

Oracle Maximum  
Availability Architecture

# Oracle Data Guardの一時ロジカル・スタンバイによるTransparent Data Encryptionへの変換

## Oracle Database 11g Release 2

Oracle ホワイト・ペーパー | 2015 年 5 月



## 目次

はじめに	1
TDE の概要	2
TDE 表領域暗号化の制限	2
変換の概要	2
一時ロジカル・スタンバイの制限	3
前提条件	4
変換の例	5
Transparent Data Encryption の有効化	5
一時ロジカル・スクリプト - 最初の実行	8
TDE 表領域へのデータの移動	10
一時ロジカル・スクリプト - 2 回目の実行 (計画保守時間の開始)	16
一時ロジカル・スクリプト - 最終実行	18
結論	19
付録 A - physru.sh スクリプトの最初の実行	20
付録 B - physru.sh スクリプトの 2 回目の実行	23
付録 C - physru.sh スクリプトの 3 回目の実行	25
付録 D - physru.sh の統計情報	28
付録 E - TDE に変換するためのその他の方法	29

## はじめに

Oracle Advanced Security Transparent Data Encryption (TDE) によるデータ暗号化では、データを暗号化するプロセスが必要です。多くのアプリケーションで 24 時間 365 日の可用性への要望が高まる中、このタスクを最小限のアプリケーション停止時間で実行することは、重要な関心事となっています。

このため Oracle Maximum Availability Architecture (Oracle MAA) のベスト・プラクティスでは、Data Guard の一時ロジカル・スタンバイを使用して、変換プロセス中に本番データベースのパフォーマンスに影響が出ることを防ぎ、停止時間を最小限に抑えることを推奨しています。TDE への変換は別のデータベース (スタンバイ・データベース) で実行され、本番の動作には影響しないため、データベース・サイズに関係なく、停止時間は最小限で済みます。その他の方法については、「付録 E - TDE に変換するためのその他の方法」を参照してください。

既存のフィジカル・スタンバイ・データベースがある場合でも、TDE に変換しやすくするために別途デプロイした新しいフィジカル・スタンバイ・データベースを使用する場合でも、変換プロセスには次のような手順が含まれます。

1. Data Guard のフィジカル・スタンバイ・データベース (アーカイブ・ログ・ギャップなし) が存在すること
2. フィジカル・スタンバイからロジカル・スタンバイへの変換
3. スタンバイ適用プロセスの一時停止
4. ロジカル・スタンバイでの TDE による表領域の再構築と、TDE 構成の設定
5. ロジカル適用プロセスの開始による、(暗号化された) スタンバイとプライマリ・データベースの再同期
6. Data Guard のスイッチオーバー。ベスト・プラクティスを使用した場合、アプリケーションの停止見込み時間は 5 分未満です。
7. 古いプライマリ (一時的なロジカル・スタンバイ) の新しいフィジカル・スタンバイ・データベースへの変換
8. 新しいスタンバイ・データベース (元のプライマリ) での、Data Guard フィジカル適用プロセスの開始
9. 本番の元のプライマリへの切り替え (オプション)。ベスト・プラクティスを使用した場合、停止見込み時間は 5 分未満です。

この Oracle Maximum Availability Architecture のベスト・プラクティスのホワイト・ペーパーは、暗号化されていない Oracle Database を最小限の停止時間で TDE に変換したいデータベース管理者を対象としています。このホワイト・ペーパーは、Data Guard と TDE を技術的に理解していることが前提となります。

## TDEの概要

TDE では、Oracle データベースに保管中のデータを暗号化できます。"保管中"とは、データが保存場所のオペレーティング・システムとストレージのレベルで暗号化されるということです。TDE では、通常のデータベース認証/認可ルールが適用されるバッファ・キャッシュをヒットすると、データが透過的に復号化されます。

TDE 暗号化には 2 つの形式があります。TDE 列暗号化では、データの特定の列が暗号化されるのに対し、TDE 表領域暗号化では、TDE で暗号化された表領域内のすべてのデータが暗号化されます。表領域暗号化では、バルク暗号化によってパフォーマンスが向上します。また、管理者が各列を分析して、暗号化する列を決める必要もありません。さらに、表領域暗号化では、列暗号化と比べて制限が少なくなります。このホワイト・ペーパーでは、TDE 表領域暗号化に変換する方法について説明します。TDE 表領域暗号化は、Oracle Database 11g Release 1 (11.1) 以降で使用できます。

TDE 暗号化について詳しくは、『[Oracle Database Advanced Security 管理者ガイド](#)』を参照してください<sup>1</sup>。

### TDE表領域暗号化の制限

TDE 表領域暗号化には、制限はほとんどありません。列暗号化を使用する SQL レイヤーとは異なり、読取り/書き込み中に暗号化/復号化が行われるためです。TDE 表領域暗号化の制限は次のとおりです。

- » TDE 表領域暗号化を使用して、外部ラージ・オブジェクト (BFILE) を暗号化することはできません。これらのファイルは、データベース外に存在するためです。
- » TDE で暗号化された表領域でインポート/エクスポート操作を実行するには、Oracle Data Pump を使用します。

## 変換の概要

TDE の表領域の暗号化は、表領域の作成中にのみ有効にすることができます。既存の表領域を変更して、TDE を有効にすることはできません。Data Guard の一時ロジカル・スタンバイ・データベースを使用すると、TDE 表領域暗号化への変換中、アプリケーションのパフォーマンスと可用性に対する影響を制限できます。Oracle Data Pump によって、データが一時ロジカル・スタンバイからエクスポートされます。既存の表領域が削除され、新しい TDE 対応の表領域が作成されてからインポートが実行されます。

---

<sup>1</sup> [https://docs.oracle.com/cd/E16338\\_01/network.112/b56286/asotrans.htm#ASOAG600](https://docs.oracle.com/cd/E16338_01/network.112/b56286/asotrans.htm#ASOAG600)

ローリング・アップグレード・プロセスについて詳しくは、[Data Guard 一時ロジカル・スタンバイ・データベースに関する Oracle MAA のベスト・プラクティス<sup>2</sup>](#)を参照してください。プロセスは似ていますが、データベース・アップグレードの代わりに変換が行われる点が異なります。

### 一時ロジカル・スタンバイの制限

このプロセスではロジカル・スタンバイ・データベースを使用するため、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は次のとおりです。[ロジカル・スタンバイの前提条件と制限](#)について詳しくは、Data Guard のドキュメントを参照してください<sup>3</sup>。

- » Data Guard Broker は無効にする必要があります。
- » Data Guard の保護モードは、MAXIMUM PERFORMANCE または MAXIMUM AVAILABILITY に設定する必要があります。
- » 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。
- » ロジカル・スタンバイ・データベースは、Oracle Label Security をサポートしていません。
- » ロジカル・スタンバイ・データベースは、Oracle E-Business Suite の実装を完全にはサポートしていません。サポートされていないデータ型が含まれる表があるためです。
- » トランスポータブル表領域で、暗号化された表領域を転送することはできません。
- » トランスポータブル表領域で、暗号化された列を含む表が含まれる表領域を転送することはできません。
- » データ型の制限 (11.2) :
  - » BFILE
  - » コレクション (VARRAY とネストした表)
  - » マルチメディア・データ型 (Spatial、Image、および Oracle Text を含む)
  - » ROWID、UROWID
  - » ユーザー定義の型

注：拡張データ型サポートを使用して、データ型の制限を減らすことができます。[My Oracle Support Note 949516.1](#) (SQL Apply Extended Datatype Support - 11.2) <sup>4</sup>を参照してください。

<sup>2</sup> <http://www.oracle.com/technetwork/jp/database/maa-wp-11g-upgrades-made-easy-326438-ja.pdf>

<sup>3</sup> [https://docs.oracle.com/cd/E16338\\_01/server.112/b56302/data\\_support.htm#CHDGFADJ](https://docs.oracle.com/cd/E16338_01/server.112/b56302/data_support.htm#CHDGFADJ)

<sup>4</sup> <https://support.oracle.com/epmos/faces/DocumentDisplay?id=949516.1>

## 前提条件

このプロセスを確実に実行するには、次の前提条件が必要です。

- » 既存のフィジカル・スタンバイ・データベースが存在する必要があります。
- » COMPATIBLE を 11.1.0 以降に設定します。ただし、拡張機能を有効にするには、11.2 に設定する必要があります。
- » Oracle MAA ベスト・プラクティスでは、プライマリ・データベースで強制ロギングが有効になっている必要があります。これはレプリケーションに必要であり、スイッチオーバー中のリカバリ不可能なオブジェクトに対して保護します。次の問合せで行が返されない場合は、リカバリ不可能なブロックがないということです。

```
SQL> select NAME from V$DATAFILE where UNRECOVERABLE_CHANGE#>0;

no rows selected
```

- » プライマリとスタンバイの両方で、フラッシュバック・データベースを有効にする必要があります。次の問合せで、プライマリとスタンバイの両方で'YES'が返されるはずです。

```
SQL> select flashback_on from v$database;
```

```
FLASHBACK_ON
-----
YES
```

- » このホワイト・ペーパーで説明するプロセスによって、既存のリストア・ポイントはすべて削除されます。これが、使用しているアプリケーションの障害にならないことを確認してください。
- » ここで説明する方法は、Data Guard Brokerとは互換性がありません。このため、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、Data Guard Brokerを無効にしてパラメータ DG\_BROKER\_STARTを FALSE に設定する必要があります。
- » このプロセス中に、TDE 暗号化が指定されているすべての表領域で、Data Pump エクスポートが実行されます。SYSTEM 表領域と SYSAUX 表領域は、この処理から除外されます。これらのエクスポートの取込みに十分な領域が必要です。expdp の estimate\_only=YES オプションを使用して、エクスポート用領域の概算を把握する必要があります。

---

注：Data Pump エクスポートを圧縮する予定の場合、expdp の見積りでは圧縮は考慮されていません。圧縮によって領域は節約できますが、エクスポートとインポートにかかる時間は長くなります。

---

- » ログのアーカイブ先がプライマリ・データベースである場合、REDO を転送するには、これをデータベースごとに設定する必要があります。ブローカがいつも構成されている場合は、スタンバイからプライマリへの転送先が設定されていない可能性があるため、構成を無効にしながら、この設定を手動で行う必要があります。スタンバイからプライマリの宛先では、`valid_for(online_logfiles,primary_role)`を使用して、ロジカル・スタンバイからプライマリへの REDO 送信によって発生するエラーを防ぐ必要があります。
- » データベースごとに、`fal_server` を正しく設定する必要があります。
- » プライマリ・データベースとスタンバイ・データベースでパラメータ `STANDBY_FILE_MANAGEMENT` を `AUTO` に設定して、REDO Apply 中に新しいデータファイルが作成されるようにする必要があります。
- » 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、`DB_FILE_NAME_CONVERT` を設定する必要があります。**ファイルの上書きを防ぐため、この設定はローカル・スタンバイ・データベースでは特に重要です。**
- » 静的サービスは、各データベースの 1 つのインスタンスに必要です。これは、この後説明するスクリプトによって、データベース・ローリング保守のプロセス中にデータベースを起動できるようにするためです。静的サービスが定義されていない場合は、[My Oracle Support Note 1387859.1](#)、[Oracle Data Guard Broker and Static Service Registration<sup>5</sup>](#)を参照して設定してください。
- » 注：Data Guard Broker で静的サービスが構成されている場合は、これを使用できます。
- » スクリプトが実行されるノードの `tnsnames.ora` では、静的サービスの接続記述子を使用するネット・サービス名が必要です。

## 変換の例

### Transparent Data Encryptionの有効化

TDE では、ウォレットを使用してマスター暗号化キーを保存します。デフォルトのデータベース・ウォレットも使用できますが、`sqlnet.ora` の `ENCRYPTION_WALLET_LOCATION` パラメータを使って、TDE 固有のウォレットを使用することを強くお勧めします。また自動ログイン・ウォレットを使用すると、データベースの起動ごとに管理者がウォレットを手動で開く必要がなくなります。

ロジカル・スタンバイ・データベースでは、ウォレットの作成が制限されます。このため、このプロセスを実行する前に、プライマリにウォレットを作成してスタンバイにコピーする必要があります。

---

注：このプロセスが終了するまでは、Oracle RAC プライマリ・データベースでの暗号化にこの新しいウォレットを使用しないでください。

---

<sup>5</sup> <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1387859.1>

このウォレットは 1 つのプライマリ・インスタンスに作成して、プライマリ・データベースとスタンバイ・データベースのその他のすべてのノードに手動でコピーする必要があります。

1. 暗号化ウォレットを作成します。

プライマリとスタンバイのすべてのノードで、sqlnet.ora にウォレットの場所を設定します。

```
ENCRYPTION_WALLET_LOCATION =  
    (SOURCE = (METHOD = FILE)  
      (METHOD_DATA =  
        (DIRECTORY = /u01/app/oracle/admin/TDE/$ORACLE_SID)  
      )  
    )
```

---

注：ディレクトリ・パスで `ORACLE_SID` を使用すると、すべてのデータベースでウォレットが共有されなくなります。システムにデータベースが 1 個しかない場合、`ORACLE_SID` は不要です。

---

2. 適切な `ORACLE_SID` を使って、すべてのノードで対応するディレクトリを作成します。

```
mkdir -p /u01/app/oracle/admin/TDE/$ORACLE_SID
```

3. 新しい SQL\*Plus セッションを開始します。この操作によって、sqlnet.ora が選択されるように変更されます。

4. マスター暗号化キーを設定します。

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "AbCdeFgh!";
```

---

注：パスワード文字列は、必ず二重引用符 (" ") で囲みます。

---


5. 自動ログイン・ウォレットを作成します。

自動ログイン・ウォレットがあると、データベースの起動時にウォレットを手動で開く必要がなくなります。

```
orapki wallet create -wallet /u01/app/oracle/admin/TDE/$ORACLE_SID -  
auto_login
```

6. キーストア・ディレクトリに生成されたファイルを、プライマリとスタンバイのすべてのノードにコピーします。ファイルを各ノードにコピーします。

```
scp /u01/app/oracle/admin/TDE/$ORACLE_SID/*  
oracle@<host>:/u01/app/oracle/admin/TDE/<SID_NAME>/
```

- 
7. ウォレットがすべてのノードで開いていることを確認します。

```
SQL> select * from gv$encryption_wallet;
```

```
INST_ID    WRL_TYPE
-----
WRL_PARAMETER
-----
STATUS
-----
          1 file
/u01/app/oracle/admin/TDE/$ORACLE_SID
OPEN.
```

## 一時ロジカル・スクリプト - 最初の実行

一時ロジカル・スタンバイ・データベースによるデータベースのローリング・アップグレード用のプロセスは、さまざまなタイプの移行に適用できます。このプロセスを使用しないと停止時間が長くなります。一時ロジカル・プロセスはフィジカル・スタンバイ・データベースで開始/終了されるため、この名前が付いています。ロジカル適用プロセスは、さまざまなバージョンのデータベース（アップグレード/移行が完了していないプライマリ・データベースやアップグレード/移行が完了しているスタンバイ・データベースなど）間の、構成のレプリケーション中にのみ使用されます。Data Guard のドキュメントに記載されているとおり、一時ロジカル・プロセスは手動で実行される場合もありますが、ここで説明するベスト・プラクティスではオラクルが提供するスクリプトを使用します。このスクリプトによって、プロセス全体の多くの手動手順が自動化され、より多くの検証が自動的に実行されます。

オラクルが提供するスクリプトによる一時ロジカル・ローリング保守プロセスについて詳しくは、[Data Guard の一時ロジカル・スタンバイ・データベース用の Oracle MAA のベスト・プラクティス<sup>6</sup>](#)を参照してください。このホワイト・ペーパーで説明している physru.sh スクリプトは、[My Oracle Support Note 949322.1<sup>7</sup>](#)からダウンロードできます。


フィジカル・スタンバイ・データベースをロジカル・スタンバイに変換します。

1. 環境内のすべての前提条件が正しいことを確認します。
2. スタンバイが同期されており、遅延が秒単位であることを確認します。
3. スタンバイが Oracle RAC である場合は、1つのインスタンスを除くすべてのインスタンスを停止し、cluster\_database=false scope=spfile を設定して、1つのインスタンスをマウント済みの状態にします。1つのスタンバイ・インスタンスが実行されているノードには、静的リスナーが存在する必要があります。これが必要なのは、スクリプトで使用されるリストア・ポイントの作成にスタンバイのマウントが必要であるためです。

---

6 <http://www.oracle.com/technetwork/jp/database/maa-wp-11g-upgrades-made-easy-326438-ja.pdf>

7 <https://support.oracle.com/epmos/faces/DocumentDisplay?id=949322.1>



```
SQL> alter system set cluster_database=false scope=spfile;
```

```
System altered.
```

```
$ srvctl stop database -d standby
```

```
$ srvctl start instance -d standby -i standby1 -o mount
```

```
SQL> recover managed standby database using current logfile disconnect;
```

```
Media recovery complete.
```

4. "physru"アップグレード・スクリプトを実行して完了します。サンプル出力については、付録 A を参照してください。

```
Example: physru.sh <username> <static_primary_tns> <static_standby_tns>
```

```
<primary_name> <standby_name> <upgrade_version>
```

Arguments:

<username> = dba username

<primary\_tns> = static tns service name to primary

<standby\_tns> = static tns service name to physical standby

<primary\_name> = db\_unique\_name of primary

<standby\_name> = db\_unique\_name of standby

<upgrade\_version> = Since there is no upgrade, use current version

データベースにサポートされていないデータ型があると、次の警告が表示される場合があります。

```
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):
```

この場合は、実行ディレクトリの physru\_unsupported.log ファイルで詳細を確認し、それに基づいて解決します。

ただし、以下のようなエラーが表示される場合は、スタンバイが読取り専用で開いています。このエラーが発生したら、cluster\_database=false でスタンバイのシングル・インスタンスをマウントして、スクリプトを再実行します。

```
ERROR at line 1:
ORA-16000: database open for read-only access
ORA-06512: at line 6
### The offending sql code in its entirety:
set pagesize 0 feedback off verify off heading off echo off tab off
whenever sqlerror exit sql.sqlcode
declare
    cursor curs is
        select name from v$restore_point where name like 'PRU_%';
begin
    for r_curs in curs loop
        execute immediate 'drop restore point ' || r_curs.name;
    end loop;
end;
/
exit;
Apr 03 20:23:51 2015 [0-1] ERROR: failed to purge script state from database
standby
```

### TDE表領域へのデータの移動

この移行段階では、[データ・ポンプ・エクスポート](#)<sup>8</sup>と[インポート](#)<sup>9</sup>によって、データのアンロードと再ロードが行われます。この操作には、多くの方法があります。データの再ロードの分割方法（表領域ごと、スキーマごと、またはデータベース全体）については、検討が必要です。最高のパフォーマンスを実現するには、エクスポートとインポートに PARALLEL パラメータを使用し、インポートに EXCLUDE=statistics を使用する必要があります。この後、ロジカルがプライマリになると統計情報が収集されます。COMPRESSION を使用するとエクスポート・ファイルのディスク・ストレージ要件を減らすことができますが、エクスポートとインポートの時間が長くなる可能性があります。

---

8 [http://docs.oracle.com/cd/E16338\\_01/server.112/b56303/dp\\_export.htm#SUTIL200](http://docs.oracle.com/cd/E16338_01/server.112/b56303/dp_export.htm#SUTIL200)

9 [http://docs.oracle.com/cd/E16338\\_01/server.112/b56303/dp\\_import.htm#SUTIL300](http://docs.oracle.com/cd/E16338_01/server.112/b56303/dp_import.htm#SUTIL300)

---

注：SYSTEM と SYSAUX の表領域は TDE と互換性がないため、変換しないでください。

---

1. ロジカル適用を停止します。

```
SQL> alter database stop logical standby apply;
```

2. TDE 表領域暗号化に変換されるすべての表領域で、Data Pump エクスポートを実行します。  
DB\_BLOCK\_CHECKING=FALSE および DB\_BLOCK\_CHECKSUM=FALSE に設定してパラレル・オプションを使用すると、全体的なパフォーマンスを上げることができます。

```
$ expdp "'sys as sysdba'" compression=all parallel=4 dumpfile=TDE_%U.dmp  
logfile=TDE_exp.log tablespaces=TS1[,TS2,...]
```

この例では、エクスポート・ディレクトリとして、デフォルト・ディレクトリ DATA\_PUMP\_DIR を使用します。前提条件で示したとおり、ディレクトリに十分な領域があることを確認します。別のディレクトリを構成して使用することもできます。

3. ガード・ステータスを無効にします。

これで、索引を確実に再構築できます。この手順を実行しないと、索引のインポートが失敗します。

```
SQL> select guard_status from v$database;
```

```
GUARD_S
```

```
-----
```

```
ALL
```

```
SQL> alter database guard none;
```

```
SQL> select guard_status from v$database;
```

```
GUARD_S
```

```
-----
```

```
NONE
```

4. スクリプトによって作成された保証付きリストア・ポイントと、スクリプト実行前の既存のリストア・ポイントを、すべて削除します。

保証付きリストア・ポイントが存在すると、表領域の変更や削除を行うことができないため、必ず削除する必要があります。

まず、既存の各リストア・ポイントの scn と名前を収集します。

```
SQL> col name format a50
```

```
SQL> script STANDBY_restore_point_history.log
```

```
SQL> select name,scn from v$restore_point order by TIME;
```

```

NAME                                                    SCN
-----
PRU_0000_0001                                         2019288602
PRU_0101                                               2019288602
PRU_0201                                               2019290873
PRU_0202                                               2019293221
PRU_0203                                               2019293425
PRU_0204                                               2019308174

```

SQL> script STANDBY\_restore\_point\_history.log  
保護対策として、プライマリからもこの情報を収集します。

SQL> col name format a50

SQL> script PRIMARY\_restore\_point\_history.log

SQL> select name,scn from v\$restore\_point order by TIME;

```

NAME                                                    SCN
-----
PRU_0000_0001                                         2019288658

```

SQL> script PRIMARY\_restore\_point\_history.log

以下のブロックを使用して、スタンバイのみですべてのリストア・ポイントを削除できます。

```

set serveroutput on
declare
  cursor curs is
    select name from v$restore_point ;
begin
  for r_curs in curs loop
    execute immediate 'drop restore point ' || r_curs.name;
  end loop;
end;

```

5. エクスポートした表領域を削除します。

DBMS\_METADATA.GET\_DDL プロシージャを使用して、表領域の DDL を取得できます。

```
SQL> set long 99999
```

```
SQL> select dbms_metadata.get_ddl('TABLESPACE','TS1') from dual;
```

```
DBMS_METADATA.GET_DDL('TABLESPACE','TS1')
```

```
-----  
  
CREATE BIGFILE TABLESPACE "TS1" DATAFILE  
SIZE 3221225472  
AUTOEXTEND ON NEXT 1073741824 MAXSIZE 33554431M  
LOGGING ONLINE PERMANENT BLOCKSIZE 8192  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE DEFAULT  
NOCOMPRESS SEGMENT SPACE MANAGEMENT AUTO  
SQL> drop tablespace TS1 including contents and datafiles;
```

6. 以下のような暗号化句を使用して、エクスポートした表領域を再作成します。

注：TDE では、次のような暗号化アルゴリズムを使用できます。

- » 3DES168
- » AES128
- » AES192
- » AES256

```
SQL> CREATE BIGFILE TABLESPACE "TS1" DATAFILE  
SIZE 3221225472  
AUTOEXTEND ON NEXT 1073741824 MAXSIZE 33554431M  
LOGGING ONLINE PERMANENT BLOCKSIZE 8192  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE ENCRYPTION using 'AES256' DEFAULT  
STORAGE(ENCRYPT) SEGMENT SPACE MANAGEMENT AUTO;  
Tablespace created.
```

7. 暗号化した TDE 表領域にデータベースをインポートします。

```
$ impdp "'sys as sysdba'" PARALLEL=4 EXCLUDE=statistics  
DUMPFILE=TDE_%U.dmp LOGFILE=TDE_imp.log
```

---

Oracle RAC データベースで PARALLEL を使用するには、すべてのインスタンスからアクセスできる共有ストレージにファイルが存在する必要があります。この手順に他のインスタンスがない場合は、この要件は不要です。

---

8. ガード・ステータスを再度有効にして、ロジカル適用を開始します。

```
SQL> alter database guard all; Database altered.  
SQL> select guard_status from v$database;  
GUARD_S  
-----  
ALL  
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY immediate;  
Database altered.
```

9. 一時ロジカル・スクリプトを適切な場所で続行するには、**次の手順が非常に重要です。**

手順 4 で収集されたデータから、最初と最後のリストア・ポイントを順番に再作成します。これらのリストア・ポイントは、スクリプトの停止場所を追跡するためだけの方法です。SCN は関係ありません。関連するのは、名前と作成順序だけです。

```
SQL> create restore point PRU_0000_0001;  
Restore point created.  
SQL> create restore point PRU_0204;  
Restore point created.
```

10. スタンバイがプライマリ・データベースになる準備中に、スタンバイで Oracle RAC を再度有効にできるようになりました。

```
SQL> alter system set cluster_database=true scope=spfile;
```

```
System altered.
```

```
SQL> shutdown immediate
```

```
Database closed.
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
$ srvctl start database -d standby -o open
```

11. スクリプトで使用される静的リスナーを使って構成されているスタンバイ・インスタンスで、ロジカル適用を開始します。

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

```
Database altered.
```

12. スイッチオーバー後に REDO 送信を継続できるように、現在のプライマリ・データベースにリモート宛先が設定されていることを確認します。

```
SQL> select value from v$parameter where name='log_archive_dest_2';
```

```
VALUE
```

```
-----  
service="primary" ASYNC db_unique_name="primary"  
valid_for=(all_logfiles,primary_role)
```

13. プライマリの現在の scn と、ロジカル・スタンバイに適用されている scn を比較します。

### スタンバイ

```
SQL> select APPLIED_SCN from V$LOGSTDBY_PROGRESS;
```

```
APPLIED_SCN
```

```
-----
```

```
2019754453
```

## プライマリ

```
SQL> select current_scn from v$database;
```

```
CURRENT_SCN
```

```
-----
```

```
2019754466
```

2 つの SCN 値の差が 200 以内である場合は、続行します。詳しくは、こちらの[ロジカル・スタンバイ・データベースの管理とチューニング](#)<sup>10</sup>に関するオラクルのドキュメントを参照してください。

### 一時ロジカル・スクリプト - 2回目の実行 (計画保守時間の開始)

同じ入力内容で 2 回目の physru.sh スクリプトを実行すると、TDE が有効な一時ロジカルへのスイッチオーバーが実行され、新しいスタンバイ (元のプライマリ) がフラッシュバックされて、新しいプライマリ・データベースのフィジカル・スタンバイに変換されます。

1. 前回と同じ引数で physru.sh を実行します。サンプル出力については、付録 B を参照してください。この操作は、アクティビティが少ない計画保守時間に実行する必要があります。

```
$ ./physru.sh sys primary1_static standby1_static primary standby
11.2.0.4.0
```

2. 古いプライマリがマウント済みのシングル・インスタンスになっている間は、手順 5 が一時停止します (以前と同様に、cluster\_database をここで変更する必要はありません)。

```
WARN: primary 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 primary1.
```

```
eg: srvctl stop instance -d primary -i primary2 -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.
```

```
$ srvctl stop instance -d primary -i primary2 -o abort
```

3. 続行するには、スクリプト・プロンプトで'y'と入力します。

---

<sup>10</sup> [https://docs.oracle.com/cd/E16338\\_01/server.112/b56302/manage\\_ls.htm#SBYDB00800](https://docs.oracle.com/cd/E16338_01/server.112/b56302/manage_ls.htm#SBYDB00800)

4. 新しいスタンバイがシャットダウンされると、このスクリプトは停止します。このスクリプトをローリング・アップグレードで使用してバイナリを変更する場合は、この処理が必要です。ただし必要なのは、この時点で、続行する前にマウント済み状態の新しいスタンバイを起動する場合のみです。Oracle RAC 構成では、すべてのインスタンスを起動できます。

NOTE: Database primary 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 primary 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 primary

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

`$ srvctl start database -d primary -o mount`

5. 新しいスタンバイでリストア・ポイントを削除します。**この操作を行わないと、表領域の欠落が見つかった場合にリカバリが失敗します。**以下のブロックを使用して、新しいスタンバイのみでリストア・ポイントを削除できます。

```
set serveroutput on
declare
  cursor curs is
    select name from v$restore_point ;
begin
  for r_curs in curs loop
    execute immediate 'drop restore point ' || r_curs.name;
  end loop;
end;
```

## 一時ロジカル・スクリプト - 最終実行

最終実行で、新しいスタンバイ（元のプライマリ）と、新しいプライマリによって生成されるすべての REDO が同期されます。この処理には、一時ロジカル・データベースが TDE に移行される間に元のプライマリ・データベースで実行されたすべてのアプリケーション・アクティビティが含まれます。また、TDE への変換に直接関連するすべての REDO と、最初のスイッチオーバー後に新しいプライマリで実行されたすべてのトランザクションも含まれます。スクリプトの最後に、以前のプライマリ/スタンバイの構成に戻すオプションがあります。

1. 最後にもう一度、前回と同じ引数を使用して `physru.sh` を実行します。サンプル出力については、付録 C を参照してください。

```
./physru.sh sys primary1_static standby_static primary standby 11.2.0.4.0
```

2. アラート・ログの監視中に、次のメッセージが表示される場合があります。

```
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、新しいプライマリのリモート・ログ・アーカイブの宛先がタイムアウトしていることが原因です。このメッセージを無視しても、最終的に再接続は行われます。または、新しいプライマリで `log_archive_dest_state_n=enable` を設定して、再接続を強制的に実行することもできます。

3. (MRP の開始前に消去が必要な) オンライン REDO ログのサイズ、適用が必要な REDO の量、およびシステムの速度によっては、スクリプトが 10 分後にタイムアウトする場合があります。

```
Apr 02 14:07:17 2015 [6-1] ERROR: timed out after 10 minutes of inactivity
```

このようなタイムアウトが発生したら、同じ引数でスクリプトを再開します。

4. スクリプトの最後に、元の構成に戻すかどうかを選択するためのプロンプトが表示されます。ここで 'y' と入力すれば、すぐに元の構成に戻すことができます。または 'n' と入力して、後で必要に応じて戻すこともできます。

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

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

5. スイッチオーバーを選択した場合は、スクリプトで指示されるシングル・インスタンスに、新しいプライマリ・データベースを移行する必要があります。

```
WARN: standby 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 standby1.  
eg: srvctl stop instance -d standby -i standby2
```

```
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):
```

```
$ srvctl stop instance -d standby -i standby2
```

6. 'y'を入力してスクリプトを続行します。
7. プライマリの最初のインスタンスのみがオープン状態になっています。スクリプトの指示に従って、マウント済みのインスタンスをオープンします。
8. スタンバイでは、1つのインスタンスのみがマウントされています。スタンバイを任意の状態（Active Data Guard またはマウント済み）に再起動し、管理リカバリを開始します。
9. スクリプトを実行すると、プロセスに関する統計情報が生成されます。この例については、付録 D を参照してください。
10. 再ロードされたオブジェクトに関する統計情報を収集します。この操作は、新しい以前のプライマリで変換からの REDO が適用されている間、または逆になったロールを一定期間使用できる場合にも実行できます。

## 結論


Oracle Advanced Security Transparent Data Encryption に変換することで、Oracle Database に保管中のデータを保護できます。多くのアプリケーションにとって、24 時間 365 日の要件に対応できるデータ暗号化プロセスが課題となっています。一時ロジカル・スタンバイ・データベースを使用することで、アプリケーションの可用性への影響を最小限に減らしながら、保管中のデータを暗号化するという目標を達成できます。

## 付録A - physru.shスクリプトの最初の実行

```
oracle@slcc32adm05 andy]$ ./physru.sh sys primary1_static standby1_static
primary standby 11.2.0.4.0
Please enter the sysdba password:
```

```
### Initialize script to either start over or resume execution
Apr 04 08:22:58 2015 [0-1] Identifying rdbms software version
Apr 04 08:22:58 2015 [0-1] database primary is at version 11.2.0.4.0
Apr 04 08:22:58 2015 [0-1] database standby is at version 11.2.0.4.0
Apr 04 08:22:58 2015 [0-1] verifying flashback database is enabled at
primary and standby
Apr 04 08:22:59 2015 [0-1] verifying available flashback restore points
Apr 04 08:22:59 2015 [0-1] verifying DG Broker is disabled
Apr 04 08:22:59 2015 [0-1] looking up prior execution history
Apr 04 08:22:59 2015 [0-1] purging script execution state from database
primary
Apr 04 08:22:59 2015 [0-1] purging script execution state from database
standby
Apr 04 08:22:59 2015 [0-1] starting new execution of script

### Stage 1: Backup user environment in case rolling upgrade is aborted
Apr 04 08:22:59 2015 [1-1] creating restore point PRU_0000_0001 on
database standby
Apr 04 08:23:00 2015 [1-1] backing up current control file on standby
Apr 04 08:23:01 2015 [1-1] created backup control file
/u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/PRU_0001_standby_f.f
Apr 04 08:23:01 2015 [1-1] creating restore point PRU_0000_0001 on
database primary
Apr 04 08:23:03 2015 [1-1] backing up current control file on primary
Apr 04 08:23:06 2015 [1-1] created backup control file
/u01/app/oracle/product/11.2.0.4/dbhome_1/dbs/PRU_0001_primary_f.f
```



NOTE: Restore point PRU\_0000\_0001 and backup control file PRU\_0001\_standby\_f.f can be used to restore standby 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 Apr 04 08:23:06 2015 [2-1] verifying RAC is disabled at standby
Apr 04 08:23:06 2015 [2-1] verifying database roles
Apr 04 08:23:06 2015 [2-1] verifying physical standby is mounted
Apr 04 08:23:07 2015 [2-1] verifying database protection mode
Apr 04 08:23:07 2015 [2-1] verifying transient logical standby datatype
support
```

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 metalink 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): y

```
Apr 04 08:23:11 2015 [2-1] continuing
Apr 04 08:23:11 2015 [2-2] starting media recovery on standby
Apr 04 08:24:30 2015 [2-2] confirming media recovery is running
Apr 04 08:24:30 2015 [2-2] waiting for apply lag to fall under 30 seconds
Apr 04 08:24:37 2015 [2-2] apply lag measured at 7 seconds
Apr 04 08:24:37 2015 [2-2] stopping media recovery on standby
Apr 04 08:24:38 2015 [2-2] executing dbms_logstdby.build on database
primary
Apr 04 08:25:02 2015 [2-2] converting physical standby into transient
logical standby
Apr 04 08:26:23 2015 [2-3] opening database standby
Apr 04 08:26:25 2015 [2-4] configuring transient logical standby
parameters for rolling upgrade
Apr 04 08:26:25 2015 [2-4] starting logical standby on database standby
```



```
Apr 04 08:26:33 2015 [2-4] waiting until logminer dictionary has fully
loaded
Apr 04 08:27:34 2015 [2-4] dictionary load 75% complete
Apr 04 08:27:44 2015 [2-4] dictionary load is complete
Apr 04 08:27:45 2015 [2-4] waiting for apply lag to fall under 30 seconds
Apr 04 08:27:51 2015 [2-4] apply lag measured at 6 seconds
```

NOTE: Database standby 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: If standby was previously a RAC database that was disabled, it 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 standby1, set the cluster\_database parameter to TRUE.

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

- 2) Shutdown instance standby1.

```
eg: SQL> shutdown abort;
```

- 3) Startup and open all instances for database standby.

```
eg: srvctl start database -d standby
```

## 付録B - physru.shスクリプトの2回目の実行


```
[oracle@slcc32adm05 andy]$ ./physru.sh sys primary1_static standby1_static  
primary standby 11.2.0.4.0
```

```
Please enter the sysdba password:
```

```
### Initialize script to either start over or resume execution Apr 04  
08:53:45 2015 [0-1] Identifying rdbms software version  
Apr 04 08:53:45 2015 [0-1] database primary is at version 11.2.0.4.0  
Apr 04 08:53:45 2015 [0-1] database standby is at version 11.2.0.4.0  
Apr 04 08:53:46 2015 [0-1] verifying flashback database is enabled at  
primary and standby  
Apr 04 08:53:46 2015 [0-1] verifying available flashback restore points  
Apr 04 08:53:46 2015 [0-1] verifying DG Broker is disabled  
Apr 04 08:53:46 2015 [0-1] looking up prior execution history  
Apr 04 08:53:46 2015 [0-1] last completed stage [2-4] using script version  
0001  
Apr 04 08:53:46 2015 [0-1] resuming execution of script
```

```
### Stage 3: Validate upgraded transient logical standby  
Apr 04 08:53:47 2015 [3-1] database standby is no longer in OPEN MIGRATE  
mode Apr 04 08:53:47 2015 [3-1] database standby is at version 11.2.0.4.0
```

```
### Stage 4: Switch the transient logical standby to be the new primary  
Apr 04 08:53:49 2015 [4-1] waiting for standby to catch up (this could  
take a while) Apr 04 08:53:49 2015 [4-1] waiting for apply lag to fall  
under 30 seconds  
Apr 04 08:53:53 2015 [4-1] apply lag measured at 4 seconds  
Apr 04 08:53:53 2015 [4-2] switching primary to become a logical standby  
Apr 04 08:54:23 2015 [4-2] primary is now a logical standby  
Apr 04 08:54:23 2015 [4-3] waiting for standby standby to process end-of-  
redo from primary
```



```
Apr 04 08:54:23 2015 [4-4] switching standby to become the new primary
Apr 04 08:54:36 2015 [4-4] standby is now the new primary
```

```
### Stage 5: Flashback former primary to pre-upgrade restore point and
convert to physical
```

```
Apr 04 08:54:37 2015 [5-1] verifying instance primary1 is the only active
instance
```

```
WARN: primary 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 primary1.
```

```
eg: srvctl stop instance -d primary -i primary2 -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): y
```

```
Apr 04 08:58:45 2015 [5-1] continuing
```

```
Apr 04 08:58:45 2015 [5-1] verifying instance primary1 is the only active
instance
```

```
Apr 04 08:58:45 2015 [5-1] shutting down database primary
```

```
Apr 04 08:59:07 2015 [5-1] mounting database primary
```

```
Apr 04 08:59:20 2015 [5-2] flashing back database primary to restore point
PRU_0000_0001
```

```
Apr 04 08:59:21 2015 [5-3] converting primary into physical standby
```

```
Apr 04 08:59:22 2015 [5-4] shutting down database primary
```

```
NOTE: Database primary 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 primary 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 primary
```

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

## 付録C - physru.shスクリプトの3回目の実行


```
[oracle@slcc32adm05 andy]$ ./physru.sh sys primary1_static standby1_static
primary standby 11.2.0.4.0
Please enter the sysdba password:

### Initialize script to either start over or resume execution Apr 04
09:08:12 2015 [0-1] Identifying rdbms software version
Apr 04 09:08:12 2015 [0-1] database primary is at version 11.2.0.4.0
Apr 04 09:08:12 2015 [0-1] database standby is at version 11.2.0.4.0
Apr 04 09:08:12 2015 [0-1] verifying flashback database is enabled at
primary and standby
Apr 04 09:08:13 2015 [0-1] verifying available flashback restore points
Apr 04 09:08:13 2015 [0-1] verifying DG Broker is disabled
Apr 04 09:08:13 2015 [0-1] looking up prior execution history
Apr 04 09:08:13 2015 [0-1] last completed stage [5-4] using script version 0001
Apr 04 09:08:13 2015 [0-1] resuming execution of script

### Stage 6: Run media recovery through upgrade redo
Apr 04 09:08:14 2015 [6-1] upgrade redo region identified as scn range
[1893786448,
1893791142]
Apr 04 09:08:14 2015 [6-1] starting media recovery on primary
Apr 04 09:08:21 2015 [6-1] confirming media recovery is running
Apr 04 09:08:21 2015 [6-1] waiting for media recovery to initialize
v$recovery_progress
Apr 04 09:09:34 2015 [6-1] monitoring media recovery's progress
Apr 04 09:13:21 2015 [6-2] last applied scn 1893763258 is approaching
upgrade redo start scn 1893786448
Apr 04 09:13:37 2015 [6-3] recovery of upgrade redo at 59% - estimated
complete at Apr 04 09:13:55
Apr 04 09:13:52 2015 [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 primary back to a primary database and standby back to
a physical standby database. If you answer 'n' to the question below,
primary will remain a physical standby database and standby will
remain a primary database.
Do you want to perform a switchover? (y/n): y
```



```
Apr 04 09:14:26 2015 [7-1] continuing
Apr 04 09:14:27 2015 [7-2] verifying instance standby1 is the only active
instance
```

```
WARN: standby 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 standby1.  
eg: `srvctl stop instance -d standby -i standby2`

```
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): y
```

```
Apr 04 09:18:59 2015 [7-2] continuing
Apr 04 09:18:59 2015 [7-2] verifying instance standby1 is the only active
instance
Apr 04 09:19:00 2015 [7-2] waiting for apply lag to fall under 30 seconds
Apr 04 09:19:01 2015 [7-2] apply lag measured at 1 seconds
Apr 04 09:19:02 2015 [7-3] switching standby to become a physical standby
Apr 04 09:19:06 2015 [7-3] standby is now a physical standby
Apr 04 09:19:06 2015 [7-3] shutting down database standby
Apr 04 09:19:06 2015 [7-3] mounting database standby
Apr 04 09:19:17 2015 [7-4] waiting for standby primary to process end-of-
redo from primary
Apr 04 09:19:18 2015 [7-5] switching primary to become the new primary
Apr 04 09:19:23 2015 [7-5] primary is now the new primary
Apr 04 09:19:23 2015 [7-5] opening database primary
Apr 04 09:19:26 2015 [7-6] starting media recovery on standby
Apr 04 09:20:44 2015 [7-6] confirming media recovery is running
```

```
NOTE: Database primary has completed the switchover to the primary role,
but instance primary1 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 primary1.  
eg: `srvctl stop instance -d primary -i primary2`

2) Startup and open all inactive instances for database primary.  
eg: `srvctl start database -d primary`

NOTE: Database standby 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 standby  
eg: `srvctl start database -d standby -o mount`

## 付録D - physru.shの統計情報

### Stage 8: Statistics

```
script start time:                04-Apr-15 08:42:28
script finish time:               04-Apr-15 09:21:27
total script execution time:      +00 00:38:59
wait time for user upgrade:       +00 00:11:14
active script execution time:     +00 00:27:45
transient logical creation start time:
transient logical creation finish time:
primary to logical switchover start time:    04-Apr-15 08:53:53
logical to primary switchover finish time:  04-Apr-15 08:54:37
primary services offline for:              +00 00:00:44
total time former primary in physical
role:                                       +00 00:14:55
time to reach upgrade redo:                +00 00:04:03
time to recover upgrade redo:              +00 00:00:15
primary to physical switchover start time:  04-Apr-15 09:14:26
physical to primary switchover finish
time:                                       04-Apr-15 09:19:26
primary services offline for:              +00 00:05:00
```

SUCCESS: The physical rolling upgrade is complete.

## 付録E – TDEに変換するためのその他の方法

TDE 暗号化に変換するには、このホワイト・ペーパー以外の方法もあります。使用できるデータベース、ストレージのサイズや、許容可能な停止時間によっては、次のアプローチも検討できます。ただし、社内での開発手順が必要です。

- » アクティブなプライマリ・データベースで DBMS\_REDEFINITION を使用する。これは、スタンバイ・データベースを使用しないデータベースや、DBMS\_REDEFINITION プロセスに精通した管理者向けのオプションです。おもな利点は、停止時間がゼロ、または非常に少なくなることで、す。ただし、ターゲット・オブジェクトによって運用コストが変動する可能性があります。
- » このプロセスで、Data Pump の代わりに ALTER TABLE MOVE を使用する。適切なサイズの暗号化された表領域を作成して、各セグメントを暗号化された表領域に移動します。この方法は、データベース・サイズが増してもストレージが制限要素にならない場合に適しています。
- » 停止する - このホワイト・ペーパーのプロセスの目的は、移行中のアプリケーションの停止時間とパフォーマンスへの影響を最小限に減らすことです。ただし、これらが問題にならない場合は、保守期間中にプライマリ・データベースを変更できます。この変換の実行に最適なアプローチは、Data Pump を使用することです。



**ORACLE®**

CONNECT WITH US



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



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



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



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

Oracle Corporation, World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065, USA

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

### Hardware and Software, Engineered to Work Together

Copyright © 2015, 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、記載内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社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。本文書はオラクル社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。0515



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