EDSによって使用されるロギング表トリガーごとに、アップグレード済みのロジカル・スタンバイ・データベース上で実行されます。

SQL Applyを初めて起動する前に、以下の手順を実行する必要があります。これらの手順を実行する前にSQL Applyを起動すると、アップグレード中、プライマリ・データベース上のEDSによってサポートされた表に対して実行された更新は、アップグレードされたロジカル・スタンバイ・データベースに適用されません。

1. SYSとしてGUARDを一時的に無効にしてから有効にし、ロギング表トリガーをコンパイルします。

```
SQL> alter session disable guard;
SQL> alter trigger OBJUSER.PLAYERS_STBY_TRG compile;
SQL> alter trigger OBJUSER.PLAYERS_STBY_TRG enable;
SQL> alter session enable guard;
```

2. すべてのロギング表トリガーのFIRE_ONCEトリガー実行プロパティをFALSEに設定します。

```
SQL> execute dbms_ddl.set_trigger_firing_property('OBJUSER',
'PLAYERS_STBY_TRG', fire_once => FALSE);
```

EDSの削除

スイッチオーバーが実行され、アップグレードされたロジカル・スタンバイが新しいプライマリになったら、プライマリのすべてのEDSオブジェクトをドロップします。

```
SQL> drop trigger OBJUSER.PLAYERS_STBY_TRG;
SQL> drop trigger OBJUSER.PLAYERS_PRI_TRG;
SQL> drop table OBJUSER.PLAYERS_LOG;
```

付録B：physruスクリプトの最初の実行のサンプル出力

```
### Initialize script to either start over or resume execution
Jul 21 16:07:18 2016 [0-1] Identifying rdbms software version
Jul 21 16:07:18 2016 [0-1] database ru is at version 11.2.0.4.0
Jul 21 16:07:18 2016 [0-1] database rustby is at version 11.2.0.4.0
Jul 21 16:07:18 2016 [0-1] verifying fast recovery area is enabled at ru and rustby
Jul 21 16:07:18 2016 [0-1] verifying available flashback restore points
Jul 21 16:07:18 2016 [0-1] verifying DG Broker is disabled
Jul 21 16:07:19 2016 [0-1] looking up prior execution history
Jul 21 16:07:19 2016 [0-1] purging script execution state from database ru
Jul 21 16:07:19 2016 [0-1] purging script execution state from database rustby
Jul 21 16:07:20 2016 [0-1] starting new execution of script
```

```
### Stage 1:Backup user environment in case rolling upgrade is aborted
Jul 21 16:07:20 2016 [1-1] stopping media recovery on rustby
Jul 21 16:07:21 2016 [1-1] creating restore point PRU_0000_0002 on database rustby
Jul 21 16:07:35 2016 [1-1] backing up current control file on rustby
Jul 21 16:07:35 2016 [1-1] created backup control file
    /u01/app/oracle/product/11.2.0.4_160119DBPSU/dbhome_1/dbs/PRU_0002_rustby_f.f
Jul 21 16:07:35 2016 [1-1] creating restore point PRU_0000_0002 on database ru
Jul 21 16:07:37 2016 [1-1] backing up current control file on ru
Jul 21 16:07:38 2016 [1-1] created backup control file
    /u01/app/oracle/product/11.2.0.4_160119DBPSU/dbhome_1/dbs/PRU_0002_ru_f.f
```

NOTE:Restore point PRU_0000_0002 and backup control file PRU_0002_rustby_f.f can be used to restore rustby back to its original state as a physical standby, in case the rolling upgrade operation needs to be aborted prior to the first switchover done in Stage 4.

```
### Stage 2:Create transient logical standby from existing physical standby
Jul 21 16:07:39
2016 [2-1] verifying RAC is disabled at rustby
```

WARN: rustby is a RAC database.Before this script can continue, you must manually reduce the RAC to a single instance, disable the RAC, and restart instance rustby2 in mounted mode.This can be accomplished with the following steps:

- 1) Shutdown all instances other than instance rustby2.
eg: `srvctl stop instance -d rustby -i rustby1 -o abort`
- 2) On instance rustby2, set the `cluster_database` parameter to FALSE.
eg: `SQL> alter system set cluster_database=false scope=spfile;`
- 3) Shutdown instance rustby2.
eg: `SQL> shutdown abort;`

4) Startup instance rustby2 in mounted mode.

eg:SQL> startup mount;


Once these steps have been performed, enter 'y' to continue the script. If desired, you may enter 'n' to exit the script to perform the required steps, and recall the script to resume from this point.

Are you ready to continue? (y/n):

```
Jul 21 16:08:14 2016 [2-1] continuing
Jul 21 16:08:14 2016 [2-1] verifying RAC is disabled at rustby
Jul 21 16:08:14 2016 [2-1] verifying database roles
Jul 21 16:08:14 2016 [2-1] verifying physical standby is mounted
Jul 21 16:08:14 2016 [2-1] verifying database protection mode
Jul 21 16:08:15 2016 [2-1] verifying transient logical standby datatype support
Jul 21 16:08:16 2016 [2-2] starting media recovery on rustby
Jul 21 16:08:22 2016 [2-2] confirming media recovery is running
Jul 21 16:08:22 2016 [2-2] waiting for apply lag to fall under 30 seconds
Jul 21 16:08:28 2016 [2-2] apply lag measured at 6 seconds
Jul 21 16:08:28 2016 [2-2] stopping media recovery on rustby
Jul 21 16:08:29 2016 [2-2] executing dbms_logstdby.build on database ru
Jul 21 16:08:56 2016 [2-2] converting physical standby into transient logical standby
Jul 21 16:09:01 2016 [2-3] opening database rustby
Jul 21 16:09:02 2016 [2-4] configuring transient logical standby parameters for rolling
upgrade
Jul 21 16:09:03 2016 [2-4] starting logical standby on database rustby
Jul 21 16:09:10 2016 [2-4] enabling log archive destination to database rustby
Jul 21 16:09:10 2016 [2-4] waiting until logminer dictionary has fully loaded
Jul 21 16:11:52 2016 [2-4] dictionary load 33% complete
Jul 21 16:12:02 2016 [2-4] dictionary load 62% complete
Jul 21 16:12:12 2016 [2-4] dictionary load 75% complete
Jul 21 16:13:12 2016 [2-4] dictionary load is complete
Jul 21 16:13:13 2016 [2-4] waiting for apply lag to fall under 30 seconds
Jul 21 16:13:20 2016 [2-4] apply lag measured at 7 seconds
```

NOTE:Database rustby is now ready to be upgraded.This script has left the database open in case you want to perform any further tasks before upgrading the database.Once the upgrade is complete, the database must be opened in READ WRITE mode before this script can be called to resume the rolling upgrade.

NOTE:Database rustby may be reverted back to a RAC database upon completion of the rdbms upgrade.This can be accomplished by performing the following steps:



1) On instance rustby2, set the cluster_database parameter to TRUE.

eg:SQL> alter system set cluster_database=true scope=spfile;

2) Shutdown instance rustby2.

eg:SQL> shutdown abort;

3) Startup and open all instances for database rustby.

eg: srvctl start database -d rustby

付録C：physruスクリプトの2回目の実行のサンプル出力

```
### Initialize script to either start over or resume execution
Jul 21 16:16:45 2016 [0-1] Identifying rdbms software version
Jul 21 16:16:45 2016 [0-1] database ru is at version 11.2.0.4.0
Jul 21 16:16:45 2016 [0-1] database rustby is at version 12.1.0.2.0
Jul 21 16:16:46 2016 [0-1] verifying fast recovery area is enabled at ru and rustby
Jul 21 16:16:46 2016 [0-1] verifying available flashback restore points
Jul 21 16:16:46 2016 [0-1] verifying DG Broker is disabled
Jul 21 16:16:46 2016 [0-1] looking up prior execution history
Jul 21 16:16:46 2016 [0-1] last completed stage [2-4] using script version 0002
Jul 21 16:16:46 2016 [0-1] resuming execution of script

### Stage 3:Validate upgraded transient logical standby
Jul 21 16:16:46 2016 [3-1] database rustby is no longer in OPEN MIGRATE mode
Jul 21 16:16:46 2016 [3-1] database rustby is at version 12.1.0.2.0


### Stage 4:Switch the transient logical standby to be the new primary
Jul 21 16:16:48 2016 [4-1] waiting for rustby to catch up (this could take a while)
Jul 21 16:16:49 2016 [4-1] starting logical standby on database rustby
Jul 21 16:16:54 2016 [4-1] waiting for apply lag to fall under 30 seconds
Jul 21 16:16:58 2016 [4-1] apply lag measured at 4 seconds
Jul 21 16:17:00 2016 [4-2] using fast switchover optimizations
NOTE:A switchover is about to be performed which will incur a brief outage of the primary
      database.If you answer 'y' to the question below, database rustby will become the
      new primary database, and database ru will be converted into a standby in preparation
      for upgrade.If you answer 'n' to the question below, the script will exit, leaving
      the databases in their current roles.
Are you ready to proceed with the switchover? (y/n): y

Jul 21 16:17:16 2016 [4-2] continuing
Jul 21 16:17:00 2016 [4-2] switching ru to become a logical standby
Jul 21 16:17:16 2016 [4-2] ru is now a logical standby
Jul 21 16:17:16 2016 [4-2] waiting for standby rustby to process end-of-redo from primary
Jul 21 16:17:16 2016 [4-2] switching rustby to become the new primary
Jul 21 16:17:18 2016 [4-2] rustby is now the new primary

### Stage 5:Flashback former primary to pre-upgrade restore point and convert to physical
Jul 21 16:17:20 2016 [5-1] verifying instance rul is the only active instance

WARN: ru is a RAC database.Before this script can continue, you must manually reduce the
      RAC to a single instance.This can be accomplished with the following step:

      1) Shutdown all instances other than instance rul.
         eg: srvctl stop instance -d ru -i ru2 -o abort
```



Once these steps have been performed, enter 'y' to continue the script. If desired, you may enter 'n' to exit the script to perform the required steps, and recall the script to resume from this point.

Are you ready to continue? (y/n):

```
Jul 21 16:17:46 2016 [5-1] continuing
Jul 21 16:17:46 2016 [5-1] verifying instance ru1 is the only active instance
Jul 21 16:17:46 2016 [5-1] shutting down database ru
Jul 21 16:18:16 2016 [5-1] mounting database ru
Jul 21 16:18:28 2016 [5-2] flashing back database ru to restore point PRU_0000_0002
Jul 21 16:18:35 2016 [5-3] converting ru into physical standby
Jul 21 16:18:37 2016 [5-4] shutting down database ru
```

NOTE: Database ru has been shutdown, and is now ready to be started using the newer version Oracle binary. This script requires the database to be mounted (on all active instances, if RAC) before calling this script to resume the rolling upgrade.

NOTE: Database ru is no longer limited to single instance operation since the database has been successfully converted into a physical standby. For increased availability, Oracle recommends starting all instances in the RAC on the newer binary by performing the following step:

- 1) Startup and mount all instances for database ru
eg: `srvctl start database -d VISMIG`

付録D : physruスクリプトの最後の実行のサンプル出力

```
### Initialize script to either start over or resume execution
Jul 21 16:23:00 2016 [0-1] Identifying rdbms software version
Jul 21 16:23:00 2016 [0-1] database ru is at version 12.1.0.2.0
Jul 21 16:23:00 2016 [0-1] database rustby is at version 12.1.0.2.0
Jul 21 16:23:00 2016 [0-1] verifying fast recovery area is enabled at ru and rustby
Jul 21 16:23:01 2016 [0-1] verifying available flashback restore points
Jul 21 16:23:01 2016 [0-1] verifying DG Broker is disabled
Jul 21 16:23:01 2016 [0-1] looking up prior execution history
Jul 21 16:23:01 2016 [0-1] last completed stage [5-4] using script version 0002
Jul 21 16:23:01 2016 [0-1] resuming execution of script

### Stage 6:Run media recovery through upgrade redo
Jul 21 16:23:01 2016 [6-1] upgrade redo region identified as scn range [4180883, 4205367]
Jul 21 16:23:01 2016 [6-1] enabling log archive destination to database ru
Jul 21 16:23:02 2016 [6-1] starting media recovery on ru
Jul 21 16:23:08 2016 [6-1] confirming media recovery is running
Jul 21 16:23:18 2016 [6-1] waiting for media recovery to initialize v$recovery_progress
Jul 21 16:24:40 2016 [6-1] monitoring media recovery's progress
Jul 21 16:24:42 2016 [6-2] last applied scn 4170971 is approaching upgrade redo start
scn 4180883
Jul 21 16:24:58 2016 [6-4] media recovery has finished recovering through upgrade

### Stage 7:Switch back to the original roles prior to the rolling upgrade

NOTE:At this point, you have the option to perform a switchover which will restore ru
      back to a primary database and rustby back to a physical standby database.If you
      answer 'n' to the question below, ru will remain a physical standby database and
      rustby will remain a primary database.

Do you want to perform a switchover?(y/n):

Jul 21 16:33:02 2016 [7-1] continuing
Jul 21 16:33:02 2016 [7-2] verifying instance rustby2 is the only active instance

WARN: rustby is a RAC database.Before this script can continue, you must manually reduce
      the RAC to a single instance.This can be accomplished with the following step:

      1) Shutdown all instances other than instance rustby2.
         eg: srvctl stop instance -d rustby -i rustby1

Once these steps have been performed, enter 'y' to continue the script.If desired,
you may enter 'n' to exit the script to perform the required steps, and recall the
script to resume from this point.
```

Are you ready to continue? (y/n):

```
Jul 21 16:33:02 2016 [7-1] continuing
Jul 21 16:33:02 2016 [7-2] verifying instance rustby2 is the only active instance
Jul 21 16:33:54 2016 [7-2] waiting for apply lag to fall under 30 seconds
Jul 21 16:34:04 2016 [7-2] apply lag measured at 10 seconds
Jul 21 16:34:04 2016 [7-3] switching rustby to become a physical standby
Jul 21 16:34:07 2016 [7-3] rustby is now a physical standby
Jul 21 16:34:07 2016 [7-3] shutting down database rustby
Jul 21 16:34:07 2016 [7-3] mounting database rustby
Jul 21 16:34:18 2016 [7-3] starting media recovery on rustby
Jul 21 16:23:08 2016 [6-1] confirming media recovery is running
Jul 21 16:34:25 2016 [7-3] waiting for standby ru to process end-of-redo from primary
Jul 21 16:34:25 2016 [7-3] switching ru to become the new primary
Jul 21 16:34:26 2016 [7-3] ru is now the new primary
Jul 21 16:34:26 2016 [7-3] opening database ru
```

NOTE:Database ru has completed the switchover to the primary role, but instance rul is the only open instance.For increased availability, Oracle recommends opening the remaining active instances which are currently in mounted mode by performing the following steps:

1) Shutdown all instances other than instance rul.

eg: `srvctl stop instance -d ru -i ru2`

2) Startup and open all inactive instances for database ru.

eg: `srvctl start database -d ru`

NOTE:Database rustby is no longer limited to single instance operation since it has completed the switchover to the physical standby role.For increased availability, Oracle recommends starting the inactive instances in the RAC by performing the following step:

1) Startup and mount inactive instances for database rustby

eg: `srvctl start database -d rustby -o mount`

Stage 8:Statistics

script start time:	21-Jul-16 16:07:21
script finish time:	21-Jul-16 16:34:41
total script execution time:	+00 00:27:20
wait time for user upgrade:	+00 00:03:26
active script execution time:	+00 00:23:54
transient logical creation start time:	21-Jul-16 16:08:15
transient logical creation finish time:	21-Jul-16 16:09:01
primary to logical switchover start time:	21-Jul-16 16:16:58
logical to primary switchover finish time:	21-Jul-16 16:17:19
primary services offline for:	
total time former primary in physical role:	+00 00:14:13
time to reach upgrade redo:	
time to recover upgrade redo:	
primary to logical switchover start time:	21-Jul-16 16:16:58
physical to primary switchover finish time:	21-Jul-16 16:34:41
primary services offline for:	+00 00:01:41

SUCCESS!The physical rolling upgrade is complete



Oracle Corporation, World Headquarters

500 Oracle Parkway

Redwood Shores, CA 94065, USA

著者：Michael Ramchand、Peter Wilson、Martien Ouwens





海外からのお問い合わせ窓口

電話：+1.650.506.7000

ファクシミリ：+1.650.506.7200

Integrated Cloud Applications & Platform Services

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

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

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

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0615

Oracle Databaseローリング・アップグレード
2016年8月



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