Transparent Data Encryption for PostgreSQL Enterprise Edition
列単位暗号化 セットアップカード

（Linux 版）

ＯＳＳＤＢＴＤＥ０１－０８

ご注意

1. 本書の内容の一部または全部を無断転載することは、禁止されています。
2. 本書の内容に関しては将来予告なしに変更することがあります。
3. 本書の内容について万全を期して作成いたしましたが、万一ご不審な点や誤り、記載漏れなど、お気付きのことがありましたらご連絡ください。

輸出する際の注意事項

本製品（ソフトウェア）は、外国為替管理令に定める提供を規制される技術に該当致しますので、日本国外へ持ち出す際には日本国政府の役務取引許可申請等必要な手続きをお取りください。

許可手続き等にあたり特別な資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

はしがき

このたびは、Transparent Data Encryption for PostgreSQL Enterprise Edition をお買い上げいただき、誠にありがとうございます。

本書は、Transparent Data Encryption for PostgreSQL を使用した透過的暗号化機能の導入を行うエンジニアを対象読者とし、Transparent Data Encryption for PostgreSQL のインストール、アップグレード、アンインストールの手順について説明します。なお、透過的暗号化機能をご使用の際は、さらに『列単位暗号化 透過的暗号化機能利用の手引』をご確認ください。

重要

本手順書に記載された方法以外でインストールおよびアンインストールを行った場合は、動作の保証はいたしません。

備考

1. 本書に説明しているすべての機能はプログラムプロダクトであり、次のプロダクト型番に対応しています。

プロダクト型番	プロダクト名	対応モデル
UL4027-H201-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 (1CPU)(1 年間)	64 ビット
UL4027-H231-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 1CPU 追加(1 年間)	64 ビット
UL4027-H203-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 Cluster Option(1 年間)	64 ビット
UL4027-H211-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 (1CPU)(3 年間)	64 ビット
UL4027-H212-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 1CPU 追加(3 年間)	64 ビット
UL4027-H213-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 Cluster Option(3 年間)	64 ビット
UL4027-J201-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 (1CPU)(1 年間)(時間延長保守)	64 ビット
UL4027-J231-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 1CPU 追加(1 年間)(時間延長保守)	64 ビット
UL4027-J203-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 Cluster Option(1 年間)(時間延長保守)	64 ビット
UL4027-J211-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 (1CPU)(3 年間)(時間延長保守)	64 ビット
UL4027-J212-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 1CPU 追加(3 年間)(時間延長保守)	64 ビット
UL4027-J213-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.1 Linux 版 Cluster Option(3 年間)(時間延長保守)	64 ビット

2. Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標または商標です。
3. Red Hat、Red Hat Enterprise Linux は、米国 Red Hat, Inc.の登録商標です。

4. Oracle、Oracle Linux は、Oracle Corporation およびその子会社，関連会社の米国およびその他の国における登録商標または商標です。
5. Amazon Web Services およびすべての AWS 関連の商標、ならびにその他の AWS のグラフィック、ロゴ、ページヘッダーボタンアイコン、スクリプト、サービス名は、米国および／またはその他の国における、AWS の商標、登録商標またはトレードドレスです。
6. その他、記載されている会社名および製品名は、一般的にそれぞれ各社の商標または登録商標です。

本書の表記規則

本書では、注意すべき事項、重要な事項および関連情報を以下のように表記します。

注

この表記は、重要であるがデータ損失やシステムおよび機器の損傷には関連しない情報を表します。

重要

この表記は、データ損失やシステムおよび機器の損傷を回避するために必要な情報を表します。

ヒント

この表記は、お客様に役立つ可能性のある情報を表します。

実行例およびファイルの設定例は以下のように表記します

```
コマンドラインの実行例を示します
```

```
ファイルの設定例を示します
```

また、本書では以下の表記法を使用します。

表記	使用方法	例
コマンドライン中の [ ] 角かっこ	かっこ内の値の指定が省略可能であることを示します	cipher_setup.sh [-s {1|2} [path] | -h］
コマンドライン中の { } 波かっこ	かっこ内の値のいずれかを指定する必要があることを示します	cipher_setup.sh [-s {1|2} [path] | -h] 上記例の場合角かっこ内に波かっこがあるため、"-s" オプションを指定した場合、 "1" または "2" を指定する必要があります
#	OS の管理者ユーザで発行するコマンドを示すプロンプトです	# `./cipher_setup.sh`
$	OS の一般ユーザ（postgres など）で発行するコマンドを示すプロンプトです	$ `psql`
=#	PostgreSQL のスーパーユーザーで SQL を発行する場合は、「=#」のように表記しますが、明示的に接続しているデータベース名を示す場合は、「postgres=#」や「testdb=#」のように先頭にデータベース名を含みます	=# SELECT count(*) FROM public.cipher_key_table;
=>	PostgreSQL の一般ユーザーで SQL を発行する場合は、「=>」のように表記しますが、明示的に接続しているデータベース名を示す場合は、「postgres=>」や「testdb=>」のように先頭にデータベース名を含みます	=> SELECT c1 FROM t1;
CMD>	Windows のコマンドプロンプトで発行するコマンドを示します	CMD>`ipconfig`
モノスペースフォント斜体	ユーザーが有効な値に置き換えて入力する項目	tde_for_pg<*PostgreSQL メジャーバージョン*>-<*Transparent Data Encryption for PostgreSQL バージョン*>-.<*Red Hat Enterprise Linux バージョン*>.x86_64.rpm

最新情報の入手先

最新の製品情報については、以下の Web サイトを参照してください。

https://jpn.nec.com/tdeforpg/

目次

第 1 章
はじめに

本章では、Transparent Data Encryption for PostgreSQL の紹介と Edition ごとの提供機能やサービスについて説明します。

1.1　Transparent Data Encryption for PostgreSQL とは

Transparent Data Encryption for PostgreSQL を使用することで、表に格納する機密データを暗号化できます。また、暗号化されたデータを処理するアプリケーションは、ほとんどあるいはまったく変更せずに透過的にデータを暗号化、復号することができます。さらに、暗号鍵の管理を簡単に行う機能も提供するサブスクリプション製品です。

1.2　Edition ごとの利用可能な機能と提供されるサービス

Transparent Data Encryption for PostgreSQL には、商用版の Enterprise Edition と OSS として公開している Free Edition があります。 各 Edition で利用可能な機能と提供されるサービスを示します。

表 1-1　Edition による機能/サービスの違い

機能/サービス		Enterprise Edition for Linux	Enterprise Edition for Windows	Free Edition
Transparent Data Encryption 機能				
列単位の暗号化機能	テキスト	○	○	○
	バイト列（画像など）	○	○	○
	NUMERIC	○	○	×
	整数型（smallint,integer,bigint）	○	○	×
	日付・時刻	○	○	×
鍵の更新、バージョン管理機能		○	○	△*1
AWS Key Management Service を利用した鍵管理		○	×	×
簡易 TDE モード		○	○	×
サポートサービス				
Transparent Data Encryption for PostgreSQL の PP サポートサービス		○	○	×
PostgreSQL 本体の保守サポートサービス		○	○	×

*1　暗号鍵のバージョン管理機能なし。一括更新のみ可

第 2 章
インストールの概要

本章では、Transparent Data Encryption for PostgreSQL のインストール、アップグレード、アンインストールの概要について説明します。

2.1　インストールの種類

本書で説明する Transparent Data Encryption for PostgreSQL のインストールの種類は以下の 3 つがあります。

- 新規インストール

  Transparent Data Encryption for PostgreSQL を初めてインストールする場合に行います。

- 再インストール

  Transparent Data Encryption for PostgreSQL が既にインストールされている環境で必要なファイルが破損した場合や、オリジナルの設定ファイルをインストールしたい場合に行います。

- ストリーミングレプリケーション構成への新規セットアップ

  PostgreSQL の標準機能である、ストリーミングレプリケーションを利用した環境に Transparent Data Encryption for PostgreSQL を初めてインストールする場合に行います。

2.2　アップグレードの種類

本書で説明するアップグレードは以下の 3 つがあります。

- Transparent Data Encryption for PostgreSQL のアップグレード

  Transparent Data Encryption for PostgreSQL のマイナーバージョンまたはメジャーバージョンをアップグレードする場合に行います。

- Free Edition から Enterprise Edition へのアップグレード

  Transparent Data Encryption for PostgreSQL Free Edition から Enterprise Edition にアップグレードする場合に行います。

- 透過的暗号化機能が有効な PostgreSQL のメジャーバージョンアップグレード

  Transparent Data Encryption for PostgreSQL がインストールされ透過的暗号化機能が有効な PostgreSQL をメジャーバージョンアップ(PostgreSQL 9.6 から PostgreSQL 13 にアップグレードなど)する場合に行います。

注

Transparent Data Encryption for PostgreSQL はメジャーバージョン、マイナーバージョンともにダウングレードはできません。

2.3 アンインストールの種類

本書で説明する Transparent Data Encryption for PostgreSQL のアンインストールには以下の 2 つがあります。

- アンインストール

  Transparent Data Encryption for PostgreSQL をアンインストールする場合に行います。

- ストリーミングレプリケーション構成からのアンインストール

  PostgreSQL の標準機能である、ストリーミングレプリケーションを利用した環境から Transparent Data Encryption for PostgreSQL をアンインストールする場合に行います。

第 3 章
動作環境の確認とインストール前の準備

本章は、Transparent Data Encryption for PostgreSQL を使用するために必要な動作環境とインストール前に確認しておくべきことについて説明します。

3.1　PostgreSQL のインストール

Transparent Data Encryption for PostgreSQL を利用するためには、事前に PostgreSQL をインストールしておく必要があります。「3.3.1.2　ソフトウェア要件（6 ページ）」の条件を満たす PostgreSQL バージョンをインストールしてください。

ヒント

Transparent Data Encryption for PostgreSQL Free Edition とは異なり、pgcrypto や psql などのセットアップに必要なツールは同梱していますので、 PostgreSQL のサーバ機能がインストールされていればセットアップが可能です。

3.2　透過的暗号化機能をセットアップするために必要な情報

透過的暗号化機能をセットアップするために必要な PostgreSQL の接続情報を確認します。

表 3-1　透過的暗号化機能をセットアップするために必要な PostgreSQL の接続情報

ポート番号	透過的暗号化機能をセットアップするデータベースが定義された PostgreSQL のサービス待ち受けポート番号です。
データベース名	透過的暗号化機能をセットアップするデータベースの名前です。
スーパユーザ名	透過的暗号化機能をセットアップするデータベースに接続するためのスーパユーザです。
スーパユーザのパスワード	透過的暗号化機能をセットアップするデータベースに接続するためのスーパユーザのパスワードです。
セキュリティ管理ユーザ名	透過的暗号化機能の暗号鍵を管理するための専用のユーザです。
セキュリティ管理ユーザのパスワード	透過的暗号化機能の暗号鍵を管理するための専用のユーザのパスワードです。

重要

禁則文字

本ツールで構築する透過的暗号化環境の中で使用する次のオブジェクトでは、「機種依存文字」「Unicode の重複文字」「改行文字」「空文字」の使用を禁止しています。また、個々のオブジェクトで使用を禁止している文字・文字列は次の通りです。

- ホスト名

{「!」,「'」}を同時使用,「マルチバイト文字」の使用を禁止しています

- データベース名

  {「!」,「'」}を同時使用,「'」,「"」,「/」,「¥」,「=」,「:」,「?」「マルチバイト文字」の使用を禁止しています。

  複数のデータベースインスタンス（データベースクラスタ）を同時に使用する場合、データベース名が重複 しないようご注意ください。

- ユーザ名

  {「!」,「'」}を同時使用,「'」,「"」,「マルチバイト文字」の使用を禁止しています。

- パスワード

  {「!」,「'」}を同時使用,「マルチバイト文字」の使用を禁止しています。

表 3-2　接続情報禁則文字一覧

	マルチバイト文字	「!」,「'」を同時使用	「template1」	「'」	「"」	「/」	「¥」	「=」	「:」	「?」
ホスト名	×	×								
データベース名	×	×	×	×	×	×	×	×	×	×
ユーザ名	×	×		×	×					
パスワード名	×	×								

×…禁則文字として扱われる文字・文字列

3.3　インストール要件の確認

3.3.1　データベースサーバー

Transparent Data Encryption for PostgreSQL をインストールする PostgreSQL がインストールされているサーバーのハードウェアとソフトウェア要件について説明します。

3.3.1.1　ハードウェア要件

Transparent Data Encryption for PostgreSQL のインストールには下記のハードウェア要件を満たす必要があります。

プロセッサ	x86_64 プロセッサ
メモリ容量	約 200M バイト以上を推奨
ディスク容量	任意のディスクに約 100M バイト以上の空き領域

ヒント

AES-NI の利用

AES による暗号化および復号の高速化を目的とした CPU の命令セット AES-NI を利用するためには、以下の条件を満たす必要があります。

- PostgreSQL 9.5 以上に対して透過的暗号化機能が有効となっていること
- Linux では Transparent Data Encryption for PostgreSQL V1.1.4 以降が利用されていること
- OpenSSL がインストールされていること
  - Red Hat Enterprise Linux および Oracle Linux では通常 OpenSSL 1.0.2 系（RHEL7、OL7）または OpenSSL 1.1.1 系（RHEL8、OL8）がインストールされています。

3.3.1.2　ソフトウェア要件

Transparent Data Encryption for PostgreSQL のインストールには下記のソフトウェア要件を満たす必要があります。

PostgreSQL バージョン	オペレーティングシステム（Linux）	
	Red Hat Enterprise Linux 7.1 以上 Oracle Linux 7.1 以上	Red Hat Enterprise Linux 8.1 以上 Oracle Linux 8.1 以上
9.5	×	×
9.6	×	×
10	×	×
11	×	×
12	○	○
13	○	○
必要パッケージ（Linux）	zlib.x86_64 glibc.x86_64	

注

SELinux（Security-Enhanced Linux）機能はサポートしていません。

第 4 章
新規セットアップ

本章では、Transparent Data Encryption for PostgreSQL Enterprise Edition を初めてセットアップする手順について説明します。また、「4.6　ストリーミングレプリケーション構成への新規セットアップ （15 ページ）」手順についても説明します。

重要

Linux 版 Transparent Data Encryption for PostgreSQL は同一データベースインスタンス(データベースクラスタ)内で異なるバージョンの Transparent Data Encryption for PostgreSQL を構成することはサポートしていません。

ヒント

鍵管理機能は PostgreSQL データベースサーバがインストールされた端末リモートコンピュータからも実行が可能です。リモートコンピュータから鍵管理機能を利用する場合、 リモートコンピュータにも Transparent Data Encryption for PostgreSQL をインストールしてください。

4.1　新規セットアップの流れ

1. 「4.2　RPM パッケージのインストール（8 ページ）」
2. 「4.3　postgresql.conf の編集（9 ページ）」
3. 「4.4　透過的暗号化機能の有効化（9 ページ）」
4. 「4.5　よりセキュアな運用のための設定（13 ページ）」

   ヒント

   PostgreSQL のユーザデータを暗号化するためには、上記手順完了後に以下の作業が必要です。詳細は『列単位暗号化 透過的暗号化機能 利用の手引き』をご確認ください。

5. 利用するモードの検討
   - 簡易 TDE モード
   - 標準 TDE モード
   - AWS KMS モード
6. 利用する暗号化アルゴリズムの検討
   - aes(Rijndael-128)
   - bf (Blowfish)
7. 暗号鍵のパスフレーズの検討
8. 通信経路の暗号化の検討

9. 暗号鍵の登録
10. 暗号化データ型を含むユーザテーブルを作成
11. 暗号化データ型のユーザデータを操作（挿入/更新/削除および参照）

4.2 RPM パッケージのインストール

OS の root 権限で以下の手順に従って RPM パッケージをインストールしてください。

1. Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を mount します。

   次の例では CD ドライブ /dev/sr0 に挿入した Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を/mnt/cdrom に mount しています。

```
# mount -t iso9660 /dev/sr0 /mnt/cdrom
```

2. rpm -ivh コマンドを実行し、RPM パッケージをインストールします。

```
# cd /mnt/cdrom/linux/rpm
# rpm -ivh tde_for_pg13-2.1.1-0.el8.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tde_for_pg13-2.1.1-0.el8         ################################# [100%]
INFO: Transparent Data Encryption for PostgreSQL 13
      was installed successfully.
HINT: To complete validation of transparent data encryption feature,
      please add "/opt/nec/tdeforpg13/lib/data_encryption.so" to
      'shared_preload_libraries' parameter in 'postgresql.conf' file
      and require a PostgreSQL server restart to take effect.
```

   RPM の命名規則は以下の通りですので、使用する OS や PostgreSQL のバージョンに合わせて RPM パッケージをインストールしてください。

tde_for_pg<*PostgreSQL バージョン*>-<*Transparent Data Encryption for PostgreSQL バージョン*>.<*Red Hat Enterprise Linux バージョン*>.x86_64.rpm

   - PostgreSQL バージョン

     Transparent Data Encryption for PostgreSQL が対応する PostgreSQL バージョンを示します。9.5 は 95、9.6 は 96、10 は 10、11 は 11、12 は 12、13 は 13 と表示されます。

   - Transparent Data Encryption for PostgreSQL バージョン

     表記形式は X.Y.Z-N です。X.Y はメジャーバージョン、Z はマイナーバージョン、N がビルド番号を示します

   - Red Hat Enterprise Linux バージョン

     Transparent Data Encryption for PostgreSQL が対応する OS を示します。Red Hat Enterprise Linux 7 は el7、Red Hat Enterprise Linux 8 は el8 と表示されます。

ヒント

RPM パッケージをインストールする際に--prefix オプションを使用することでインストール先ディレクトリを指定することができます。RPM パッケージのインストール先に存在しないディレクトリを指定した場合、インストール時にディレクトリが新規に作成され、 オーナーおよびグループは root となります。

次の例では/cal/nec ディレクトリにインストールしています。

```
# rpm -ivh --prefix /cal/nec tde_for_pg13-2.1.1-0.el8.x86_64.rpm
```

4.3　postgresql.conf の編集

RPM パッケージのインストールが完了後、透過的暗号化機能を利用するために PostgreSQL の設定ファイル（postgresql.conf）を変更し、設定の変更を有効にします。

1. OS のデータベース管理者ユーザ（一般的に postgres）でログインします。
2. PostgreSQL の設定ファイル（postgresql.conf）の shared_preload_libraries パラメータに Transparent Data Encryption for PostgreSQL の共有ライブラリ data_encryption.so を設定します。

   [postgresql.conf 設定例]

   ```
   shared_preload_libraries='/opt/nec/tdeforpg13/lib/data_encryption.so'
   ```

3. 変更した設定を有効にするため、PostgreSQL を再起動します。

   次の例では、pg_ctl[*1] コマンドを利用して PostgreSQL を再起動しています。

   ```
   $ pg_ctl restart
   ```

4.4　透過的暗号化機能の有効化

透過的暗号化機能を有効化する方法として以下の 2 つを提供しています。

- 「4.4.1　透過的暗号化機能を対話型で有効化する方法（10 ページ）」
- 「4.4.2　透過的暗号化機能を非対話型で有効化する方法（11 ページ）」

*1　pg_ctl コマンドの詳細な利用方法は PostgreSQL マニュアルをご確認ください。

4.4.1　透過的暗号化機能を対話型で有効化する方法

OS の root 権限で以下の手順に従って対話型で透過的暗号化機能を有効化してください。

1. 引数なしで cipher_setup.sh を実行します。

   次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh
Transparent data encryption feature setup script
```

2. 透過的暗号化機能を有効化、または無効化するか聞かれますので「1」（有効化）を入力します。

```
Please select from the setup menu below
Transparent data encryption feature setup menu
1: activate  the transparent data encryption feature
2: inactivate the transparent data encryption feature
select menu [1 - 2] > 1
```

3. 「3.2　透過的暗号化機能をセットアップするために必要な情報（4 ページ）」を参考に PostgreSQL への接続情報、およびセキュリティ管理ユーザを入力します。

   透過的暗号化機能を有効化するデータベース は事前に作成されている必要があります。入力したセキュリティ管理ユーザ名が PostgreSQL に存在しない場合、新規に PosgreSQL ユーザを作成します。この際にセキュリティ管理ユーザは、 MD5 パスワードで定義されます。

注

セキュリティ管理ユーザとして PostgreSQL のスーパユーザを指定することはできません。

各項目で入力する内容については後述します。

```
Please enter database server port to connect : 5432
Please enter database user name to connect : postgres
Please enter password for authentication : *****
Please enter database name to connect : tdedb
Please enter normal database user name for security management: secman
Please enter password for database user secman: *****
Retype password for database user secman: *****
```

表 4-1　各項目の説明

項目	説明
Please enter database server port to connect	ポート番号
Please enter database user name to connect	スーパユーザ名
Please enter password for authentication	スーパユーザのパスワード
Please enter database name to connect	データベース名
Please enter normal database user name for security management	セキュリティ管理ユーザ名
Please enter password for database user secman	セキュリティ管理ユーザのパスワード

項目	説明
Retype password for database user secman	セキュリティ管理ユーザのパスワード(再入力)

入力した情報に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が有効化されます。また、セキュリティ管理ユーザの接続情報が記載された設定ファイルが作成されます（本手順では/opt/nec/tdeforpg13/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザは、このファイルを透過的暗号化機能コマンド（pgtde）実行時の接続情報ファイルとして使用することが可能です。

```
INFO: Transparent data encryption feature has been activated
PostgreSQL connection info for security user has created: /opt/nec/tdeforpg
13/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQ
L security user
```

4.4.2　透過的暗号化機能を非対話型で有効化する方法

必要な情報を記載した透過的暗号化機能の構成ファイル（cipher_setup.conf）を使用することで非対話型で透過的暗号化機能を有効化することが可能です。OS の root 権限で以下の手順に従って非対話型で透過的暗号化機能を有効化してください。

1. 透過的暗号化機能の構成ファイル（cipher_setup.conf）を準備します。

   a. インストールディレクトリ配下の template/cipher_setup.conf.template を同ディレクトリ配下の conf/cipher_setup.conf としてコピーします。

   次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

   ```
   # cp /opt/nec/tdeforpg13/template/cipher_setup.conf.template \
   /opt/nec/tdeforpg13/conf/cipher_setup.conf
   ```

   b. 先ほどの手順で作成した cipher_setup.conf を「3.2　透過的暗号化機能をセットアップするために必要な情報（4 ページ）」を参考に編集します。

   [cipher_setup.conf 設定例]

   ```
   connect_db_port=5432
   connect_db_name=tdedb
   connect_db_user=postgres
   connect_db_password=*****
   security_db_user=*****
   security_db_password=*****
   ```

   注

   「設定項目=設定値」の書式でスペースやタブを使わず記載します。

表 4-2　各項目の説明

項目	説明
connect_db_port	ポート番号
connect_db_name	データベース名
connect_db_user	スーパユーザ名
connect_db_password	スーパユーザのパスワード
security_db_user	セキュリティ管理ユーザ名
security_db_password	セキュリティ管理ユーザのパスワード

2. `-s 1` オプションを利用して `cipher_setup.sh` を実行します。透過的暗号化機能の構成ファイルを指定しない場合、インストールディレクトリ配下の `conf/cipher_setup.conf` の使用を試みます。

   透過的暗号化機能を有効化するデータベース は事前に作成されている必要があります。入力したセキュリティ管理ユーザ名が PostgreSQL に存在しない場合、新規に PosgreSQL ユーザを作成します。この際にセキュリティ管理ユーザは、 MD5 パスワードで定義されます。

   注

   セキュリティ管理ユーザとして PostgreSQL のスーパユーザを指定することはできません。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 1
```

   ヒント

   透過的暗号化機能の構成ファイルは書式が正しければファイル名は自由です。次の例では透過的暗号化機能の構成ファイルとして`/tmp/setup.conf` を指定しています。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 1 \
/tmp/setup.conf
```

透過的暗号化機能の構成ファイルの内容に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が有効化されます。また、セキュリティ管理ユーザの接続情報が記載された設定ファイルが作成されます（本手順では`/opt/nec/tdeforpg13/conf/pgtde_secuser.properties`）。暗号鍵を管理する OS ユーザは、このファイルを透過的暗号化機能コマンド（`pgtde`）実行時の接続情報ファイルとして使用することが可能です。

```
INFO: Transparent data encryption feature has been activated
PostgreSQL connection info for security user has created: /opt/nec/tdeforpg13/conf/
pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQL securi
ty user
```

4.5　よりセキュアな運用のための設定

透過的暗号化機能は、OS ユーザおよびファイルの権限を適切に設定することでよりセキュアな運用が実現できます。よりセキュアな運用を行いたい場合は以下の設定を実施してください。

1. 透過的暗号化機能をよりセキュアな状態で運用するためには、各機能毎に OS ユーザおよび OS グループを作成します。

   それぞれの OS ユーザが適切な PostgreSQL ユーザを使用するような運用方針を策定する必要があります。作成するユーザと対応する PostgreSQL ユーザの一覧については下記をご参考の上作成してください。

表 4-3　作成する OS ユーザー一覧

OS ユーザ	OS グループ	役割	使用可能な PostgreSQL ユーザ
データベース管理者	透過的暗号化機能管理グループ	PostgreSQL 起動ユーザであり、PostgreSQL に対する全権限を持つユーザ。	スーパユーザ
セキュリティ管理者	透過的暗号化機能管理グループ	透過的暗号化機能で利用する鍵の管理権限を持つユーザ	透過的暗号化機能のセットアップで作成または指定したセキュリティ管理ユーザ
アプリケーション管理者（アプリケーション開発者）	透過的暗号化機能利用グループ	透過的暗号化機能を利用しているユーザデータに対する暗号化・復号権限を持つユーザ	透過的暗号化機能を利用するユーザデータにアクセスできる一般ユーザ

   次の例では透過的暗号化機能管理グループ「tde_manger」と透過的暗号化機能利用グループ「tde_user」を作成し、データベース管理者「dbauser」、セキュリティ管理者「secuser」、アプリケーション管理者（アプリケーション開発者）「apuser」をそれぞれのグループに所属させるよう作成しています。

```
# groupadd tde_manager
# groupadd tde_user
# useradd -G tde_manager secuser
# useradd -G tde_manager dbauser
# useradd -G tde_user apuser
```

2. 透過的暗号化機能をよりセキュアな状態で運用するためには、各種ファイルをそれぞれ適切な所有者に設定します。

   次の表を参考に、作成したユーザ毎にファイルの権限を設定してください。

表 4-4　アクセス権限設定を推奨する透過的暗号化機能関連ファイル一覧

対象ファイル	所有者
conf/aws_info.properties	データベース管理者
conf/kms_info.properties	データベース管理者
conf/pgtde_secuser.properties	セキュリティ管理者

対象ファイル	所有者
lib/jar/pgtde.jar	アプリケーション管理者（アプリケーション開発者）
lib/jar/pgtde_regist.jar	セキュリティ管理者
lib/jar/kms-agent.jar	データベース管理者

次の例では、データベース管理者に「dbauser」、セキュリティ管理者に「secuser」、アプリケーション管理者（アプリケーション開発者）に「apuser」として各種ファイルの所有者を設定しています。また、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

```
# cd /opt/nec/tdeforpg13/
# chown dbauser:tde_manager conf/aws_info.properties
# chown dbauser:tde_manager conf/kms_info.properties
# chown secuser:tde_manager conf/pgtde_secuser.properties
# chown apuser:tde_user lib/jar/pgtde.jar
# chown secuser:tde_manager lib/jar/pgtde_regist.jar
# chown dbauser:tde_manager lib/jar/kms-agent.jar
```

上記ファイルの権限設定により、透過的暗号化機能コマンド（pgtde）の各-m オプションの実行がユーザ毎に以下のように制限されます。(各-m オプションの詳細は『透過的暗号化機能 利用の手引』をご確認ください)

表 4-5　モード毎実行可能ユーザー覧

各-m オプション	実行可能ユーザ
暗号鍵の登録・更新(-m regist)	セキュリティ管理者
モードの変更(-m switch)	
利用状況を表示(-m show)	
最新の暗号鍵による再暗号化(-m cipher)	アプリケーション管理者（アプリケーション開発者）

3. 透過的暗号化機能を利用したいデータベースの一般ユーザは、暗号鍵情報テーブルに対して適切なアクセス権限を設定します。対象のデータベースに存在する暗号鍵情報テーブル(cipher_key_table)に対して GRANT 文を利用して一般ユーザに UPDATE と DELETE 権限を設定します。

   次の例では、データベースの一般ユーザ「apuser」に対して暗号鍵情報テーブル(cipher_key_table)の UPDATE と DELETE 権限を設定しています。

```
=# CREATE ROLE apuser WITH LOGIN ENCRYPTED PASSWORD '******';
=# GRANT UPDATE ON cipher_key_table TO apuser;
=# GRANT DELETE ON cipher_key_table TO apuser;
```

ヒント

PostgreSQL のセキュリティ管理ユーザに透過的暗号化機能のセットアップで作成したユーザ以外の一般ユーザを割り当てる場合、対象のデータベースに対して次の権限を設定しま

す。次の例では 一般ユーザ「secuser」を透過的暗号化機能のセキュリティ管理者用として設定しています。

```
=# CREATE ROLE secuser WITH LOGIN ENCRYPTED PASSWORD '******';
=# GRANT INSERT ON cipher_key_table TO secuser;
=# GRANT UPDATE ON cipher_key_table TO secuser;
=# GRANT DELETE ON cipher_key_table TO secuser;
=# GRANT SELECT ON keyid_table TO secuser;
=# GRANT INSERT ON keyid_table TO secuser;
=# GRANT UPDATE ON keyid_table TO secuser;
=# GRANT DELETE ON keyid_table TO secuser;
=# GRANT EXECUTE ON FUNCTION cipher_key_backup() TO secuser;
```

4.6　ストリーミングレプリケーション構成への新規セットアップ

PostgreSQL の標準機能である、ストリーミングレプリケーションを利用した環境に Transparent Data Encryption for PostgreSQL を初めてインストールする手順について説明します。以下の手順でセットアップを行います。なお、手順 1～4 は参考として記載していますが、詳細な手順については、各手順の参照先をご確認ください。

表 4-6　インストール時の手順要否

手順	作業項目		参照先
	プライマリサーバ	スタンバイサーバ	
1	PostgreSQL のインストール		関連リンク参照
2	インスタンスの作成・設定		関連リンク参照
3		インスタンスの作成・設定	関連リンク参照
4	ストリーミングレプリケーションの状態確認		関連リンク参照
5	rpm パッケージのインストール		「4.6.1　RPM パッケージのインストール（手順 5）（16 ページ）」
6	postgresql.conf の編集		「4.6.2　postgresql.conf の編集（手順 6）（16 ページ）」
7	透過的暗号化機能の有効化		「4.6.3　透過的暗号化機能の有効化（手順 7）（16 ページ）」
8	よりセキュアな運用のための設定		「4.6.4　よりセキュアな運用のための設定（手順 8）（16 ページ）」

関連リンク

PostgreSQL のインストール（PostgreSQL の Windows インストーラ、Linux ディストリビューション・パッケージなどのリンク集、およびインストールガイド URL https://www.postgresql.jp/download）

インスタンスの作成・設定（最新バージョンの PostgreSQL マニュアル URL https://www.postgresql.jp/document/current/html/admin.html）

ストリーミングレプリケーションの状態確認（最新バージョンの PostgreSQL マニュアル URL https://www.postgresql.jp/document/current/html/high-availability.html）

4.6.1　RPM パッケージのインストール（手順 5）

ストリーミングレプリケーションを利用する場合は、「4.2　RPM パッケージのインストール（8 ページ）」を参考にプライマリサーバとスタンバイサーバの両方にインストールを行ってください。

重要

RPM パッケージによるインストールパスの指定は、プライマリサーバとスタンバイサーバを同じディレクトリパスに統一する必要があります。

4.6.2　postgresql.conf の編集（手順 6）

ストリーミングレプリケーションを利用する場合は、「4.3　postgresql.conf の編集（9 ページ）」を参考にプライマリサーバとスタンバイサーバの両方の postgresql.conf の shared_preload_libraries パラメータに Transparent Data Encryption for PostgreSQL の共有ライブラリ data_encryption.so を設定してください。

4.6.3　透過的暗号化機能の有効化（手順 7）

「4.4　透過的暗号化機能の有効化（9 ページ）」を参考にプライマリサーバのみ透過的暗号化機能を有効化してください。

4.6.4　よりセキュアな運用のための設定（手順 8）

ストリーミングレプリケーションを利用した環境でよりセキュアな運用を行いたい場合は、「4.5　よりセキュアな運用のための設定（13 ページ）」を参考にプライマリサーバとスタンバイサーバの両方を同一の構成となるよう設定してください。

第 5 章
再インストール

本章では、必要なファイルが削除された場合や、RPM パッケージからオリジナルの設定ファイルをインストールしたい場合などにインストール済みの RPM パッケージを再インストールする手順について説明します。

5.1　RPM パッケージの再インストール

以下の手順に従って RPM パッケージを再インストールしてください。インストール済みの RPM パッケージに対して、バージョンが同一の RPM パッケージを再度インストールする場合にのみ本手順を実施してください。

ヒント

異なるバージョンをインストールしたい場合は「第 7 章　アップグレード（27 ページ）」をご確認ください。

1. OS の管理者ユーザ（root 権限）でログインします。
2. Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を `mount` します。

   次の例では CD ドライブ `/dev/sr0` に挿入した Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を `/mnt/cdrom` に `mount` しています。

```
# mount -t iso9660 /dev/sr0 /mnt/cdrom
```

3. `rpm -ivh` コマンドを実行する際に`--replacepkgs` と`--replacefiles` オプションを利用し、RPM パッケージを再インストールします。

```
# cd /mnt/cdrom/linux/rpm
# rpm -ivh --replacepkgs --replacefiles tde_for_pg13-2.1.1-0.el8.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tde_for_pg13-2.1.1-0.el8         ################################# [100%]
```

注

RPM パッケージを任意のディレクトリにインストールしている場合は`--prefix` でインストール先ディレクトリを指定します。`--prefix` を指定せずに再インストールした場合、デフォルトディレクトリ（`/opt/nec`）にインストールされます。

第 6 章
アンインストール

本章では、Transparent Data Encryption for PostgreSQL Enterprise Edition をアンインストールする手順について説明します。また、「6.6　ストリーミングレプリケーション構成からのアンインストール（22 ページ）」や「6.7　透過的暗号化機能の再有効化（22 ページ）」についても説明します。

重要

透過的暗号化機能を無効化しても暗号化されたデータは復号されません。そのため、Transparent Data Encryption for PostgreSQL Enterprise Edition アンインストール後も暗号化されたデータを利用する場合は、アンインストール前に暗号化されたデータを復号してください。

6.1　アンインストールの流れ

1. 「6.2　透過的暗号化機能の無効化（18 ページ）」
2. 「6.3　RPM パッケージのアンインストール（20 ページ）」
3. 「6.4　postgresql.conf の編集（21 ページ）」
4. 「6.5　インストールディレクトリの削除（21 ページ）」

6.2　透過的暗号化機能の無効化

透過的暗号化機能を無効化する方法として以下の 2 つを提供しています。

- 「6.2.1　透過的暗号化機能を対話型で無効化する方法（18 ページ）」
- 「6.2.2　透過的暗号化機能を非対話型で無効化する方法（19 ページ）」

6.2.1　透過的暗号化機能を対話型で無効化する方法

OS の root 権限で以下の手順に従って対話型で透過的暗号化機能を無効化してください。

1. 引数なしで `cipher_setup.sh` を実行します。

   次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が`/opt/nec`（デフォルト）にインストールされていることとします。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh
Transparent data encryption feature setup script
```

2. 透過的暗号化機能を有効化、または無効化するか聞かれますので「2」（無効化）を入力します。

```
Please select from the setup menu below
Transparent data encryption feature setup menu
1: activate  the transparent data encryption feature
2: inactivate the transparent data encryption feature
select menu [1 - 2] > 2
```

3. 透過的暗号化機能を無効化する PostgreSQL への接続情報を入力します。

```
Please enter database server port to connect : 5432
Please enter database user name to connect : postgres
Please enter password for authentication : *****
Please enter database name to connect : tdedb
```

表 6-1　各項目の説明

項目	説明
Please enter database server port to connect	ポート番号
Please enter database user name to connect	スーパユーザ名
Please enter password for authentication	スーパユーザのパスワード
Please enter database name to connect	データベース名

入力した情報に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が無効化されます。

```
INFO: The transparent data encryption feature has been inactivated
```

6.2.2　透過的暗号化機能を非対話型で無効化する方法

必要な情報を記載した透過的暗号化機能の構成ファイル（cipher_setup.conf）を使用することで非対話型で透過的暗号化機能を無効化することが可能です。OS の root 権限で以下の手順に従って非対話型で透過的暗号化機能を有効化してください。

1. 透過的暗号化機能の構成ファイル（cipher_setup.conf）を準備します。

   a. インストールディレクトリ配下の template/cipher_setup.conf.template を同ディレクトリ配下の conf/cipher_setup.conf としてコピーします。

   次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

   ```
   # cp /opt/nec/tdeforpg13/template/cipher_setup.conf.template \
   /opt/nec/tdeforpg13/conf/cipher_setup.conf
   ```

   b. 先ほどの手順で作成した cipher_setup.conf を「3.2　透過的暗号化機能をセットアップするために必要な情報（4 ページ）」を参考に編集します。

   [cipher_setup.conf 設定例]

   ```
   connect_db_port=5432
   connect_db_name=tdedb
   ```

```
connect_db_user=postgres
connect_db_password=*****
security_db_user=*****
security_db_password=*****
```

注

「設定項目=設定値」の書式でスペースやタブを使わず記載します。

表 6-2　各項目の説明

項目	説明	無効化の際の入力有無
connect_db_port	ポート番号	入力必要
connect_db_name	データベース名	入力必要
connect_db_user	スーパユーザ名	入力必要
connect_db_password	スーパユーザのパスワード	入力必要
security_db_user	セキュリティ管理ユーザ名	入力不要
security_db_password	セキュリティ管理ユーザのパスワード	入力不要

2. 非対話型で透過的暗号化機能を無効化するために-s 2 オプションを利用して cipher_setup.sh を実行します。透過的暗号化機能の構成ファイルを指定しない場合、インストールディレクトリ配下の conf/cipher_setup.conf の使用を試みます。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 2
```

ヒント

透過的暗号化機能の構成ファイルは書式が正しければファイル名は自由です。次の例では透過的暗号化機能の構成ファイルとして/tmp/setup.conf を指定しています。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 2 \
/tmp/setup.conf
```

透過的暗号化機能の構成ファイルの内容に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が無効化されます。

```
INFO: The transparent data encryption feature has been inactivated
      Encrypted data will NOT be able to access from now.
      Please reactivate if you need to re-access them.
```

6.3　RPM パッケージのアンインストール

OS の root 権限で以下の手順に従って RPM パッケージをアンインストールしてください。

1. rpm -ql を実行し、 Transparent Data Encryption for PostgreSQL がインストールされていることを確認します。

```
# rpm -ql | grep tde_for_pg13-2.1.1-0.el8.x86_64
/opt/nec/tdeforpg13
...
/opt/nec/tdeforpg13/template/cipher_setup.conf.template
```

2. rpm -e コマンドを実行し、Transparent Data Encryption for PostgreSQL をアンインストールします。

```
# rpm -e tde_for_pg13-2.1.1-0.el8.x86_64
INFO: Transparent Data Encryption for PostgreSQL 13
      was uninstalled successfully.
HINT: To complete invalidation of transparent data encryption feature,
      please remove "data_encryption.so" from
      'shared_preload_libraries' parameter in 'postgresql.conf'
```

6.4　postgresql.conf の編集

透過的暗号化機能を利用を停止するために PostgreSQL の設定ファイル（postgresql.conf）を変更し、設定の変更を有効にします。

1. OS のデータベース管理者ユーザ（一般的に postgres）でログインします。
2. PostgreSQL の設定ファイル（postgresql.conf）の shared_preload_libraries パラメータに設定されている Transparent Data Encryption for PostgreSQL の共有ライブラリ data_encryption.so を削除、またはパラメータ自体をコメントアウトします。

```
shared_preload_libraries=''
```

3. 変更した設定を有効にするため、PostgreSQL を再起動します。

   次の例では、pg_ctl[*1] コマンドを利用して PostgreSQL を再起動しています。

```
$ pg_ctl restart
```

6.5　インストールディレクトリの削除

Transparent Data Encryption for PostgreSQL を今後利用しない場合、インストールディレクトリを削除します。

1. インストールディレクトリを削除します。

*1　pg_ctl コマンドの詳細な利用方法は PostgreSQL マニュアルをご確認ください。

次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

```
# cd /opt/nec
# ls -ld tdeforpg13
drwxr-xr-x 5 root root 37  2 月 28 19:35 tdeforpg13
# rm -rf tdeforpg13
```

6.6　ストリーミングレプリケーション構成からのアンインストール

ストリーミングレプリケーションを利用した環境から Transparent Data Encryption for PostgreSQL をアンインストールする手順について説明します。以下の手順でアンインストールします。

また、アンインストール時のプライマリサーバとスタンバイサーバのセットアップ手順の要否については、以下の通りです。

表 6-3　アンインストール時の手順要否

手順	作業項目		参照先
	プライマリサーバ	スタンバイサーバ	
1	透過的暗号化機能の無効化		「6.2　透過的暗号化機能の無効化（18 ページ）」
2	RPM パッケージのアンインストール		「6.3　RPM パッケージのアンインストール（20 ページ）」
3	postgresql.conf の編集		「6.4　postgresql.conf の編集（21 ページ）」
4	インストールディレクトリの削除		「6.5　インストールディレクトリの削除（21 ページ）」

6.7　透過的暗号化機能の再有効化

透過的暗号化機能を再有効化する方法として以下の 2 つを提供しています。

注

透過的暗号化機能を有効化している状態でデータベースを削除した場合、同名のデータベースを再作成しても透過的暗号化機能を有効化することはできません。再度有効化したい場合、PP サポートサービスにご連絡ください。

- 「6.7.1　透過的暗号化機能を対話型で再有効化する方法（23 ページ）」
- 「6.7.2　透過的暗号化機能を非対話型で再有効化する方法（24 ページ）」

6.7.1　透過的暗号化機能を対話型で再有効化する方法

OS の root 権限で以下の手順に従って対話型で透過的暗号化機能を再有効化してください。

1. 引数なしで cipher_setup.sh を実行します。

   次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh
Transparent data encryption feature setup script
```

2. 透過的暗号化機能を有効化、または無効化するか聞かれますので「1」（再有効化）を入力します。

```
Please select from the setup menu below
Transparent data encryption feature setup menu
1: activate  the transparent data encryption feature
2: inactivate the transparent data encryption feature
select menu [1 - 2] > 1
```

3. 透過的暗号化機能を再有効化する PostgreSQL への接続情報を入力します。

```
Please enter database server port to connect : 5432
Please enter database user name to connect : postgres
Please enter password for authentication : *****
Please enter database name to connect : tdedb
```

表 6-4　各項目の説明

項目	説明
Please enter database server port to connect	ポート番号
Please enter database user name to connect	スーパユーザ名
Please enter password for authentication	スーパユーザのパスワード
Please enter database name to connect	データベース名

4. 再有効化処理を行ってよいか確認されるため、問題がない場合は「Yes」を入力します。

```
WARN: Are you sure you want to reactivate the transparent data encryption feat
ure?
Please input [Yes/No] > Yes
```

5. セキュリティ管理ユーザを入力します。

   入力したセキュリティ管理ユーザ名が PostgreSQL に存在しない場合、新規に PosgreSQL ユーザを作成します。この際にセキュリティ管理ユーザは、 MD5 パスワードで定義されます。

注

セキュリティ管理ユーザとして PostgreSQL のスーパユーザを指定することはできません。

- 新規にユーザを作成する場合。次の例では、セキュリティ管理ユーザとして「secuser」を新規に作成します。

```
Please enter normal database user name for security management: secuser
Please enter password for database user secuser: *****
Retype password for database user secuser: *****
```

- 既存ユーザをセキュリティ用ユーザとして使用する場合。次の例ではセキュリティ管理ユーザとして既存ユーザである「secman」を指定します。

```
Please enter normal database user name for security management: secman
Please enter password for database user secman: *****
WARN: Do you want to use existing user: "secman" for security management?
Please input [Yes/No] > Yes
```

表 6-5 各項目の説明

項目	説明
Please enter normal database user name for security management	セキュリティ管理ユーザ名
Please enter password for database user secman	セキュリティ管理ユーザのパスワード
Retype password for database user secman	セキュリティ管理ユーザのパスワード(再入力)

入力した情報に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が再有効化されます。また、セキュリティ管理ユーザの接続情報が記載された設定ファイルが作成されます（本手順では/opt/nec/tdeforpg13/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザは、このファイルを透過的暗号化機能コマンド（pgtde）実行時の接続情報ファイルとして使用することが可能です。

```
INFO: Transparent data encryption feature has been activated
PostgreSQL connection info for security user has created:  /opt/nec/tdeforp
g13/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQ
L security user
```

6.7.2 透過的暗号化機能を非対話型で再有効化する方法

必要な情報を記載した透過的暗号化機能の構成ファイル（cipher_setup.conf）を使用することで非対話型で透過的暗号化機能を再有効化することが可能です。OS の root 権限で以下の手順に従って非対話型で透過的暗号化機能を再有効化してください。

1. 透過的暗号化機能の構成ファイル（cipher_setup.conf）を準備します。

   a. インストールディレクトリ配下の template/cipher_setup.conf.template を同ディレクトリ配下の conf/cipher_setup.conf としてコピーします。

次の例では、PostgreSQL 13 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）にインストールされていることとします。

```
# cp /opt/nec/tdeforpg13/template/cipher_setup.conf.template \
/opt/nec/tdeforpg13/conf/cipher_setup.conf
```

b. 先ほどの手順で作成した cipher_setup.conf を「3.2　透過的暗号化機能をセットアップするために必要な情報（4 ページ）」を参考に編集します。

[cipher_setup.conf 設定例]

```
connect_db_port=5432
connect_db_name=tdedb
connect_db_user=postgres
connect_db_password=*****
security_db_user=*****
security_db_password=*****
```

注

「設定項目=設定値」の書式でスペースやタブを使わず記載します。

表 6-6　各項目の説明

項目	説明
connect_db_port	ポート番号
connect_db_name	データベース名
connect_db_user	スーパユーザ名
connect_db_password	スーパユーザのパスワード
security_db_user	セキュリティ管理ユーザ名
security_db_password	セキュリティ管理ユーザのパスワード

2. -s 1 オプションを利用して cipher_setup.sh を実行します。透過的暗号化機能の構成ファイルを指定しない場合、インストールディレクトリ配下の conf/cipher_setup.conf の使用を試みます。

入力したセキュリティ管理ユーザ名が PostgreSQL に存在しない場合、新規に PosgreSQL ユーザを作成します。この際にセキュリティ管理ユーザは、 MD5 パスワードで定義されます。

注

セキュリティ管理ユーザとして PostgreSQL のスーパユーザを指定することはできません。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 1
```

ヒント

透過的暗号化機能の構成ファイルは書式が正しければファイル名は自由です。次の例では透過的暗号化機能の構成ファイルとして/tmp/setup.conf を指定しています。

```
# /opt/nec/tdeforpg13/bin/cipher_setup.sh -s 1 \
/tmp/setup.conf
```

透過的暗号化機能の構成ファイルの内容に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が再有効化されます。また、セキュリティ管理ユーザの接続情報が記載された設定ファイルが作成されます（本手順では/opt/nec/tdeforpg13/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザは、このファイルを透過的暗号化機能コマンド（pgtde）実行時の接続情報ファイルとして使用することが可能です。

```
INFO: Reactivating transparent data encryption feature.
INFO: Transparent data encryption feature has been activated
PostgreSQL connection info for security user has created:  /opt/nec/tdeforpg13/conf
/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQL securi
ty user
```

第 7 章
アップグレード

本章では下記 3 パターンのアップグレードについて説明します。

- 「7.1　Free Edition から Enterprise Edition へのアップグレード（27 ページ）」
- 「7.2　Transparent Data Encryption for PostgreSQL のアップグレード（28 ページ）」
- 「7.3　透過的暗号化機能が有効な PostgreSQL のメジャーバージョンアップグレード（30 ページ）」

注

以下のような場合は PP サポートサービスにご連絡ください。

- クラスタ構成のアップグレードをご検討の場合

  クラスタ構成の仕様（利用製品）によっては、待機系のアップグレード手順が異なります。
- クラスタ構成で透過的暗号化機能を有効化した端末以外で透過的暗号化機能を制御したい場合
- PostgreSQL の標準機能ストリーミングレプリケーション構成でのアップグレードをご検討の場合

7.1　Free Edition から Enterprise Edition へのアップグレード

アップグレードを行う前に `pg_dumpall` を使用してデータベース全体のバックアップを取得していただくことを推奨いたします。

Transparent Data Encryption for PostgreSQL Free Edition を利用しているデータベースの Transparent Data Encryption for PostgreSQL Enterprise Edition へのアップグレードを行う場合、バージョン番号 X.Y.Z.N の X と Y がアップグレード先のバージョンと同一である場合に限り、以下の手順に従ってアップグレードを行ってください。この条件に合致しない場合は、後述する「7.3　透過的暗号化機能が有効な PostgreSQL のメジャーバージョンアップグレード（30 ページ）」をご参照ください。

1. Transparent Data Encryption for PostgreSQL Free Edition が提供している `bin/cipher_setup.sh` を使用し、データベースにインストールされている透過的暗号化機能を無効化してください。無効化の手順は「6.2　透過的暗号化機能の無効化（18 ページ）」を参考に実施してください。
2. 透過的暗号化機能の無効化が完了したデータベースにスーパユーザで接続し、インストールされている pgcrypto モジュールを DROP EXTENSION クエリでアンインストールします。

```
=# DROP EXTENSION pgcrypto;
DROP EXTENSION
```

3. OS のデータベース管理者ユーザ（一般的に postgres）でログインし、透過的暗号化機能を利用しているデータベースを停止します。

   次の例では、pg_ctl[*1] コマンドを利用して PostgreSQL を停止しています。

```
$ pg_ctl stop
waiting for server to shut down.... done
server stopped
```

4. 「4.2　RPM パッケージのインストール（8 ページ）」を参考にアップグレード先の Transparent Data Encryption for PostgreSQL Enterprise Edition をインストールします。
5. 「4.3　postgresql.conf の編集（9 ページ）」を参考に postgresql.conf ファイルを編集し、PostgreSQL を起動します。shared_preload_libraries パラメータには、Transparent Data Encryption for PostgreSQL Free Edition で設定した値が記載されているため、当該設定を削除の上で設定値を変更してください。
6. 新規インストールした Transparent Data Encryption for PostgreSQL Enterprise Edition の bin/cipher_setup.sh を使用して透過的暗号化機能を再有効化することでアップグレードが完了します。再有効化の手順は「6.7　透過的暗号化機能の再有効化（22 ページ）」をご確認ください。

   Transparent Data Encryption for PostgreSQL Free Edition からのアップグレードを伴う対話型の再有効化の場合、次の確認メッセージが出力されます。問題がない場合は「Yes」を入力します。次の例 では Transparent Data Encryption for PostgreSQL Free Edition V1.1.0 から Transparent Data Encryption for PostgreSQL Enterprise Edition V1.1.1 へのアップグレー ドを実施しています。

```
WARN: Are you sure you want to upgrade transparent data encryption feature fro
m "Free Edition 1.1.0.0" to "Enterprise Edition 1.1.1.0"?
Please input [Yes/No] > Yes
```

7. 以上でアップグレードは完了となります。必要に応じて旧バージョンの Transparent Data Encryption for PostgreSQL Free Edition をアンインストールしてください。

7.2　Transparent Data Encryption for PostgreSQL のアップグレード

アップグレードを行う前に pg_dumpall を使用してデータベース全体のバックアップを取得していただくことを推奨いたします。

*1　pg_ctl コマンドの詳細な利用方法は PostgreSQL マニュアルをご確認ください。

Transparent Data Encryption for PostgreSQL のメジャーバージョン、マイナーバージョンともに以下の手順に従ってアップグレードを行ってください。

1. 「6.2　透過的暗号化機能の無効化（18 ページ）」を参考に透過的暗号化機能を無効化します。
2. OS のデータベース管理者ユーザ（一般的に postgres）でログインし、透過的暗号化機能を利用しているデータベースを停止します。

   次の例では、pg_ctl[*2] コマンドを利用して PostgreSQL を停止しています。

```
$ pg_ctl stop
waiting for server to shut down.... done
server stopped
```

3. OS の管理者ユーザ（root 権限）でログインし、Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を mount します。

   次の例では CD ドライブ /dev/sr0 に挿入した Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体を/mnt/cdrom に mount しています。

```
# mount -t iso9660 /dev/sr0 /mnt/cdrom
```

4. rpm -Uvh コマンドを実行し、RPM パッケージをアップグレードインストールします。

```
# cd /mnt/cdrom/linux/rpm
# rpm -Uvh tde_for_pg96-1.3.0-0.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tde_for_pg96-1.3.0-0.el7         ################################# [100%]
INFO: Transparent Data Encryption for PostgreSQL 9.6
      was installed successfully.
HINT: To complete validation of transparent data encryption feature,
      please add "/opt/nec/tdeforpg96/lib/data_encryption.so" to
      'shared_preload_libraries' parameter in 'postgresql.conf' file
      and require a PostgreSQL server restart to take effect.
```

注

インストール先ディレクトリを指定してインストールした場合

RPM パッケージを任意のディレクトリにインストールしている場合は--prefix でインストール先ディレクトリを指定します。--prefix を指定せずに再インストールした場合、デフォルトディレクトリ（/opt/nec）にインストールされます。

次の例では/cal/nec にインストールされている Transparent Data Encryption for PostgreSQL に対して再インストールしています。

*2　pg_ctl コマンドの詳細な利用方法は PostgreSQL マニュアルをご確認ください。

```
# rpm -Uvh --replacepkgs --prefix /cal/nec tde_for_pg96-1.3.0-0.el7.x86_64.rpm
```

5. OS のデータベース管理者ユーザ（一般的に postgres）でログインし、透過的暗号化機能を利用しているデータベースを起動します。

```
$ pg_ctl start
```

6. 「6.7　透過的暗号化機能の再有効化（22 ページ）」を参考に透過的暗号化機能を再有効します。

   アップグレードを伴う対話型の再有効化の場合、次の確認メッセージが出力されますので、問題がない場合は「Yes」を入力します。次の例では V1.2.0 から V1.3.0 へのアップグレードを実施しています。

```
WARN: Are you sure you want to upgrade transparent data encryption feature fro
m "Enterprise Edition 1.2.0.0" to "Enterprise Edition 1.3.0.0"?
Please input [Yes/No] > Yes
```

   非対話型でアップグレードを伴う再有効化を実施した場合、確認メッセージは出力されません。次の例では V1.2.0 から V1.3.0 へのアップグレードを実施しています。

```
INFO: Being upgrade transparent data encryption feature from "Enterprise Editi
on 1.2.0.0" to "Enterprise Edition 1.3.0.0".
```

   入力した情報に問題が無ければ以下のようなメッセージが表示され、指定したデータベースに対して透過的暗号化機能が再有効化されます。また、セキュリティ管理ユーザの接続情報が記載された設定ファイルが作成されます（本手順では/opt/nec/tdeforpg96/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザは、このファイルを透過的暗号化機能コマンド pgtde 実行時の接続情報ファイルとして使用することが可能です。

```
INFO: Transparent data encryption feature has been activated
PostgreSQL connection info for security user has created:  /opt/nec/tdeforpg96
/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQL s
ecurity user
```

7. 旧バージョンでよりセキュアな運用のための設定を行っていた場合、再度「4.5　よりセキュアな運用のための設定（13 ページ）」を参考に設定を行います。

7.3　透過的暗号化機能が有効な PostgreSQL のメジャーバージョンアップグレード

アップグレードを行う前に pg_dumpall を使用してデータベース全体のバックアップを取得していただくことを推奨いたします。

透過的暗号化機能が有効となっている PostgreSQL のメジャーバージョンアップグレードを行う場合、以下の手順に従って PostgreSQL のメジャーバージョンアップグレードを行ってください。透過的暗号化機能を利用しているデータベースは PostgreSQL の標準機能である pg_dump/pg_restore コマンドを利用します。透過的暗号化機能を利用するデータベースを含んだ状態で pg_dumpall コマンドや pg_upgrade コマンドを使ってデータベースクラスタ全体を移行する方法はサポートしていません。本節の手順を利用することで、Transparent Data Encryption for PostgreSQL と PostgreSQL のメジャーバージョンアップグレードを同時に行うことができます。本節の例では、Transparent Data Encryption for PostgreSQL V 1.3.0 がセットアップされた PostgreSQL 9.6 を Transparent Data Encryption for PostgreSQL V 2.1.1 がセットアップされた PostgreSQL 13 にアップグレードします。また、手順では Transparent Data Encryption for PostgreSQL が`/opt/nec`（デフォルト）にインストールされていることとし、透過的暗号化機能コマンド（`pgtde`）のデータベース接続ファイルとして`/opt/nec/tdeforpg96/conf/pgtde_secuser.properties` を使用します。

注

PostgreSQL 9.6.23 まで動作確認を行っておりますが、それ以降の PostgreSQL バージョンにて手順が失敗する場合は別途お問合せください。

1. 透過的暗号化機能を利用しているデータベースが AWS KMS モード（Transparent Data Encryption for PostgreSQL V1.1.4 までの AWS KMS 管理方式）を利用している場合は、`pgtde -m switch --standard-tde`[*3] コマンドを利用して標準 TDE モード（Transparent Data Encryption for PostgreSQL V1.1.4 までのローカル鍵管理方式）に変更します。

```
# /opt/nec/tdeforpg96/bin/pgtde -m switch --standard-tde \
> -conf /opt/nec/tdeforpg96/conf/pgtde_secuser.properties
Enter AWS Access Key ID: ******
Enter AWS Secret Access Key: ******
Enter current cipher key: *******
Are you sure you want to switch key management mode of "tdedb"(DATABASE) to "S
tandard TDE mode" ? (Press Y(y) key to execute): Y
Switch key management mode to standard TDE mode is successfully.

Data key file path: /tmp/datakey_20180309110430
You need to manage this key to using in case of changing key management mode.
INFO: <I002> For security purposes please delete this data key file after used
.
```

ヒント

Transparent Data Encryption for PostgreSQL V1.1.4 以前は、`pgtde -m switch --tde-only` コマンドを利用します。

```
# /opt/nec/tdeforpg96/bin/pgtde -m switch --tde-only \
> -conf /opt/nec/tdeforpg96/conf/pgtde_secuser.properties
Enter AWS Access Key ID: ******
Enter AWS Secret Access Key: ******
Enter current cipher key: *******
Are you sure you want to switch key management mode of "tdedb"(DATABASE) to "TDE lo
```

```
cal mode" ? (Press Y(y) key to execute): Y
Switch key management mode to TDE local mode is successfully.

Data key: ******
```

2. 対象のデータベースが複数の暗号鍵を利用している場合、次の手順でユーザデータを再暗号化します。

注

最新の暗号鍵によるデータ再暗号化は負荷状況に注意する必要があります。

a. 対象のデータベースに接続し、次のクエリを発行した結果が「1」でなければ次の手順を実施します。

```
=# SELECT count(*) FROM public.cipher_key_table;
```

b. pgtde -m cipher --reset コマンドを使用してデータの再暗号化 を実施します。

```
# /opt/nec/tdeforpg96/bin/pgtde -m cipher --reset \
> -conf /opt/nec/tdeforpg96/conf/pgtde_secuser.properties
Enter current data key:******
Are you sure you want to Reencrypt "tdedb"(DATABASE) with Interval="0" an
d Reset="true"? (Press Y(y) key to execute): Y
All data in tdedb are reencrypted
```

3. アップグレード対象のデータベースに対して pg_dump[*4] コマンドを実行します。透過的暗号化機能を利用するデータベースが複数存在する場合はデータベース毎に実施してください。なお、pg_dump 実行前に PGOPTIONS 環境変数で encrypt.enable パラメータを off にすることでユーザデータを暗号化したままの状態でバックアップすることが可能です。

   次の例では、ダンプファイル名として「pg_dump_tdedb.dump」、バックアップ対象のデータベースとして「tdedb」を指定しています。

```
$ PGOPTIONS="-c encrypt.enable=off" pg_dump \
-f pg_dump_tdedb.dump -Fc tdedb
```

4. ダンプファイルには旧バージョンの透過的暗号化機能オブジェクトも含まれているため、当該ダンプファイルをそのまま利用してデータベースのリストアを行うことはで

*3 モードの切り替え手順や pgtde コマンドの詳細は対象バージョンの『列単位暗号化 透過的暗号化機能 利用の手引き』をご確認ください。

*4 pg_dump コマンドの詳細な利用方法は PostgreSQL マニュアルをご確認ください。

きません。そのため、旧バージョンの透過的暗号化機能オブジェクトを除いたユーザデータのみをリストアする為のリストアリストを作成する必要があります。以下の手順に従って透過的暗号化機能を利用するデータベース毎のバックアップファイルからリストア対象ユーザデータオブジェクトファイルを作成し てください。

a. バックアップしたダンプファイル(本節の手順では pg_dump_tdedb.dump)のリストファイルを作成します。

次の例では、ダンプファイルとして前の手順で作成した「pg_dump_tdedb.dump」を、バック アップリストファイル名として「pg_dump_tdedb.list」を指定しています。

```
$ pg_restore -l pg_dump_tdedb.dump > pg_dump_tdedb.list
```

b. 透過的暗号化機能で使用しているオペレータのリストファイルを作成しま す。この手順は透過的暗号化機能を利用するデータベースに psql で接続します。

次の例では、透過的暗号化機能を利用するデータ ベースとして「tdedb」を、オペレータリストファイル名として「tdedb_operator.list」を指定しています。

```
$ psql -F ' ' -d tdedb \
<< EOF | grep -E [0-9]+ > tdedb_operator.list
\t
\a
SELECT a.oid,
       'OPERATOR',
       c.nspname ,
       regexp_replace(oprname, '\*','\\\*'),
       rolname
  FROM pg_operator a
  JOIN pg_authid b ON a.oprowner=b.oid
  JOIN pg_namespace c ON c.oid=a.oprnamespace
 WHERE (   a.oprleft IN (SELECT oid
                           FROM pg_type
                          WHERE typname LIKE 'encrypt_%')
        OR a.oprright IN (SELECT oid
                            FROM pg_type
                           WHERE typname LIKE 'encrypt_%'));
EOF
```

c. 作成したバックアップリストファイル（本節の手順では pg_dump_tdedb.list）とオペレータリストファイル（本節の手順では tdedb_operator.list）、そして Transparent Data Encryption for PostgreSQL Enterprise Edition インストール媒体に同梱されている透過的暗号化機能オブジェクトリストファイル (upgrade/installed_list_v1_3_0.txt)を使用して、ユーザデータリストファイルを作成します。

透過的暗号化機能オブジェクトリストファイルの命名規則は以下の通りですので、Transparent Data Encryption for PostgreSQL の旧バージョンとバージョンが合致する透過的暗号化機能オブジェクトリストファイルをご利用ください。

表 7-1　透過的暗号化機能オブジェクトリストファイルの命名規則

Edition	命名規則
Enterprise Edition	installed_list_vX_Y_Z.txt
Free Edition	installed_list_fe_VX_Y_Z_T.txt

X_Y_Z はアップグレード前の Transparent Data Encryption for PostgreSQL バージョンです。X.Y はメジャーバージョン、Z はマイナーバージョンを示します。T は Free Edition のみのリビジョン情報を示します。

次の例では、オペレータリストファイルとして前の手順で作成した「tdedb_operator.list」、バック アップリストファイルとして先ほど作成した「pg_dump_tdedb.list」、ユーザデータリストファイル名として「tdedb_user_data.list」を指定しています。

```
$ grep -v -f installed_list_v1_3_0.txt \
-f tdedb_operator.list pg_dump_tdedb.list > tdedb_user_data.list
```

注

複数の透過的暗号化機能を利用するデータベースがある場合はバック アップリストファイルとオペレータリストファイルの対応付けにご注意く ださい。

5. Transparent Data Encryption for PostgreSQL のアップグレードも同時に行っており、旧バージョンの Transparent Data Encryption for PostgreSQL が不要な場合、「アンインストール（18 ページ）」を参考にアンインストールします。アンインストールを行わない場合は、「6.2　透過的暗号化機能の無効化（18 ページ）」のみ実施します。
6. ここまでの手順が完了後、透過的暗号化機能を利用するデータベースを削除します。

```
$ dropdb tdedb
```

7. PostgreSQL のメジャーバージョンアップグレードを行います。バージョンアップ手順については本節下部の関連リンクをご確認ください。
8. メジャーアップグレード後の PostgreSQL で透過的暗号化機能を利用するデータベースを作成します。

```
$ createdb tdedb
```

9. 「第 4 章　新規セットアップ（7 ページ）」を参考にメジャーアップグレード後の PostgreSQL に透過的暗号機能をインストールします。
10. 透過的暗号化機能を利用するデータベースに対してバックアップファイルと作成したユーザデータリストファイルを使用し、pg_restore コマンドでリストアします。

    次の例では、バックアップファイルに「pg_dump_tdedb.dump」を、リストア対象のデータベースとして「tdedb」を、作成したユーザデータリストファイルには「tdedb_user_data.list」を指定します。

```
$ PGOPTIONS="-c encrypt.enable=off" pg_restore -d tdedb \
-L tdedb_user_data.list -e pg_dump_tdedb.dump
```

注

透過的暗号化機能を利用するデータベースが複数ある場合は、バックアップファイルとユーザデータリストファイルの対応付けにご注意ください。

11. リストアが成功しても、暗号鍵が登録されていないため、ユーザデータを復号して参照することはできません。そのため旧バージョンで利用していた最新の暗号鍵と同じ暗号鍵をリストアしたデータベースに登録する必要があります。また登録する暗号鍵は暗号化アルゴリズムも一致している必要がある点にご注意ください。

```
# /opt/nec/tdeforpg13/bin/pgtde -m regist \
-conf /opt/nec/tdeforpg13/conf/pgtde_secuser.properties
Key management mode is not yet set.
Please select key management mode:
1. Simple TDE mode.
2. Standard TDE mode.
3. AWS-KMS mode.
1
Enter new data key:
Retype new data key:
Select algorithm:
1. aes
2. bf
1
Are you sure you want to Regist new key to "tdedb"(DATABASE) with "aes" algori
thm? (Press Y(y) key to execute): Y
New key version 1 is registered to tdedb
```

関連リンク

PostgreSQL アップグレード手順

付録 A. セットアップ機能で出力されるエラーメッセージ

セットアップ機能で表示されるエラーメッセージについて説明します。

A.1 コマンドエラーメッセージ

セットアップ機能 `cipher_setup.sh` で表示されるエラーメッセージの一覧を下記に記載します。

表 A-1 Linux 版エラーメッセージ一覧

エラーメッセージ	対処方法
ERROR: Invalid menu number : <入力文字列>	メニューに表示されている項目番号を選択してください。
ERROR: You must be root to execute this command.	root ユーザで再度実行してください。
ERROR: The length of port number must not be zero	ポート番号空文字以外を入力してください。
ERROR: The length of user name must not be zero	ユーザ名は空文字以外を入力してください。
ERROR: Can not use template1 database	「template1」以外のデータベースを指定してください。
ERROR: The length of database name must not be zero	空文字以外のデータベース名を指定してください。
ERROR: must be superuser to execute this command	接続ユーザは PostgreSQL のスーパユーザを指定してください。
ERROR: There is not exist a definition-script : <ファイル名>	インストールしたファイル構成が破損している可能性があります。Transparent Data Encryption for PostgreSQL の再インストールを実行してください。
ERROR: Invalid input.	入力内容を確認して、正しい値を入力してください。
ERROR: input length must not be zero.	空文字は指定できません。
ERROR: Could not connect to the database	接続情報の内容を確認してください。
WARN: Transparent data encryption feature has not been activated yet	透過的暗号化機能が有効化されているデータベースに対して再実行してください。
WARN: Transparent data encryption function has already been activated	既に対象データベースは透過的暗号化機能が有効になっているため、有効化は不要です。
ERROR: Lock file does not exist. File name : ${INSTALLFILE}	内部エラーが発生しています。システム管理者に連絡を行ってください。
ERROR: Could not inactivate the transparent data encryption feature	透過的暗号化機能の無効化に失敗しました。出力されたエラーメッセージファイルを確認してください。
ERROR: Could not activate transparent data encryption feature	透過的暗号化機能の無効化に失敗しました。出力されたエラーメッセージファイルを確認してください。
ERROR: input user must not be super user	スーパユーザでないユーザを指定してください。
ERROR: Retype password does not match.	パスワードが一致しません。正しいパスワードを入力してください。
ERROR: Could not access to DB.	データベースに接続できませんでした。接続情報の内容を確認してください。
ERROR: Invalid arguments.	コマンドパラメータが間違っています。表示された Usage に従い、再実行してください。
ERROR: Could not read config file: ＜ファイル名＞	非対話型実行で、コンフィグファイルが読み込めません。コンフィグファイルが指定した場所に存在するか、または権限の設定が正しいか確認してください。
ERROR: Setting of ＜設定項目＞ is not found.	非対話型実行で、コンフィグファイルの設定項目が見つかりません。コンフィグファイルの設定項目を正しく記載しているか確認してください。

エラーメッセージ	対処方法
ERROR: Security user must not be super user	非対話型実行で、セキュリティ管理ユーザは非スーパユーザを指定してください。
ERROR: Security user could not access to DB.	非対話型実行で、セキュリティ管理ユーザでユーザデータベースに接続できませんでした。接続情報の内容を確認してください。
ERROR: Transparent data encryption feature does not support downgrade version (from "Enterprise Edition ＜現在のバージョン＞" to "Enterprise Edition ＜新しいバージョン＞").	アップグレードしてください。(Transparent Data Encryption for PostgreSQL はメジャーバージョン、マイナーバージョンともにダウングレードはできません。)

付録 B. ディレクトリ・ファイル構成

表 B-1 Linux ディレクトリ・ファイル構成

ディレクトリ・ファイル構成			説明
tdeforpg<XX>/ XX は PostgreSQL メジャーバージョン	bin/	cipher_setup.sh	透過的暗号化機能セットアップスクリプト
		pgtde	暗号化機能実行コマンド
	conf/	aws_info.properties	暗号鍵を復号化するための AWS アクセス情報（AccessKey、SecretAccessKey）を記入するファイル
		kms_info.properties	AWS KMS へのアクセス時の proxy 情報、および KMS サーバーの URL 情報
	lib/	libpgcrypto<XX>.so	透過的暗号化機能用ライブラリ
		data_encryption.so	透過的暗号化機能用ライブラリ
		data_encryption.so.X.Y.Z-N X.Y はメジャーバージョン、Z はマイナーバージョン、N がビルド番号を示します	透過的暗号化機能用ライブラリ
	lib/tool	libpq.so.5.<XX>	PostgreSQL 接続用ライブラリ
		libpq.so.5	PostgreSQL 接続用ライブラリ
		psql	内部コマンド発行用 PostgreSQL クライアントプログラム
	lib/init		透過的暗号鍵機能内部実行スクリプト群
	lib/conf		透過的暗号化機能内部設定ファイル群
	lib/prop		透過的暗号化機能定義ファイル群
	lib/jar		透過的暗号化機能実行基盤ファイル群
	jre/		透過的暗号化機能用 Java 実行環境
	template/	cipher_setup.conf.template	透過的暗号化機能セットアップスクリプト用設定ファイルのテンプレート
		pgtde.properties.template	透過的暗号化機能コマンド pgtde 用設定ファイルのテンプレート
	log/		Transparent Data Encryption for PostgreSQL 用のデフォルトログ出力先
	sys/		透過的暗号化機能システム管理用ディレクトリ
	LICENSE		利用しているオープンソースライセンスについて

付録 C. 改訂履歴

本マニュアルの改訂履歴は以下のとおりです。

表 C-1 改訂履歴一覧

版数	発行日	改訂履歴
第一版	2015 年 6 月	• 初版作成
第二版	2015 年 11 月	• 第二版改訂
第三版	2017 年 2 月	• rpm インストール時に LANG=ja_JP.UTF8 か C に指定する(第 3 章 セットアップの方法) • cipher_setup.sh の非対話型でのセットアップ手順の追記(第 3 章セットアップの方法) • エラーメッセージを追加(エラーメッセージ)
第四版	2017 年 12 月	• 実行コマンドや実行例を修正(全体) • PostgreSQL 9.6 に対応したことを追記(第 2 章セットアップの前に) • AES-NI に対応したことを追記(第 2 章セットアップの前に) • 透過的暗号化機能のマイナーバージョンアップグレード手順の追記(第 3 章セットアップの方法) • 透過的暗号化機能はダウングレード(メジャー、マイナーともに)できないことを追記(エラーメッセージ、注意事項) • 複数 DB インスタンス(DB クラスタ)でのデータベース名の重複について注意事項を追記(注意事項) • 同一 DB インスタンス(DB クラスタ)内で複数バージョンの Transparent Data Encryption for PostgreSQL を構成することについて注意事項を追記(注意事項)
第五版	2018 年 4 月	• 表記規則の追記 • 最新の情報の入手先の追記 • 章構成の変更 - インストールの概要を追記 - セットアップの方法として 1 つの章にまとめられていた新規インストール、再インストール、アップグレードインストールなどを章ごとに分割 - エラーメッセージを付録に変更 - 注意事項の各項目を関連する箇所に記載。併せて注意事項の章を廃止 - 禁則文字の章の内容を「3.2 透過的暗号化機能をセットアップするために必要な情報（4 ページ）」に移動。併せて禁則文字の章を廃止 - ライセンスの章を『透過的暗号化機能 利用の手引き』に移動 • 実行コマンドや実行例を修正（全体） • メンテナンス機能の廃止に伴い関連する記述を削除（全体） • PostgreSQL 10 に対応したことを追記(第 3 章動作環境の確認とインストール前の準備) • Windows プラットフォームに対応したことを追記(第 3 章動作環境の確認とインストール前の準備) • 暗号化データ型の整数型に対応したことを追記（第 1 章はじめに）

<table>
<tr><th>版数</th><th>発行日</th><th>改訂履歴</th></tr>
<tr><td></td><td></td><td>• 用語の変更（全体）<table><tr><th>V1.1.4 まで</th><th>V1.2.0 以降</th></tr><tr><td>暗号鍵管理方式</td><td>モード</td></tr><tr><td>AWS KMS 管理方式</td><td>AWS KMS モード</td></tr><tr><td>ローカル鍵管理方式</td><td>標準 TDE モード</td></tr><tr><td>－</td><td>簡易 TDE モード</td></tr></table></td></tr>
<tr><td>第六版</td><td>2020 年 9 月</td><td>• 実行コマンドや実行例を修正（全体）
• OpenSSL 1.1.1 系（RHEL8）を追記(第 3 章動作環境の確認とインストール前の準備)
• PostgreSQL 11 に対応したことを追記(第 3 章動作環境の確認とインストール前の準備)
• Red Hat Enterprise Linux 8.1 以上に対応したことを追記(第 3 章動作環境の確認とインストール前の準備)</td></tr>
<tr><td>第七版</td><td>2021 年 4 月</td><td>• 実行コマンドや実行例を修正（全体）
• PostgreSQL 12 に対応したことを追記(第 3 章動作環境の確認とインストール前の準備)</td></tr>
<tr><td>第八版</td><td>2022 年 4 月</td><td>• 実行コマンドや実行例を修正（全体）
• PostgreSQL 13 に対応したことを追記(第 3 章動作環境の確認とインストール前の準備)</td></tr>
</table>

マニュアルコメント用紙

読者各位

説明書に関するご意見、ご要望、内容不明確な部分について具体的にご記入のうえ、販売店または、当社担当営業、担当ＳＥにお渡しください。		お客様ご提出日	年　月　日
		〒 ご住所	〒
マニュアルコード	ＯＳＳＤＢＴＤＥ０１－０８	貴社名 所 属	
マニュアル名	Transparent Data Encryption for PostgreSQL 列単位暗号化 セットアップカード（Linux 版）	お名前	

項番	ﾍﾟｰｼﾞ	行・図番	指摘区分	指 摘 内 容	添付資料
1					

備考　指摘区分　１：誤り　２：誤字・脱字　３：難解　９：ご要望

ご協力ありがとうございます。

（注意）販売店員または、当社営業部員、ＳＥは、すみやかに所定の手続きに従ってマニュアル担当までお送りください。（メール：２２－Ａ０７０３）

なお、ＮＥＣメールがない場合は、下記まで郵送してください。

〒２１１－８６６６ 神奈川県川崎市中原区下沼部１７５３

日本電気(株) AI プラットフォーム事業部 SDP グループ宛

販売店員 営業部員 ＳＥ記入	販売店名 または 所属名		担当		ﾒｰﾙ TEL	

ＮＥＣ

Transparent Data Encryption for PostgreSQL Enterprise Edition
列単位暗号化 セットアップカード
（Linux 版）

ＯＳＳＤＢＴＤＥ０１－０８

2022 年 04 月 第八版 発行

日本電気株式会社