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

（Linux 版）

ＯＳＳＤＢＴＤＥ０７－０７

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『行単位暗号化 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-H204-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU(1 年間)	64 ビット
UL4027-H205-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU 追加(1 年間)	64 ビット
UL4027-H206-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 待機用 1CPU(1 年間)	64 ビット
UL4027-H214-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU(3 年間)	64 ビット
UL4027-H215-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU 追加(3 年間)	64 ビット
UL4027-H216-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 待機用 1CPU(3 年間)	64 ビット
UL4027-J204-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU(1 年間)(時間延長保守)	64 ビット
UL4027-J205-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU 追加(1 年間)(時間延長保守)	64 ビット
UL4027-J206-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 待機用 1CPU(1 年間)(時間延長保守)	64 ビット
UL4027-J214-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU(3 年間)(時間延長保守)	64 ビット
UL4027-J215-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 1CPU 追加(3 年間)(時間延長保守)	64 ビット
UL4027-J216-I	Transparent Data Encryption for PostgreSQL Enterprise Edition V2.4 Linux 版 待機用 1CPU(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. AlmaLinux は、The AlmaLinux OS Foundation の商標です。
6. Rocky Linux は、Rocky Enterprise Software Foundation の商標または登録商標です。
7. Amazon Linux 2 および Amazon Linux 2023 は、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です。
8. 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 の管理者ユーザーで発行するコマンドを示すプロンプトです	# `./cipher_setup.sh`
$	OS の一般ユーザー（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	tdeforpg2_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　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 があります。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　機能/サービス

機能/サービス	Enterprise Edition for Linux	Enterprise Edition for Windows
Transparent Data Encryption 機能		
行単位の暗号化機能	○	○
鍵の更新、バージョン管理機能	○	○
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	○	○

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 5 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。

- 高可用性構成（HA 構成）への新規セットアップ

  CLUSTERPRO X の高可用性構成（HA 構成）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 2 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。

- 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 13 から PostgreSQL 16 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 4 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。

- 高可用性構成（HA 構成）からのアンインストール

  CLUSTERPRO X の高可用性構成（HA 構成）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。「3.4.1.2　ソフトウェア要件（7 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。

3.2　JDK のインストール

Transparent Data Encryption for PostgreSQL を利用するためには、事前に JDK Version 8 以降の長期サポート（LTS）リリースされているものをインストールしておく必要があります。

JDK がインストール済で `alternatives` に登録されているか確認する場合は、`alternatives --display java` を使用してください。`alternatives` に `java` が登録済の場合は、以降の手順は不要です。

1. OpenJDK のバイナリファイルをダウンロードし、解凍します。

   次の例では、任意の場所にダウンロードした OpenJDK （OpenJDK Runtime Environment OpenLogic-OpenJDK version 17.0.7）のバイナリファイルを解凍します。

```
# tar xvzf openlogic-openjdk-17.0.7+7-linux-x64.tar.gz
```

   注

   JDK のインストールは、`yum` コマンドや rpm 形式のファイルから実施することも可能です。

2. OpenJDK のバイナリファイルを解凍後、任意のフォルダへ移動します。

   次の例では、解凍したバイナリファイルを`/usr/lib/java/`へ移動します。

```
# mv openlogic-openjdk-17.0.7+7-linux-x64 /usr/lib/java/
```

3. 利用する `java` コマンドを `alternatives` に登録します。

   次の例では、`java` コマンドを `alternatives --install` を使用して登録します。

```
# alternatives --install /usr/bin/java java /usr/lib/java/openlogic-openjdk-17.
0.7+7-linux-x64/bin/java 1
```

4. インストールした JDK が正しく呼び出されているか確認します。

   次の例では、java -version を実行してインストールした JDK のバージョンを確認します。

```
# java -version
openjdk version "17.0.7" 2023-04-18
OpenJDK Runtime Environment OpenLogic-OpenJDK (build 17.0.7+7-adhoc.root.jdk17u)
OpenJDK 64-Bit Server VM OpenLogic-OpenJDK (build 17.0.7+7-adhoc.root.jdk17u, mixed mode, sharing)
```

3.3　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。
スーパーユーザー名	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ーユーザーです。
スーパーユーザーのパスワード	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ーユーザーのパスワードです。
セキュリティ管理ユーザー名	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ーです。
セキュリティ管理ユーザーのパスワード	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ーのパスワードです。

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。

- ユーザー名

{「!」,「'」}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	×	×	×	×	×	×	×	×	×	×
ユーザー名	×	×		×	×					
パスワード名	×	×								

×…禁則文字として扱われる文字・文字列

3.4　インストール要件の確認

3.4.1　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。

3.4.1.1　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。

表 3-3　データベースサーバー側のハードウェア要件

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 13 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 V2.1.0 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 Red Hat Enterprise Linux 互換 OS では通常 OpenSSL 1.0.2 系（RHEL7）または OpenSSL 1.1.1 系（RHEL8）、OpenSSL 3.0 系（RHEL9）がインストールされています。

3.4.1.2　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。

なお、オペレーティングシステム（Linux）につきましては Red Hat Enterprise Linux 互換 OS（Oracle Linux、AlmaLinux、Rocky Linux、Amazon Linux）も含まれます。

表 3-4　データベースサーバー側のソフトウェア要件（Linux 版）

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	Red Hat Enterprise Linux 8.1 以上	Red Hat Enterprise Linux 9.0 以上
13	○	○	○
14	○	○	○
15	○	○	○
16	×	○	○
必要パッケージ（Linux）	zlib.x86_64 glibc.x86_64 JDK Version 8 以降の長期サポート（LTS）リリースされているもの「表 3-5　動作確認済 JDK のバージョン一覧（2024 年 11 月時点）（7 ページ）」		

注

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

動作確認済 JDK のバージョンは下記になります。

表 3-5　動作確認済 JDK のバージョン一覧（2024 年 11 月時点）

JDK の種類	動作確認済バージョン
Oracle Java SE	8, 11, 17, 20, 21
OpenJDK	8, 11, 17, 21

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 （16 ページ)」、「4.7　高可用性構成（HA 構成）への新規セットアップ （19 ページ)」の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ーがインストールされた端末リモートコンピューターからも実行が可能です。リモートコンピューターから鍵管理機能を利用する場合、リモートコンピューターにも 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（9 ページ）」
2. 「4.3　透過的暗号化機能の有効化（10 ページ）」
3. 「4.4　postgresql.conf の編集（13 ページ）」
4. 「4.5　よりセキュアな運用のための設定（14 ページ）」

   ヒント

   PostgreSQL のユーザー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『行単位暗号化 透過的暗号化機能 利用の手引』をご確認ください。

5. 利用するモードの検討
   - 簡易 TDE モード
   - 標準 TDE 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、非サポート)
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. 暗号化対象のユーザーテーブルを作成
11. 暗号化対象のユーザーテーブルに対するデータ操作

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 tdeforpg2_pg16-2.4.0-0.el8.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tdeforpg2_pg16-2.4.0-0.el8         ################################# [100
%]
INFO: Transparent Data Encryption for PostgreSQL 16
      was installed successfully.
HINT: To complete validation of transparent data encryption feature,
      please add "tdeforpg2.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。

tdeforpg2_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。

- 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 、Red Hat Enterprise Linux 9 は el9 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 tdeforpg2_pg16-2.4.0-0.el8.x86_64.rpm
```

4.3　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。

- 「4.3.1　透過的暗号化機能を対話型で有効化する方法（10 ページ）」
- 「4.3.2　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）」

4.3.1　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 16 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/tdeforpg2_pg16/bin/cipher_setup.sh
   Transparent data encryption feature setup script
   ```

2. 「3.3　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（5 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ーを入力します。

   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ー名が PostgreSQL に存在しない場合、新規に PostgreSQL ユーザーを作成します。この際にセキュリティ管理ユーザーは、 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ーとして PostgreSQL のスーパーユーザー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	スーパーユーザー名
Please enter password for authentication	スーパーユーザーのパスワード
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ー名
Please enter password for database user secman	セキュリティ管理ユーザーのパスワード
Retype password for database user secman	セキュリティ管理ユーザーのパスワード(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ー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/tdeforpg2_pg16/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザーは、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
2_pg16/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQ
L security user
```

4.3.2　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 16 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/tdeforpg2_pg16/template/cipher_setup.conf.template \
/opt/nec/tdeforpg2_pg16/conf/cipher_setup.conf
```

b. 先ほどの手順で作成した cipher_setup.conf を「3.3　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（5 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	スーパーユーザー名
connect_db_password	スーパーユーザーのパスワード
security_db_user	セキュリティ管理ユーザー名
security_db_password	セキュリティ管理ユーザーのパスワード

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ー名が PostgreSQL に存在しない場合、新規に PostgreSQL ユーザーを作成します。この際にセキュリティ管理ユーザーは、 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ーとして PostgreSQL のスーパーユーザーを指定することはできません。

```
# /opt/nec/tdeforpg2_pg16/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/tdeforpg2_pg16/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ー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/tdeforpg2_pg16/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザーは、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/tdeforpg2_pg16/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQL security user
```

4.4　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ー（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 tdeforpg2.so を設定します。

   [postgresql.conf 設定例]

   ```
   shared_preload_libraries='tdeforpg2.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
   ```

*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。

4.5　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 ユーザー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 ユーザーおよび OS グループを作成します。

   それぞれの OS ユーザーが適切な PostgreSQL ユーザーを使用するような運用方針を策定する必要があります。作成するユーザーと対応する PostgreSQL ユーザー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 ユーザー	OS グループ	役割	使用可能な PostgreSQL ユーザー
データベース管理者	透過的暗号化機能管理グループ	PostgreSQL 起動ユーザーであり、PostgreSQL に対する全権限を持つユーザー。	スーパーユーザー
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ー	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ー
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ーデータに対する暗号化・復号権限を持つユーザー	透過的暗号化機能を利用するユーザーデータにアクセスできる一般ユーザー

   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、作成したユーザー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/pgtde_secuser.properties	セキュリティ管理者
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）

対象ファイル	所有者
lib/jar/pgtde_regist.jar	セキュリティ管理者

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 16 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/tdeforpg2_pg16/
# 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
```

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ー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『行単位暗号化 透過的暗号化機能 利用の手引』をご確認ください)

表 4-5　モード毎実行可能ユーザー一覧

各-m オプション	実行可能ユーザー
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ーは、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ーに 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ー「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ーに透過的暗号化機能のセットアップで作成したユーザー以外の一般ユーザーを割り当てる場合、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 一般ユーザー「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 EXECUTE ON FUNCTION cipher_key_backup() TO secuser;
```

4. 暗号化対象テーブルの所有者は、アプリケーション管理者（アプリケーション開発者）「apuser」へ変更します。暗号化対象テーブルの所有者がアプリケーション管理者（アプリケーション開発者）「apuser」でない場合、最新の暗号鍵による再暗号化(-m cipher)はエラーになります。

   次の例では、暗号化対象テーブル「Employee」の所有者をアプリケーション管理者（アプリケーション開発者）「apuser」へ変更します。

```
=# ALTER TABLE Employee OWNER TO apuser;
```

注

暗号化対象テーブルは、アプリケーション管理者（アプリケーション開発者）「apuser」で作成して使用することも可能です。

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
	プライマリサーバー	スタンバイサーバー	
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）（17 ページ）」
6		透過的暗号化機能のファイル配置	「4.6.2　透過的暗号化機能のファイル配置（手順 6）（17 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）（18 ページ）」
8	postgresql.conf の編集		「4.6.4　postgresql.conf の編集（手順 8）（18 ページ）」

手順	作業項目		参照先
	プライマリサーバー	スタンバイサーバー	
9	よりセキュアな運用のための設定		「4.6.5　よりセキュアな運用のための設定（手順 9）（18 ページ）」

関連リンク

PostgreSQL のインストール（PostgreSQL の Windows インストーラー、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（9 ページ）」を参考にプライマリサーバーとスタンバイサーバー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、プライマリサーバーとスタンバイサーバー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　透過的暗号化機能のファイル配置（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、透過的暗号化機能で使用するファイルをスタンバイサーバー内に配置してください。

次の例では、PostgreSQL 16 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。

1. PostgreSQL の<SHAREDIR>のパスを確認します。

   [<SHAREDIR>のパス確認例]

```
$ pg_config --sharedir
/usr/pgsql-16/share
```

2. <SHAREDIR>に tdeforpg2--2.1.sql をコピーします。

   [tdeforpg2--2.1.sql のコピー例]

```
$ cp -pf "/opt/nec/tdeforpg2_pg16/lib/tdeforpg2--2.1.sql" "/usr/pgsql-16/share
/extension/tdeforpg2--2.1.sql"
```

注

Transparent Data Encryption for PostgreSQL V2.1.1 以降は、他の SQL ファイル（tdeforpg2--*.sql）も同様にコピーします。

3. <SHAREDIR>に tdeforpg2.control をコピーします。

   [tdeforpg2--2.1.control のコピー例]

```
$ cp -pf "/opt/nec/tdeforpg2_pg16/lib/tdeforpg2.control" "/usr/pgsql-16/share/extension/tdeforpg2.control"
```

4. PostgreSQL の<PKGLIBDIR>のパスを確認します。

   [<PKGLIBDIR>のパス確認例]

```
$ pg_config --pkglibdir
/usr/pgsql-16/lib
```

5. <PKGLIBDIR>に tdeforpg2.so へのシンボリックリンクを作成します。

   [tdeforpg2.so のシンボリックリンク例]

```
$ ln -sf "/opt/nec/tdeforpg2_pg16/lib/tdeforpg2.so" "/usr/pgsql-16/lib/tdeforpg2.so"
```

4.6.3　透過的暗号化機能の有効化（手順 7）

「4.3　透過的暗号化機能の有効化（10 ページ）」を参考にプライマリサーバーのみ透過的暗号化機能を有効化してください。

4.6.4　postgresql.conf の編集（手順 8）

ストリーミングレプリケーションを利用する場合は、「4.4　postgresql.conf の編集（13 ページ）」を参考にプライマリサーバーとスタンバイサーバーの両方の `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 `tdeforpg2.so` を設定してください。

4.6.5　よりセキュアな運用のための設定（手順 9）

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（14 ページ）」を参考にプライマリサーバーとスタンバイサーバーの両方を同一の構成となるよう設定してください。

4.7　高可用性構成（HA 構成）への新規セットアップ

CLUSTERPRO X の高可用性構成（HA 構成）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 1～6 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 4-7　インストール時の手順要否

手順	作業項目		参照先
	現用系サーバー	待機系サーバー	
1	CLUSTERPRO のインストール、設定		関連リンク参照
2	PostgreSQL のインストール		関連リンク参照
3	PostgreSQL の設定		関連リンク参照
4		PostgreSQL の設定	関連リンク参照
5	CLUSTERPRO の EXEC リソースの設定		関連リンク参照
6	CLUSTERPRO の監視リソースの設定		関連リンク参照
7	RPM パッケージのインストール		「4.7.1　RPM パッケージのインストール（手順 7）（19 ページ）」
8		透過的暗号化機能のファイル配置	「4.7.2　透過的暗号化機能のファイル配置（手順 8）（20 ページ）」
9	透過的暗号化機能の有効化		「4.7.3　透過的暗号化機能の有効化（手順 9）（20 ページ）」
10	postgresql.conf の編集		「4.7.4　postgresql.conf の編集（手順 10）（20 ページ）」
11	よりセキュアな運用のための設定		「4.7.5　よりセキュアな運用のための設定（手順 11）（20 ページ）」

関連リンク

CLUSTERPRO のインストール、設定（CLUSTERPRO X システム構築ガイド URL https://jpn.nec.com/clusterpro/clpx/manual.html）

PostgreSQL のインストール（PostgreSQL の Windows インストーラー、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）

PostgreSQL の設定、CLUSTERPRO の EXEC リソースの設定、監視リソースの設定（CLUSTERPRO® X for Linux PP ガイド(PostgreSQL) URL https://jpn.nec.com/clusterpro/clpx/doc/guide/HOWTO_PostgreSQL_Linux_JP_03.pdf）

4.7.1　RPM パッケージのインストール（手順 7）

CLUSTERPRO X の高可用性構成（HA 構成）を利用する場合は、「4.2　RPM パッケージのインストール（9 ページ）」を参考に現用系サーバーと待機系サーバー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、現用系サーバーと待機系サーバーを同じディレクトリパスに統一する必要があります。

4.7.2　透過的暗号化機能のファイル配置（手順 8）

CLUSTERPRO X の高可用性構成（HA 構成）を利用する場合は、「4.6.2　透過的暗号化機能のファイル配置（手順 6）（17 ページ）」を参考に待機系サーバー内に透過的暗号化機能で使用するファイルを配置してください。

4.7.3　透過的暗号化機能の有効化（手順 9）

「4.3　透過的暗号化機能の有効化（10 ページ）」を参考に現用系サーバーのみ透過的暗号化機能を有効化してください。

4.7.4　postgresql.conf の編集（手順 10）

CLUSTERPRO X の高可用性構成（HA 構成）を利用する場合は、「4.4　postgresql.conf の編集（13 ページ）」を参考に現用系サーバーと待機系サーバーの両方の 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 tdeforpg2.so を設定してください。

4.7.5　よりセキュアな運用のための設定（手順 11）

CLUSTERPRO X の高可用性構成（HA 構成）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（14 ページ）」を参考に現用系サーバーと待機系サーバー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 の管理者ユーザー（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 tdeforpg2_pg16-2.4.0-0.el8.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tdeforpg2_pg16-2.4.0-0.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、「6.7　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（25 ページ）」、「6.8　高可用性構成（HA 構成）からのアンインストール（25 ページ）」についても説明します。

重要

本製品をアンインストールするには、暗号化対象テーブルを全て削除する必要があります。

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（22 ページ）」
2. 「6.3　RPM パッケージのアンインストール（22 ページ）」
3. 「6.4　postgresql.conf の編集（23 ページ）」
4. 「6.5　ファイルの削除（24 ページ）」
5. 「6.6　インストールディレクトリの削除（24 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 つを実施します。

- 暗号化対象テーブルを手動で削除します。DROP EXTENSION 実行時に CASCADE オプションを指定することで、暗号化対象テーブルを一括削除することも可能です。
- データベースにスーパーユーザーで接続し、tdeforpg2 を DROP EXTENSION クエリでアンイ ンストールします。

```
=# DROP EXTENSION tdeforpg2;
DROP EXTENSION
```

注

DROP EXTENSION クエリ実施後は、暗号鍵情報を格納する cipher_key_table テーブルと key_management_table テーブルが削除されます。

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 tdeforpg2_pg16
/opt/nec/tdeforpg2_pg16
...
/opt/nec/tdeforpg2_pg16/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 tdeforpg2_pg16-2.4.0-0.el8.x86_64
INFO: Transparent Data Encryption for PostgreSQL 16
      was uninstalled successfully.
HINT: To complete invalidation of transparent data encryption feature,
      please remove "tdeforpg2.so" from
      'shared_preload_libraries' parameter in 'postgresql.conf'
      file and require a PostgreSQL server restart to take effect.
      In addition, please remove the following files.
        * <SHAREDIR>/extension/tdeforpg2.control
        * <SHAREDIR>/extension/tdeforpg2--2.1.sql
        * <PKGLIBDIR>/tdeforpg2.so
```

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ー（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 tdeforpg2.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
```

*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。

6.5　ファイルの削除

Transparent Data Encryption for PostgreSQL を今後利用しない場合、透過的暗号化機能を有効化した際に配置されたファイルを削除します。

1. tdeforpg2.control を削除します。

   次の例では、PostgreSQL 16 用の Transparent Data Encryption for PostgreSQL が/opt/nec（デフォルト）、PostgreSQL 16.1 を RPM パッケージ（デフォルト）でインストールされていることとします。

```
# ls -ld /usr/pgsql-16/share/extension/tdeforpg2.control
-rw-r--r--. 1 root root 160  6 月 30 19:35 /usr/pgsql-16/share/extension/tdefo
rpg2.control
# rm -i /usr/pgsql-16/share/extension/tdeforpg2.control
rm: 通常ファイル `/usr/pgsql-16/share/extension/tdeforpg2.control' を削除しますか?
 y
```

2. tdeforpg2--2.1.sql を削除します。

```
# ls -ld /usr/pgsql-16/share/extension/tdeforpg2--2.1.sql
-rw-r--r--. 1 root root 33235  6 月 30 19:35 /usr/pgsql-16/share/extension/tde
forpg2--2.1.sql
# rm -i /usr/pgsql-16/share/extension/tdeforpg2--2.1.sql
rm: 通常ファイル `/usr/pgsql-16/share/extension/tdeforpg2--2.1.sql' を削除しますか
? y
```

注

Transparent Data Encryption for PostgreSQL V2.1.1 以降は、他の SQL ファイル（tdeforpg2--*.sql）も同様に削除します。

3. tdeforpg2.so を削除します。

```
# ls -ld /usr/pgsql-16/lib/tdeforpg2.so
lrwxrwxrwx. 1 root root 41  6 月 30 19:35 /usr/pgsql-16/lib/tdeforpg2.so -> /o
pt/nec/tdeforpg2_pg16/lib/tdeforpg2.so
# rm -i /usr/pgsql-16/lib/tdeforpg2.so
rm: シンボリックリンク `/usr/pgsql-16/lib/tdeforpg2.so' を削除しますか? y
```

6.6　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。

   次の例では、PostgreSQL 16 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 tdeforpg2_pg16
drwxr-xr-x. 7 root root 82  6 月 30 19:35 tdeforpg2_pg16
# rm -rf tdeforpg2_pg16
```

6.7　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ーとスタンバイサーバー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 6-1　アンインストール時の手順要否

手順	作業項目		参照先
	プライマリサーバー	スタンバイサーバー	
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（22 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（22 ページ）」
3	postgresql.conf の編集		「6.4　postgresql.conf の編集（23 ページ）」
4	ファイルの削除		「6.5　ファイルの削除（24 ページ）」
5	インストールディレクトリの削除		「6.6　インストールディレクトリの削除（24 ページ）」

6.8　高可用性構成（HA 構成）からのアンインストール

CLUSTERPRO X の高可用性構成（HA 構成）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現用系サーバーと待機系サーバー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 6-2　アンインストール時の手順要否

手順	作業項目		参照先
	現用系サーバー	待機系サーバー	
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（22 ページ）」

手順	作業項目		参照先
	現用系サーバー	待機系サーバー	
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（22 ページ）」
3	postgresql.conf の編集		「6.4　postgresql.conf の編集（23 ページ）」
4	ファイルの削除		「6.5　ファイルの削除（24 ページ）」
5	インストールディレクトリの削除		「6.6　インストールディレクトリの削除（24 ページ）」

第 7 章
アップグレード

本章では下記 2 パターンのアップグレードについて説明します。

- 「7.1　Transparent Data Encryption for PostgreSQL のアップグレード（27 ページ）」
- 「7.2　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　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。

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. アップグレード対象のデータベースに対して `pg_dump`[*1] 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.cipherkey パラメータにデータ鍵の情報を設定することで復号したデータをバックアップすることが可能です（簡易 TDE モードの場合データ鍵の情報の設定は不要です）。バックアップしたデータは復号した状態のため、漏えいのリスクがありますのでご注意ください。

   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。

   - 標準 TDE モードの場合

```
$ PGOPTIONS="-c encrypt.cipherkey=key1234567890" pg_dump -f pg_dump_tdedb
.dump -Fc tdedb
```

- 簡易 TDE モードの場合

```
$ pg_dump -f pg_dump_tdedb.dump -Fc tdedb
```

2. 「6.2　透過的暗号化機能の無効化（22 ページ）」を参考に透過的暗号化機能を無効化します。
3. OS のデータベース管理者ユーザー（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
```

4. OS の管理者ユーザー（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
```

5. 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 tdeforpg2_pg13-2.4.0-0.el8.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:tdeforpg2_pg13-2.4.0-0.el8       ################################# [ 50%]
Cleaning up / removing...
   2:tdeforpg2_pg13-2.3.2-0.el8       ################################# [100%]
INFO: Transparent Data Encryption for PostgreSQL 13
      was updated successfully.
```

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

*1　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。

*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 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。

```
# rpm -Uvh --replacepkgs --prefix /cal/nec tdeforpg2_pg13-2.4.0-0.el8.x86_64.rpm
```

6. OS のデータベース管理者ユーザー（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
```

7. 「4.3　透過的暗号化機能の有効化（10 ページ）」を参考に透過的暗号化機能を有効化します。

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ー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/tdeforpg2_pg13/conf/pgtde_secuser.properties）。暗号鍵を管理する OS ユーザーは、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/tdeforpg2_
pg13/conf/pgtde_secuser.properties
Let use this conf file in [pgtde] command with option "-conf" for PostgreSQL s
ecurity user
```

8. 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（14 ページ）」を参考に設定を行います。

9. 透過的暗号化機能を利用するデータベースに対して暗号鍵の再登録を行います。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する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/tdeforpg2_pg13/bin/pgtde -m regist \
-conf /opt/nec/tdeforpg2_pg13/conf/pgtde_secuser.properties
Key management mode is not yet set.
Please select key management mode:
1. Simple TDE mode.
2. Standard TDE 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
```

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を使用し、pg_restore[*3] 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」を指定します（簡易 TDE モードの場合 encrypt.cipherkey パラメータは不要です）。

    - 標準 TDE モードの場合

    ```
    $ PGOPTIONS="-c encrypt.cipherkey=key1234567890" pg_restore -d tdedb -e p
    g_dump_tdedb.dump
    ```

    - 簡易 TDE モードの場合

    ```
    $ pg_restore -d tdedb -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の対応付けにご注意ください。

7.2　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 2.4.0 がセットアップされた PostgreSQL 13 を PostgreSQL 16 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

*3　pg_restore 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。

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/tdeforpg2_pg13/conf/pgtde_secuser.properties を使用します。

注

PostgreSQL 13.13 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. 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.cipherkey パラメータにデータ鍵の情報を設定することで復号したデータをバックアップすることが可能です（簡易 TDE モードの場合データ鍵の情報の設定は不要です）。バックアップしたデータは復号した状態のため、漏えいのリスクがありますのでご注意ください。

   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。

   - 標準 TDE モードの場合

   ```
   $ PGOPTIONS="-c encrypt.cipherkey=key1234567890" pg_dump -f pg_dump_tdedb.dump -Fc tdedb
   ```

   - 簡易 TDE モードの場合

   ```
   $ pg_dump -f pg_dump_tdedb.dump -Fc tdedb
   ```

2. 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（22 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（22 ページ）」のみ実施します。
3. 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
   ```

4. 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。
5. 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
   ```

*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。

6. 「第 4 章　新規セットアップ（8 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。

7. 透過的暗号化機能を利用するデータベースに対して暗号鍵の再登録を行います。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する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/tdeforpg2_pg16/bin/pgtde -m regist \
-conf /opt/nec/tdeforpg2_pg16/conf/pgtde_secuser.properties
Key management mode is not yet set.
Please select key management mode:
1. Simple TDE mode.
2. Standard TDE 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
```

8. 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を使用し、pg_restore[*5] 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」を指定します（簡易 TDE モードの場合 encrypt.cipherkey パラメータは不要です）。

   - 標準 TDE モードの場合

   ```
   $ PGOPTIONS="-c encrypt.cipherkey=key1234567890" pg_restore -d tdedb -e p
   g_dump_tdedb.dump
   ```

   - 簡易 TDE モードの場合

   ```
   $ pg_restore -d tdedb -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の対応付けにご注意ください。

関連リンク

PostgreSQL アップグレード手順

*5　pg_restore 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。

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: You must be root to execute this command.	root ユーザー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	ユーザー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	接続ユーザーは PostgreSQL のスーパーユーザー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 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。
WARN: Column-based encryption feature is already activated	既に列単位暗号化　透過的暗号化機能が有効になっています。
ERROR: input user must not be super user	スーパーユーザーでないユーザー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。
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ーは非スーパーユーザー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ーでユーザー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。)

エラーメッセージ	対処方法
ERROR:'PKGLIBDIR' was not found in pg_config.	PostgreSQL の動的ローディング可能なモジュールの場所を取得できませんでした。データベース管理者に連絡を行ってください。
ERROR:'SHAREDIR' was not found in pg_config.	PostgreSQL のアーキテクチャ非依存のサポートファイルの場所を取得できませんでした。データベース管理者に連絡を行ってください。
ERROR: Failed to copy sql file for tdeforpg2	sql ファイルのコピーに失敗しました。PP サポートサービスにご連絡ください。
ERROR: Failed to copy tdeforpg2.control	tdeforpg2.control のコピーに失敗しました。PP サポートサービスにご連絡ください。
ERROR: Failed to Link tdeforpg2.so	tdeeforpg2.so のハードリンクに失敗しました。PP サポートサービスにご連絡ください。

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
tdeforpg2_pg<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/		
	lib/	tdeforpg2.control	透過的暗号化機能用拡張ファイル
		tdeforpg2--2.1.sql	透過的暗号化機能内部実行スクリプト群
		tdeforpg2--2.1--2.1.1.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2--2.1.1--2.2.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2--2.2--2.3.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2--2.3--2.3.1.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2--2.3.1--2.3.2.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2--2.3.2--2.4.sql	透過的暗号化機能内部実行スクリプト（バージョン更新用）
		tdeforpg2.so	透過的暗号化機能用ライブラリ
		tdeforpg2.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.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/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
	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

ディレクトリ・ファイル構成			説明
	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	2021 年 4 月	• 初版作成
第二版	2022 年 4 月	• 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) • アップグレードを追記
第三版	2023 年 1 月	• 高可用性構成（HA 構成）への新規セットアップを追加（第 4 章 新規セットアップ） • 高可用性構成（HA 構成）からのアンインストールを追加（第 6 章 アンインストール）
第四版	2023 年 8 月	• 実行コマンドや実行例を修正（全体） • PostgreSQL 14、PostgreSQL 15 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） • 「JDK のインストール」を追加（第 3 章 動作環境の確認とインストール前の準備） • 「pgtde の編集」を削除（第 4 章 新規セットアップ） • 利用する暗号化アルゴリズム bf(Blowfish)は、非サポートであることを追記（第 4 章 新規セットアップ）
第五版	2023 年 11 月	• 実行コマンドや実行例を修正（全体） • 表 B-1 Linux ディレクトリ・ファイル構成 を更新（付録 B. ディレクトリ・ファイル構成）
第六版	2023 年 12 月	• 実行コマンドや実行例を修正（全体） • 表 B-1 Linux ディレクトリ・ファイル構成 を更新（付録 B. ディレクトリ・ファイル構成）
第七版	2024 年 11 月	• 実行コマンドや実行例を修正（全体） • PostgreSQL 16 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 9.0 以上、Red Hat Enterprise Linux 互換 OS（Oracle Linux、AlmaLinux、Rocky Linux、Amazon Linux）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) • 表 B-1 Linux ディレクトリ・ファイル構成 を更新（付録 B. ディレクトリ・ファイル構成）

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	ＯＳＳＤＢＴＤＥ０７－０７	貴社名 所 属	
マニュアル名	Transparent Data Encryption for PostgreSQL 行単位暗号化 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－Ａ０７０４）

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) プラットフォーム・テクノロジーサービス事業部門 データ基盤サービス統括部 カスタマーサクセスグループ宛

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
行単位暗号化 セットアップカード
（Linux 版）

ＯＳＳＤＢＴＤＥ０７－０７

2024 年 11 月 第七版 発行

日本電気株式会社