

Oracle Maximum
Availability Architecture

Active Data Guard (DBMS_ROLLING) による Transparent Data Encryption への変換

Oracle Database 12c

Oracle ホワイト・ペーパー | 2015 年 5 月





目次

概要.....	1
TDE の概要	2
TDE 表領域暗号化の制限.....	2
変換の概要	2
DBMS_ROLLING/ロジカル・スタンバイの制限	3
前提条件	4
変換のサンプル	5
Transparent Data Encryption の有効化	5
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.....	7
TDE 表領域へのデータの移動.....	9
スイッチオーバー - ロジカル・スタンバイからプライマリへ.....	12
結論.....	15
付録 A – TDE に変換するためのその他の方法.....	16

概要

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。

Oracle Maximum Availability Architecture (Oracle MAA) のベスト・プラクティスでは、Oracle Active Data Guard スタンバイ・データベースと [DBMS_ROLLING PL/SQL パッケージ](#)¹を使用して、TDE への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ないようにす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、本番データベース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。別の方法については、「付録 A - 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. Active 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. DBMS_ROLLING PL/SQL パッケージを使用した、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. TDE による表領域の再構築と、ロジカル・スタンバイでの TDE 構成の設定。
5. 論理適用プロセスの開始による、(暗号化済み) スタンバイとプライマリ・データベースの再同期。
6. Oracle 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、Active Data Guard 物理適用プロセスの開始。
9. 本番データベースの元のプライマリへのスイッチバック (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おり、Active Data Guard と TDE について技術的に理解していることが前提となります。

¹ http://docs.oracle.com/cd/E57425_01/121/SBYDB/dbms_rolling_upgrades.htm#CJACBBBC

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ているということです。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。

TDE表領域暗号化の制限

TDE 表領域暗号化に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、読み取り/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

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。表領域の暗号化は、表領域の作成中のみ有効にすることができます。Oracle MAA のベスト・プラクティスでは、Active Data Guard スタンバイ・データベースを使用して、表領域を TDE に変換する間に、プライマリ・データベースのパフォーマンスに影響が出ないようにすることを推奨しています。TDE への移行は、DBMS_ROLLING PL/SQL パッケージを使用して、フィジカル・スタンバイ・データベースを一時ロジカル・スタンバイに一時的に変換することで開始されます。管理者は次に、Oracle Data Pump を使用してデータをエクスポートし、既存の表領域をドロップしてから、インポートを使用して新しい TDE 対応の表領域を作成します。これらの操作が完了すると、Active Data Guard によって、データの暗号化中にプライマリで発生したすべてのトランザクションが、スタンバイとの間で自動的に再同期されます。これらの処理はすべて、プライマリ・データベースで実行中の本番環境に影響を与えることはありません。アプリケーションが停止するのは、本番ユーザーから本番データベースの新しい暗号化コピーに切り替える時間だけです。

Active Data Guard の DBMS_ROLLING PL/SQL パッケージを使用すると、次の操作を自動化できます。

- » 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。
- » ロジカル・スタンバイが TDE に変換された後の再同期。
- » 新しいプライマリ・データベースとするための、TDE で暗号化されたロジカル・スタンバイへの本番データベースのスイッチオーバー。
- » 元のプライマリから新しいフィジカル・スタンバイへの変換、およびその TDE への変換。
- » 新しいスタンバイと新しいプライマリ・データベースの再同期。
- » 本番から元のプライマリ・データベースへのスイッチバック。

注：DBMS_ROLLING には、Active Data Guard のライセンスが必要です。DBMS_ROLLING を使用すると、一時ロジカル・スタンバイ・データベースを使って、ローリング方式でデータベースの保守やアップグレードを行うことが非常に簡単になります。

DBMS_ROLLING/ロジカル・スタンバイの制限

DBMS_ROLLING 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³。

- » 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。このような表を Data Guard のロール移行後にレプリケートすれば、この制約を回避できます。
- » 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 (12.1) :
 - » BFILE
 - » ROWID、UROWID
 - » コレクション (VARRAY とネストした表)
 - » ネストした表と REF が含まれるオブジェクト

³ http://docs.oracle.com/cd/E57425_01/121/SBYDB/data_support.htm#CHDGFADJ

» 次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

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。[Oracle Database 12の拡張データ型サポートについて詳しくは](#)、オラクルのドキュメントを参照してください⁴。

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
```

このプロセスによって、既存のリストア・ポイントはすべてドロップされます。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。

⁴ http://docs.oracle.com/cd/E57425_01/121//SBYDB/manage_1s.htm#BEIJDJAC

- » 説明する方法は、Data Guard Brokerとは互換性がありません。Brokerは、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無効に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。常にブローカが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、構成を無効にしてから、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 REDO 送信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。**

変換のサンプル

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。

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され、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。

- 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
```

- 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。
- パスワード・ベースのキーストアを作成します。

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
'/u01/app/oracle/admin/TDE/<ORACLE_SID>' 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。

- ウォレットを開きます。

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "AbCdEfGh!";
```

- 暗号化キーを設定します。

```
ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "AbCdEfGh!" WITH BACKUP
USING 'TDE';
```

- 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。

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE
'/u01/app/oracle/admin/TDE/$ORACLE_SID' IDENTIFIED BY "AbCdEfGh!";
```

- 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>/
```

- 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/primary1
OPEN.
```

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

プロセスの一時ロジカル部分では、DBMS_ROLLING パッケージを使用します。DBMS_ROLLING の操作は再開可能であるため、エラーが発生した場合は、問題を修正してから手順を再実行します。

- DBMS_ROLLING で計画を初期化します。

```
SQL> exec DBMS_ROLLING.init_plan('standby');
PL/SQL procedure successfully completed.
```

- スタンバイを単一のインスタンスでマウントし、DBMS_ROLLING に必要なリカバリを再開します。

```
$ srvctl stop database -d standby
$ srvctl start instance -d standby -i standby1
SQL> recover managed standby database disconnect;
```

- DBMS_ROLLING 計画を構築します。

```
SQL> exec dbms_rolling.build_plan;
PL/SQL procedure successfully completed.
この計画は、次の問合せで表示できます。
```

```
SQL> col instid format 999
```

```
SQL> col target format a10
```

```
SQL> col phase format a10
SQL> col description format a65
SQL> set lines 99
SQL> set pages 999
SQL> SELECT instid, target, phase, description FROM DBA_ROLLING_PLAN;
```

4. DBMS_ROLLING 計画を開始します（これにより、フィジカル・スタンバイがロジカルに変換されます）。

```
SQL> exec dbms_rolling.start_plan;
```

5. 以下の問合せを実行して、スタンバイがロジカルであり、プライマリから REDO を適用していることを確認し、APPLIED_SCNが増加していることを確認します。

```
SQL> select database_role, open_mode from v$databases;
DATABASE_ROLE          OPEN_MODE
-----
LOGICAL STANDBY       READ WRITE
SQL> SELECT APPLIED_SCN FROM V$LOGSTDBY_PROGRESS;
```

注：REDO が適用されておらず、ロジカル・スタンバイのアラート・ログに以下のようなエラーが表示されている場合は、バグ 20889894 が発生している可能性があります。

```
ORA-20000:Unable to gather statistics concurrently:Resource Manager is not
enabled.
```

回避策としては、次のコマンドを実行して論理適用を再開します。

```
SQL> exec dbms_stats.set_global_prefs('CONCURRENT', 'FALSE');
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY immediate;
```

TDE表領域へのデータの移動

1. 論理適用を停止します。

```
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。

```
$ expdp "'sys as sysdba'" compression=all dumpfile=TDE.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 を使用しています。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. DBMS_ROLLING パッケージで作成されたすべての保証付きリストア・ポイントをドロップします。

保証付きリストア・ポイントが存在すると、表領域の変更やドロップを行うことができないため、必ずドロップ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
```

```
-----  
DBMSRU_INITIAL 197267580
```

```
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;
```



```
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'" DUMPFILE=TDE.dmp LOGFILE=TDE_imp.log
```

8. ガード・ステータスを再度有効にして、論理適用を開始します。

```
SQL> alter database guard all;
Database altered.
SQL> select guard_status from v$databases;

GUARD_S
-----
ALL
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY immediate;
Database altered.
```

9. 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、必要に応じてプライマリをポイントするように設定します。

```
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)
```

10. 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$databases;
```

```
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。

注：REDO が適用されておらず、ロジカル・スタンバイのアラート・ログに以下のようなエラーが表示されている場合は、バグ20889894が発生している可能性があります。

```
ORA-20000:Unable to gather statistics concurrently:Resource Manager is not
enabled.
```

回避策としては、次のコマンドを実行して論理適用を再開します。

```
SQL> exec dbms_stats.set_global_prefs('CONCURRENT', 'FALSE');
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY immediate;
```

スイッチオーバー - ロジカル・スタンバイからプライマリへ

1. 現在のプライマリからDBMS_ROLLING.switchover プロシージャを実行して、ロジカル・スタンバイをプライマリ・データベースに変換します。

```
SQL> exec dbms_rolling.switchover;
```

2. 古いロジカルがプライマリ・データベースになっており、読取り/書込みでオープンされていることを確認します。

```
SQL> select database_role,open_mode from v$database;
DATABASE_ROLE      OPEN_MODE
-----
PRIMARY            READ WRITE
```

3. この時点で、新しいプライマリ・インスタンスが1つだけ起動されています。別のインスタンスも起動します。

4. フィジカル・スタンバイに変換するために、新しいスタンバイ（元のプライマリ）を単一のマウント済みインスタンスとして再起動します。

```
$ srvctl stop database -d primary
```

```
$ srvctl start instance -d primary -i primary1 -o mount
```

5. 新しいプライマリでDBMS_ROLLING計画の最終手順を実行します。これで、古いプライマリのフラッシュ・バック、フィジカル・スタンバイへの変換、新しいプライマリへのREDO適用が行われます。リカバリ時にドロップした表領域コマンドのREDO変更が検出されると、このコマンドは失敗します。この手順の実行中に、古いプライマリのアラート・ログを監視すると役立ちます。

DBMS_ROLLING で作成される保証付きリストア・ポイントが存在するために、暗号化されていない古い表領域のドロップがリカバリ時に検出されると、FINISH_PLANは最終的に失敗します。

```
SQL> exec dbms_rolling.finish_plan;
BEGIN dbms_rolling.finish_plan; END;
*
ERROR at line 1:
ORA-45415: instruction execution failure
ORA-06512: at "SYS.DBMS_ROLLING", line 36
ORA-06512: at line 1
```

古いプライマリのアラート・ログに、次のメッセージが表示されます。

```
Thu Apr 09 11:34:15 2015
Errors in file
/u01/app/oracle/diag/rdbms/primary/primary1/trace/primary1_pr00_98452.trc:
ORA-38882: Cannot drop tablespace TS1 on standby database due to
guaranteed restore points.
Thu Apr 09 11:34:15 2015
Managed Standby Recovery not using Real Time Apply
Thu Apr 09 11:34:15 2015
Recovery interrupted!
```

次の問合せを実行すると、新しいプライマリで障害の原因を確認できます。

DBA_ROLLING_EVENTSによって、計画の手順の完了とエラー・メッセージが追跡されます。

```
SQL> select event_time,MESSAGE from dba_rolling_events where TYPE='ERROR';
EVENT_TIME
-----
MESSAGE
-----
```

```
09-APR-15 11.34.37.120484 AM
failed with status 45426 in instruction 62
09-APR-15 11.34.37.122758 AM
failure while executing one or more instructions from batch 44
09-APR-15 11.34.37.129782 AM
DBMS_ROLLING.FINISH_PLAN halted due to error
```

失敗した命令番号を見つけて、リカバリが追いつくまでその命令が待機していたことを確認します。

DBA_ROLLING_PLAN に、計画の手順が表示されます。

```
SQL> SELECT instid, target, phase, description FROM DBA_ROLLING_PLAN where
instid=62;
```

INSTID	TARGET	PHASE	DESCRIPTION
62	primary	FINISH	Wait until upgrade redo has been fully recovered

6. DBMS_ROLLING で作成されたリストア・ポイントは古いプライマリのフラッシュ・バックに使用されます。そのためドロップすることができ古いプライマリでリカバリを再開します。

```
SQL> drop restore point DBMSRU_INITIAL;
Restore point dropped.
SQL> recover managed standby database disconnect;
Media recovery complete.
```

7. DBMS_ROLLING.finish_plan を再開します。これは中断した場所から開始されます。

```
SQL> exec dbms_rolling.finish_plan;
```

8. 完了すると、古いプライマリのすべてのインスタンスを起動できます。

9. この時点で、元の構成にスイッチバックできることが望ましいです。

```
SQL> alter database switchover to primary;
```

元のプライマリを再起動します。

```
$ srvctl stop database -d primary
$ srvctl start database -d primary
```



10.元のスタンバイを再起動して、リカバリを開始します（または、ブローカ構成を置き換えます）。

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

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。スタンバイ・データベースを使って暗号化プロセスを簡単にすることで、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、保管中の暗号化データの目標を達成できます。Active Data Guard 12cにはDBMS_ROLLING PL/SQL パッケージを使った新しい自動化機能があり、このためにフィジカル・スタンバイ・データベースと一時ロジカル・ローリングの保守プロセスを非常に簡単に使用できます。

付録A – 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。




CONNECT WITH US



blogs.oracle.com/oracle



facebook.com/oracle



twitter.com/oracle



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