



EDB*Plus User's Guide

リリース **38**

2020 年 06 月 28 日

目次

第 1 章	EDB*Plus	2
第 2 章	インストール EDB*プラス	3
2.1	インストールの前提条件	3
2.2	RPM インストールの実行	5
2.2.1	RPM インストールの構成	7
2.3	グラフィカルインストーラーの使用	9
第 3 章	EDB*Plus の使用	14
第 4 章	SecureSocketsLayer (SSL) 接続の使用	18
4.1	AdvancedServer での SSL の構成	19
4.1.1	EDB*Plus クライアントの SSL の構成	22
4.1.2	EDB*Plus と AdvancedServer データベース間の SSL 接続の要求	25
第 5 章	コマンドの要約	28
5.1	ACCEPT	28
5.2	APPEND	29
5.3	CHANGE	29
5.4	CLEAR	30
5.5	COLUMN	30
5.6	CONNECT	34
5.7	DEFINE	35
5.8	DEL	35

5.9	DESCRIBE	37
5.10	DISCONNECT	37
5.11	EDIT	37
5.12	EXECUTE	38
5.13	EXIT	38
5.14	GET	39
5.15	HELP	39
5.16	HOST	40
5.17	INPUT	40
5.18	LIST	40
5.19	PASSWORD	41
5.20	PAUSE	42
5.21	PROMPT	42
5.22	QUIT	42
5.23	REMARK	42
5.24	SAVE	43
5.25	SET	44
5.26	SHOW	48
5.27	SPOOL	49
5.28	START	49
5.29	UNDEFINE	49
5.30	WHENEVER SQLERROR	49
第 6 章	結論	51
索引		53

このガイドでは、EDB*Plus を使用して AdvancedServer データベースに接続する方法について説明します。EDB*Plus は、次のことを可能にする SQL コマンドを受け入れる EDBPostgresAdvancedServer へのコマンドラインユーザーインターフェイスを提供します。

- 特定のデータベースオブジェクトをクエリする
- ストアドプロシージャを実行する
- SQL コマンドからの出力のフォーマット
- バッチスクリプトを実行する
- OS コマンドを実行する
- 記録出力

AdvancedServer でサポートされている機能の詳細については、次の URL にある AdvancedServer ガイドの完全なライブラリを参照してください。

<https://www.enterprisedb.com/edb-docs>

EDB*Plus

EDB*Plus は、EDBPostgresAdvancedServer へのコマンドラインユーザーインターフェイスを提供するユーティリティプログラムです。EDB*Plus は、SQL コマンド、SPL 匿名ブロック、および EDB*Plus コマンドを受け入れます。

EDB*Plus コマンドは、OracleSQL*Plus コマンドと互換性があり、次のようなさまざまな機能を提供します。

- 特定のデータベースオブジェクトのクエリ
- ストアドプロシージャの実行
- SQL コマンドからの出力のフォーマット
- バッチスクリプトの実行
- OS コマンドの実行
- 録音出力

第 2

インストール EDB*プラス

RPM インストーラーまたはグラフィカルインストーラーを使用して、EDB*Plus を AdvancedServer インストールに追加できます。

2.1 インストールの前提条件

EDB*Plus をインストールする前に、まず Java（バージョン 1.7 以降）をインストールする必要があります。Linux システムでは、yum パッケージマネージャーを使用して Java をインストールできます。ターミナルウィンドウを開き、スーパーユーザー権限を引き受け、次のように入力します。

- RHEL または CentOS7 の場合:

```
# yum install java
```

- RHEL または CentOS8 の場合:

```
# dnf install java
```

Windows を使用している場合、Java インストーラーと手順は次のサイトからオンラインで入手できます。

<http://www.java.com/en/download/manual.jsp>

EnterpriseDB リポジトリへのアクセスを許可する資格情報も必要です。資格情報のリクエストについては、次の URL にアクセスしてください。

<https://info.enterprisedb.com/rs/069-ALB-339/images/Repository%20Access%2004-09-2019.pdf>

リポジトリの資格情報を受け取ったら、次のことができます。

1. リポジトリ構成ファイルを作成します。
2. ユーザー名とパスワードを指定して、ファイルを変更します。
3. EDB*Plus をインストールします。

2.2 RPM インストールの実行

EnterpriseDB リポジトリを作成および使用して AdvancedServer またはそのサポートコンポーネントをインストールする方法の詳細については、次の URL にある*EDBPostgresAdvancedServer インストールガイド*を参照してください。

<https://www.enterprisedb.com/edb-docs>

リポジトリ設定ファイルの作成

リポジトリ構成ファイルを作成するには、スーパーユーザー特権を想定して、次のコマンドを呼び出します。

- RHEL または CentOS7 の場合:

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/
edb-repo-latest.noarch.rpm
```

- RHEL または CentOS8 の場合:

```
dnf -y install https://yum.enterprisedb.com/edb-repo-rpms/
edb-repo-latest.noarch.rpm
```

リポジトリ設定ファイルの名前は `edb.repo` です。ファイルは `/etc/yum.repos.d` にあります。

After creating the `edb.repo` file, use your choice of editor to ensure that the value of the `enabled` parameter is 1, and replace the `username` and `password` placeholders in the `baseurl` specification with the name and password of a registered EnterpriseDB user.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
↪$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

構成ファイルへの変更を保存した後、次のコマンドを使用して EDB*Plus をインストールできます。

- RHEL または CentOS7 の場合:

```
yum install edb-as<version>-edbplus
```

- RHEL または CentOS8 の場合:

```
dnf install edb-as<version>-edbplus
```

ここで、***version***は AdvancedServer のバージョンです。

システムで認識されないソースによって署名された **RPM** パッケージをインストールすると、**yum** はローカルサーバーにキーをインポートする許可を要求する場合があります。プロンプトが表示され、パッケージが信頼できるソースからのものであることに満足したら、**y** を入力し、Return を押して続行します。

インストール中に、**yum** は解決できない依存関係に遭遇する可能性があります。含まれている場合は、手動で解決する必要がある必要な依存関係のリストが表示されます。

2.2.1 RPM インストールの構成

EDB*Plus の RPM インストールを実行した後、EDB*Plus が Java インストールを検索できるようにする環境変数の値を設定する必要があります。次のコマンドを使用して、変数値を設定します。

```
export JAVA_HOME=<path_to_java>
export PATH=<path_to_java>/bin:$PATH
```

By default, the `pg_hba.conf` file for the RPM installer enforces IDENT authentication. Before invoking EDB*Plus, you must either modify the `pg_hba.conf` file, changing the authentication method to a form other than IDENT (and restarting the server), or perform the following steps to ensure that an IDENT server is accessible:

You must confirm that an `identd` server is installed and running. You can use the `yum` package manager to install an `identd` server by invoking the command:

- RHEL または CentOS7 の場合:

```
yum install xinetd authd
```

- RHEL または CentOS8 の場合:

```
dnf install xinetd authd
```

コマンドは以下を含む `/etc/xinetd.d/auth` という名前のファイルを作成する必要があります:

```
service auth
{
    disable = yes
    socket_type = stream
    wait =no
    user = ident
    cps = 4096 10
    instances = UNLIMITED
    server = /usr/sbin/in.authd server_args = -t60 --xerror -os
}
```

注釈: if the file includes a `-E` argument at the end of the server arguments, please erase `-E`.

次に、`identd` サーバーを起動するには、次のコマンドを呼び出します:

```
systemctl enable xinetd
```

```
systemctl start xinetd
```

pg_ident.conf ファイルを開き、ユーザーマッピングを作成します:

# map_name	system_username	postgres_username
edbas	enterprisedb	enterprisedb

どこ:

- map_name 列に指定された名前は、pg_hba.conf ファイル内のマッピングを識別するユーザー定義の名前です。
- system_username 列に指定された名前は enterprisedb です。
- postgres_username 列に指定された名前は enterprisedb です。

次に、pg_hba.conf ファイルを開き、IDENT エントリを変更します。

- IPv4 ローカル接続を使用している場合は、次のようにファイルエントリを変更します。

```
host all all 127.0.0.0/0 ident map=edbas
```

- IPv6 ローカル接続を使用している場合は、ファイルエントリを次のように変更します。

```
host all all:::1/128 ident map=edbas
```

EDB*Plus を起動する前に、AdvancedServer サービスを再起動する必要があります。AdvancedServer サービスの制御の詳細については、次の場所にある*EDBPostgresAdvancedServer インストールガイド*を参照してください。

<https://www.enterprisedb.com/resources/product-documentation>

2.3 グラフィカルインストーラーの使用

EDB*Plus のグラフィカルインストーラーは StackBuilderPlus 経由で利用できます。StackBuilderPlus には、Windows または Linux のスタートメニューからアクセスできます。StackBuilderPlus を開き、EDB*Plus をインストールするインストールを選択したら、コンポーネント選択画面のツリーコントロールを展開して、EDB*Plus インストーラーを選択してダウンロードします。

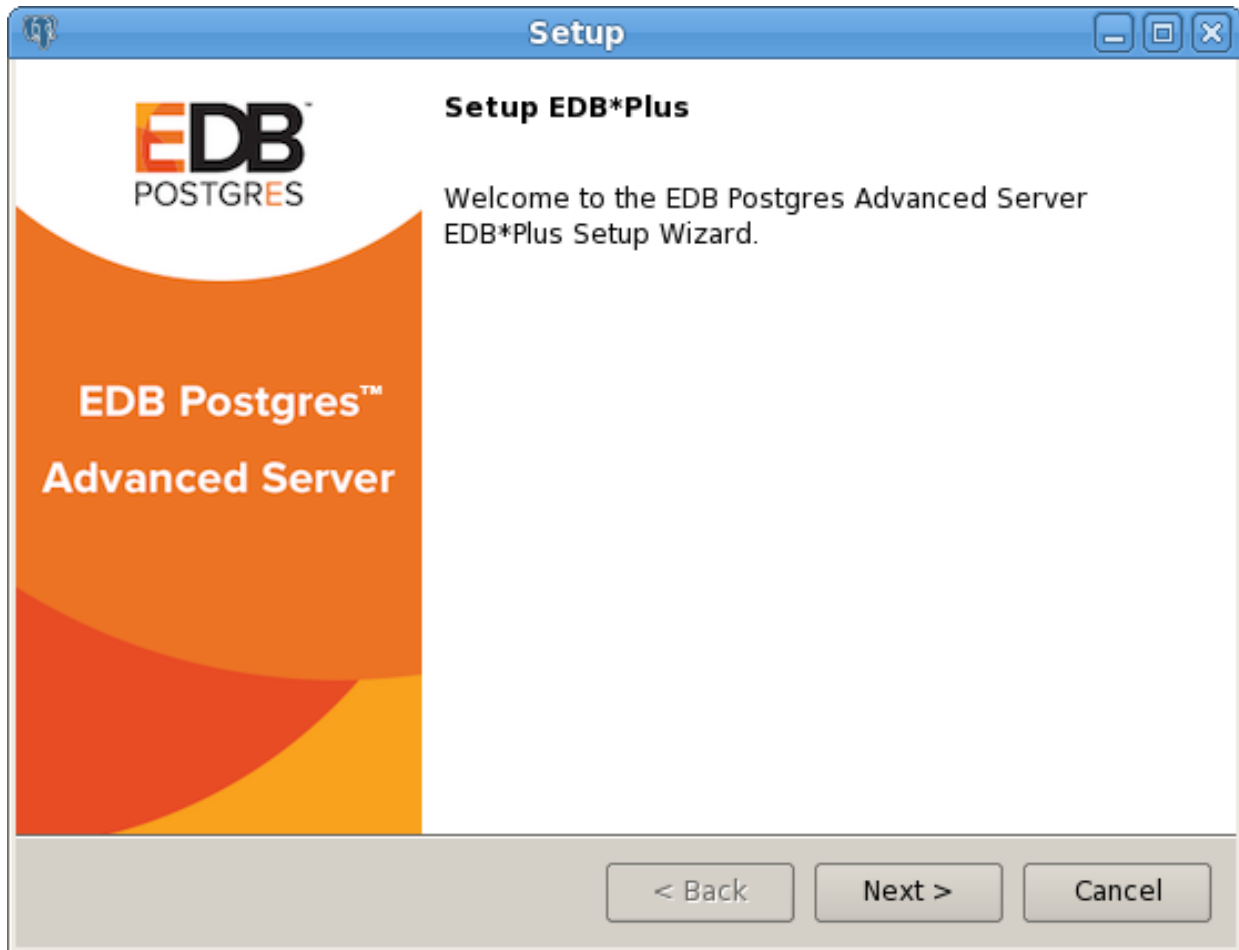


図 1 EDB*Plus ウェルカムウィンドウ

次の図に示すように、EDB*Plus インストーラーはセットアップウィザードによるこそ。

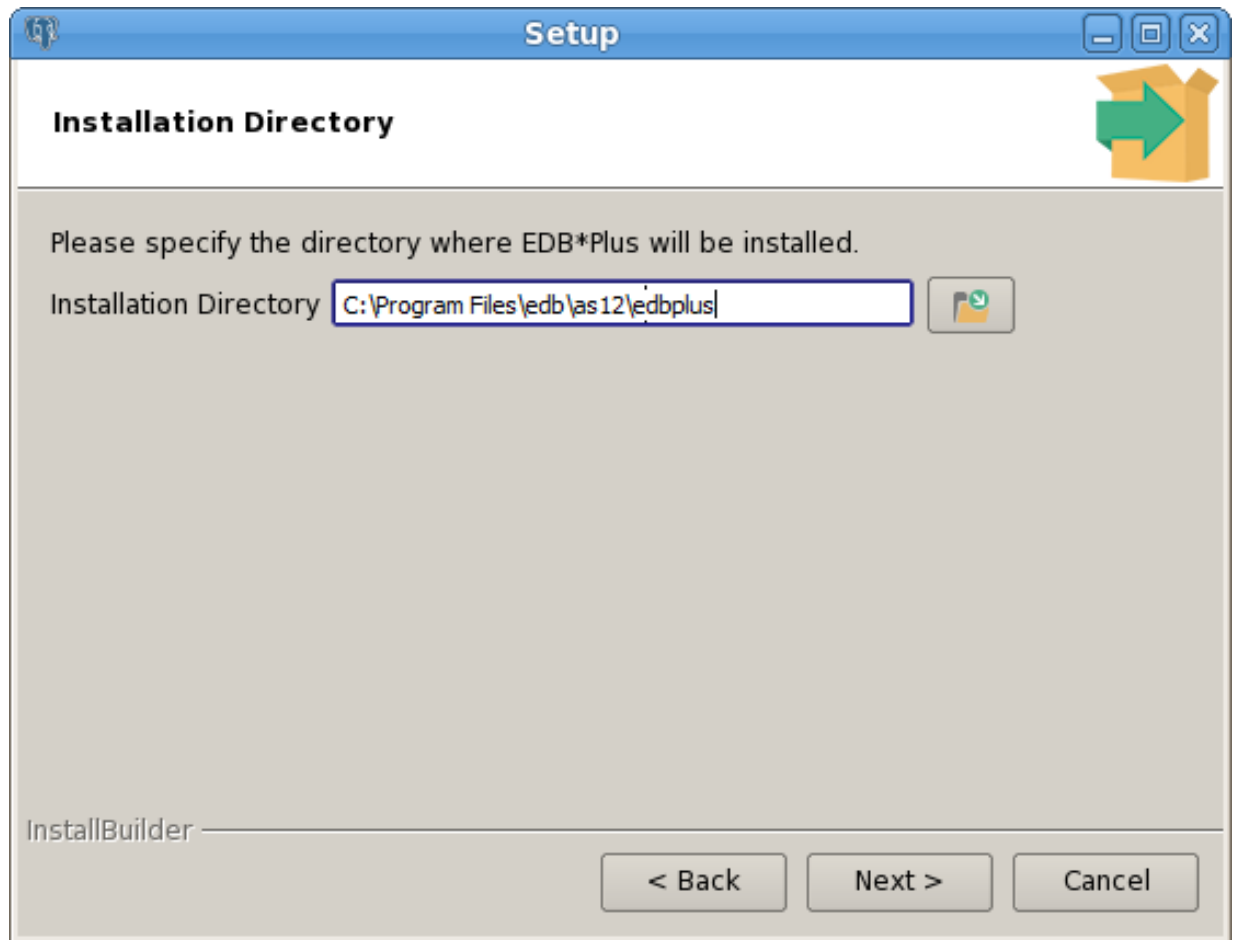


図 2 インストールディレクトリウィンドウ

Installation Directory フィールドを使用して、EDB*Plus ソフトウェアをインストールするディレクトリを指定します。次に、Next をクリックして続行します。

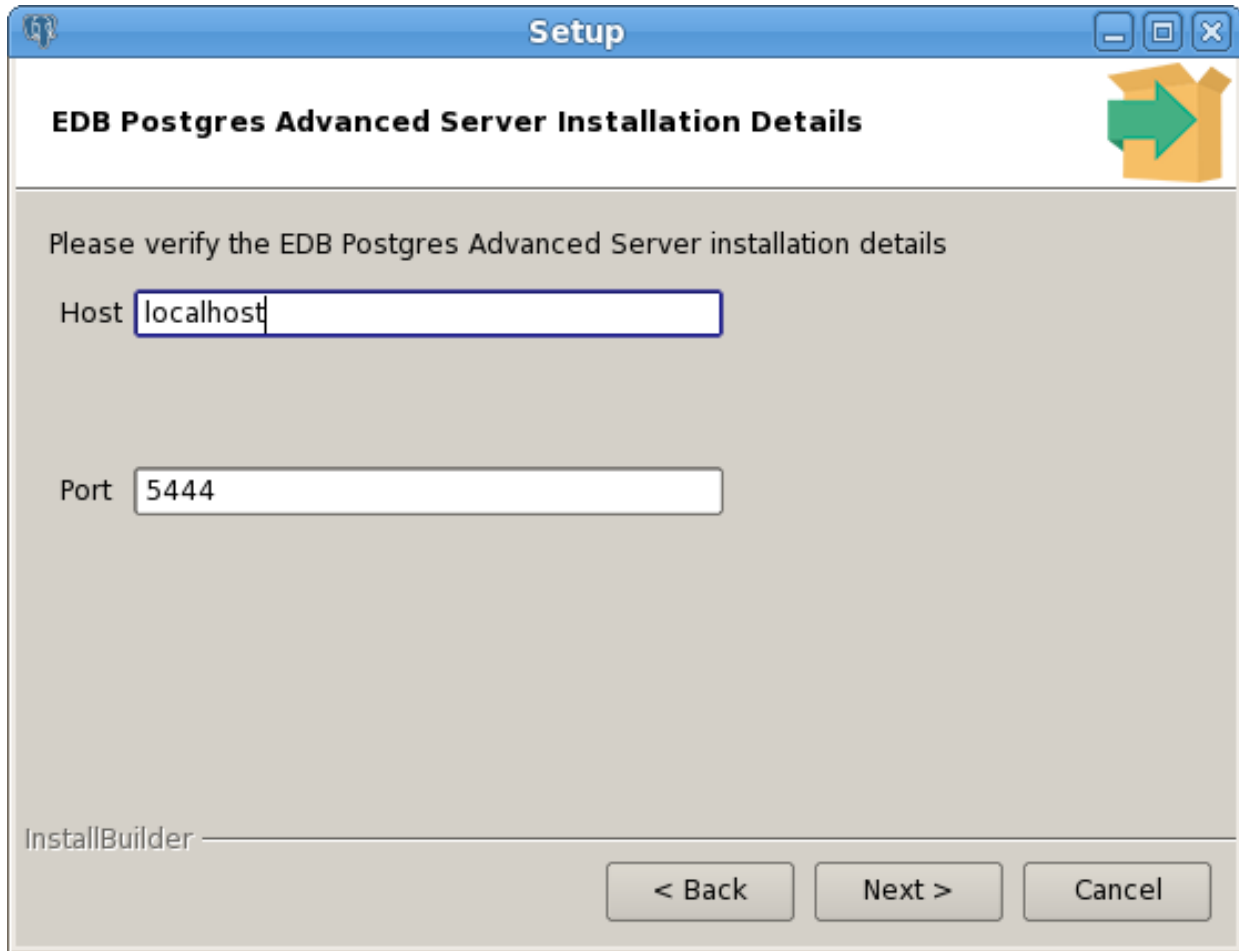


図3 [詳細サーバーインストールの詳細] ウィンドウ

EDB Postgres Advanced Server Installation Details ウィンドウのフィールドを使用して、AdvancedServer ホストの場所を特定します。

- ホスト フィールドを使用して、AdvancedServer が常駐するシステムを識別します。
- [ポート] フィールドを使用して、AdvancedServer がクライアント接続を監視するリスナーポートを識別します。

次に、Next をクリックして続行します。

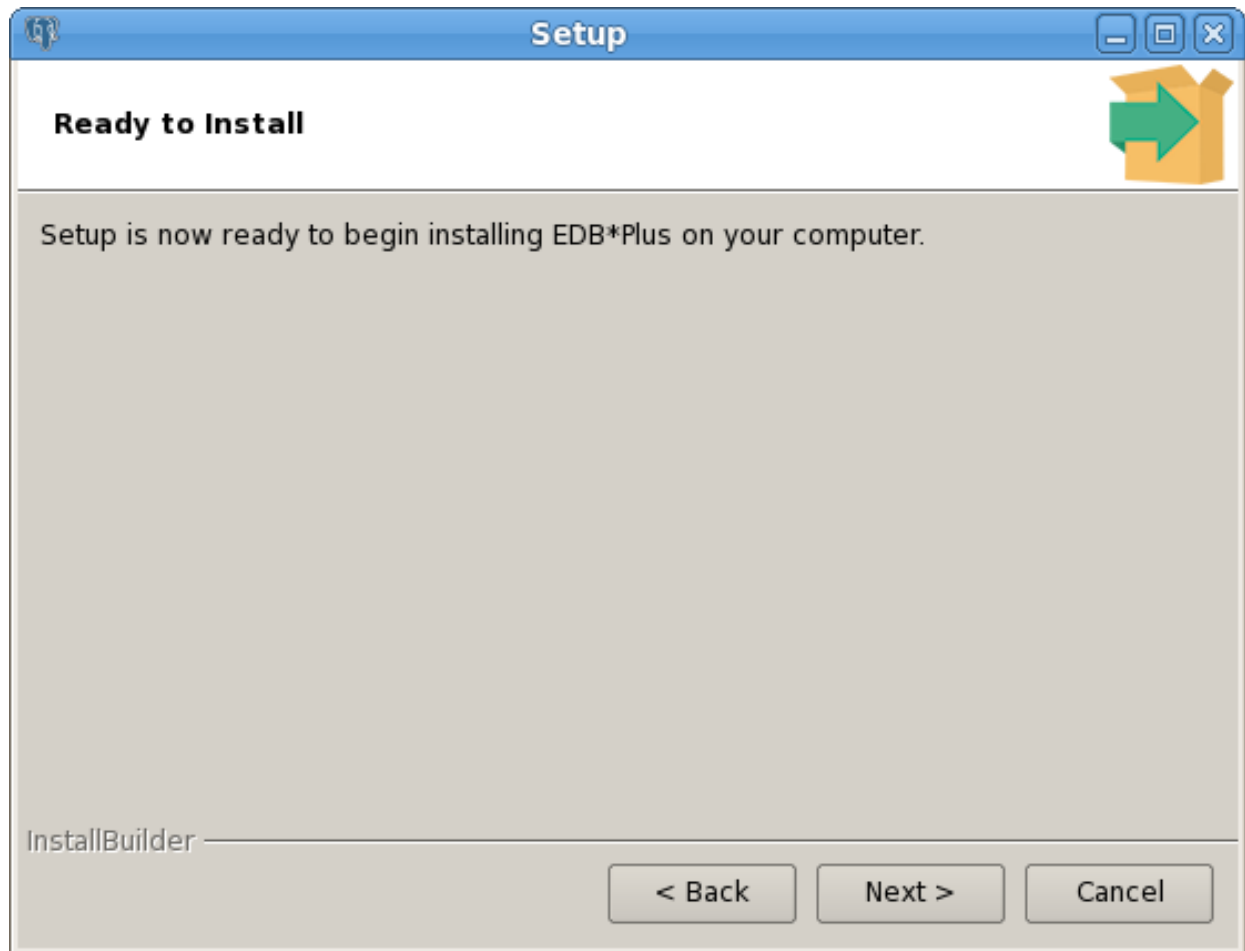


図 4 [インストールの準備完了] ウィンドウ

インストールの準備ウィンドウは、システムに EDB*Plus をインストールするために必要なすべての情報がインストーラーにあることを通知します。EDB*Plus をインストールするには、Next をクリックします。

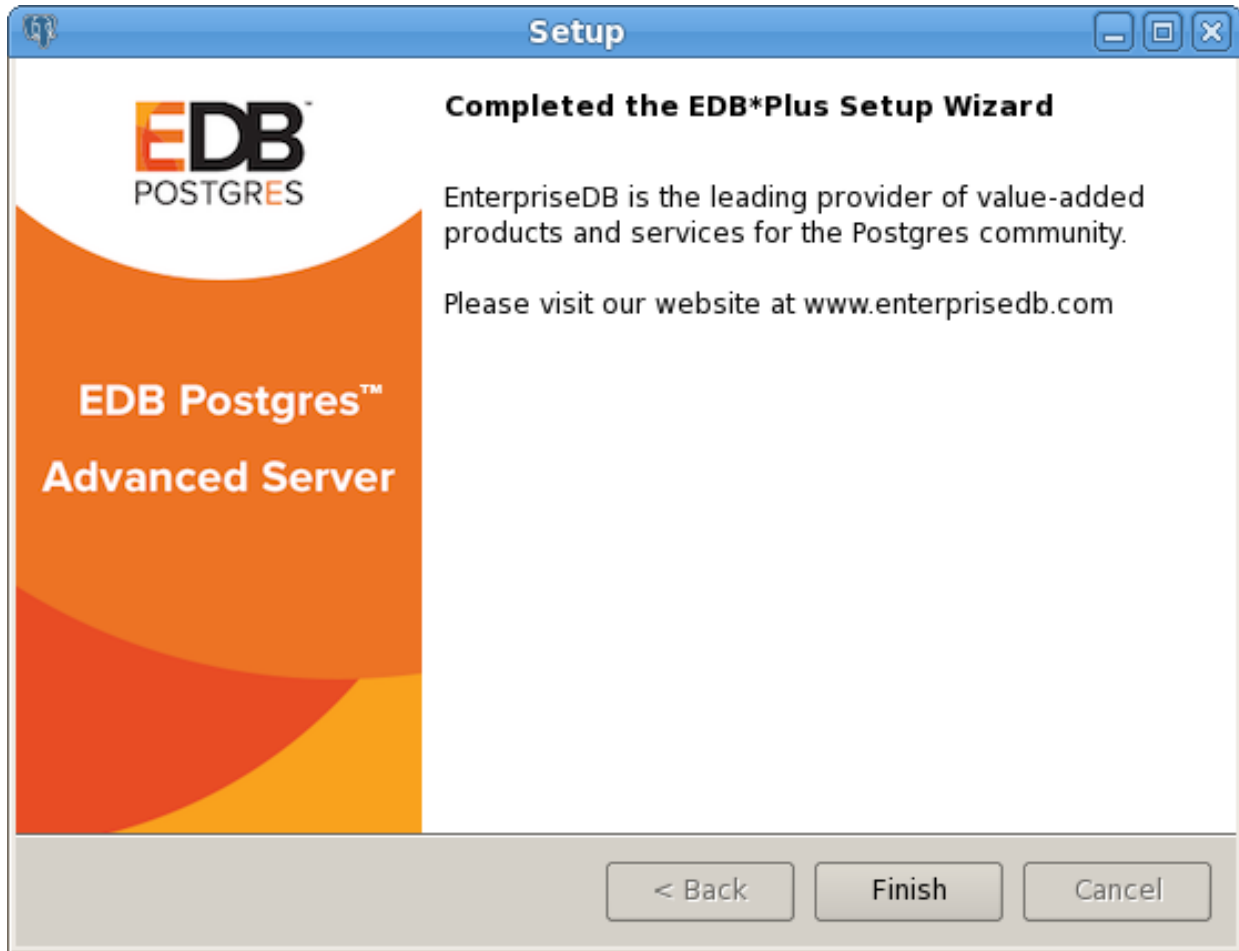


図5 インストールが完了しました

インストーラーは、セットアップウィザードが EDB*Plus のインストールを完了すると通知します。
Finish をクリックしてインストーラーを終了します。

EDB*Plus の使用

To open an EDB*Plus command line, navigate through the Applications or Start menu to the Advanced Server menu, to the Run SQL Command Line menu, and select the EDB*Plus option. You can also invoke EDB*Plus from the operating system command line with the following command:

```
edbplus [ -S[ILENT ] ] [ <login> | NOLOG ] [ @<scriptfile> [.<ext>  
] ] -SILENT
```

指定した場合、EDB*Plus サインオンバナーはすべてのプロンプトとともに抑制されます。

login

データベースサーバーとデータベースに接続するためのログイン情報。login は次の形式を取ります。ログイン情報内に空白があってはならないことに注意してください:

```
<username>[/<password>][@{<connectstring> | <variable> } ]
```

どこ:

- username はデータベースへの接続に使用するデータベースのユーザー名です。
- password は指定された*username*に関連付けられたパスワードです。***password***が提供されていないが、認証にパスワードが必要な場合、利用可能な場合はパスワードファイルが使用され

ます。パスワード・ファイルがないか、一致する接続パラメーターを持つパスワード・ファイルにエントリーがない場合、EDB*Plus はパスワードを求めるプロンプトを出します。

- `connectstring` は次の形式のデータベース接続文字列です:

```
<host> [:<port>][ / <dbname> ][?ssl={true | false}]
```

どこ:

- `host` は、データベースサーバーが存在するホスト名または IP アドレスです。@ `connectstring` も @ `variable` も /NOLOG も指定されていない場合、デフォルトのホストはローカルホストであると見なされます。
- `port` はデータベースサーバーで接続を受信するポート番号です。指定しない場合、デフォルトは 5444 です。
- `dbname` は接続するデータベースの名前です。指定しない場合のデフォルトは `edb` です。
- IPv4 の代わりに Internet Protocol version 6 (IPv6) が接続に使用されている場合、IP アドレスは角括弧で囲まれている必要があります (つまり、[`ipv6_address`])。以下は、IPv6 接続を含む例です。

```
edbplus.sh enterprisedb/password@[fe80::20c:29ff:fe7c:78b2]:5444/
edb
```

データベースサーバーの `pg_hba.conf` ファイルには、IPv6 接続用の適切なエントリが含まれている必要があります。次の例は、すべてのアドレスを許可するエントリを示しています。

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	host	all	all	:::0/0	md5

`pg_hba.conf` ファイルの詳細については、次の PostgreSQL コアのドキュメントを参照してください:

<https://www.postgresql.org/docs/Current/static/auth-pg-hba-conf.html>

SSL 接続が必要な場合は、接続文字列に `?ssl=true` パラメータを含めます。このような場合、接続文字列には、`/dbname*`の有無にかかわらず、少なくとも `*host:*port*`を含める必要があります。ssl パラメータが指定されていない場合、デフォルトは `false` です。SSL 接続の設定方法については、セクション 2.3 を参照してください。

- `variable` はデータベース接続文字列を含む `login.sql` ファイルで定義された変数です。`login.sql` ファイルは AdvancedServer ホームディレクトリの `edbplus` サブディレクトリにあります。

/NOLOG

データベース接続を確立せずに EDB*Plus を起動するには、/NOLOG を指定します。データベース接続を必要とする SQL コマンドおよび EDB*Plus コマンドは、このモードでは使用できません。/NOLOG オプションを指定して EDB*Plus を起動した後、CONNECT コマンドを指定してデータベースに接続できます。

scriptfile/ [.<ext>]

scriptfile is the name of a file residing in the current working directory, containing SQL and/or EDB*Plus commands that will be automatically executed after startup of EDB*Plus. ext is the filename extension. If the filename extension is sql, then the .sql extension may be omitted when specifying *scriptfile*. When creating a script file, always name the file with an extension, otherwise it will not be accessible by EDB*Plus. (EDB*Plus will always assume a .sql extension on filenames that are specified with no extension.)

次の例は、ポート 5444 の localhost 上のデータベースサーバーで実行されているデータベース edb に接続し、パスワード password を持つユーザー enterprisedb を示しています。

```
C:\Program Files\edb\as13\edbplus>edbplus enterprisedb/password
Connected to EnterpriseDB 13.0.1 (localhost:5444/edb) AS enterprisedb
EDB*Plus: Release 13 (Build 38.0.0)
Copyright (c) 2008-2017, EnterpriseDB Corporation. All rights reserved.
SQL>
```

次の例は、ポート 5445 の localhost 上のデータベースサーバーで実行されているデータベース edb に接続している、パスワード password を持つユーザー enterprisedb を示しています。

```
C:\Program Files\edb\as13\edbplus>edbplus enterprisedb/
↪password@localhost:5445/edb
Connected to EnterpriseDB 13.0.1 (localhost:5445/edb) AS enterprisedb

EDB*Plus: Release 13 (Build 38.0.0)
Copyright (c) 2008-2017, EnterpriseDB Corporation. All rights reserved.

SQL>
```

次の例は、login.sql ファイルで変数 hr_5445 を使用し、ポート 5445 で localhost のデータベース hr に接続する方法を示しています。

```
C:\Program Files\edb\as13\edbplus>edbplus enterprisedb/password@hr_5445
Connected to EnterpriseDB 13.0.1 (localhost:5445/hr) AS enterprisedb
```

(次のページに続く)

(前のページからの続き)

```
EDB*Plus: Release 13 (Build 38.0.0)
Copyright (c) 2008-2017, EnterpriseDB Corporation. All rights reserved.

SQL>
```

以下は、前の例で使用された login.sql ファイルの内容です。

```
define edb="localhost:5445/edb"
define hr_5445="localhost:5445/hr"
```

次の例は、スクリプトファイル dept_query.sql を実行した後 ポート 5444 でサーバー localhost のデータベース edb に接続しています。

```
C:\Program Files\edb\as13\edbplus>edbplus enterprisedb/password @dept_query
Connected to EnterpriseDB 13.0.1 (localhost:5444/edb) AS enterprisedb

SQL> SELECT * FROM dept;

DEPTNO  DNAME                LOC
-----  -
10      ACCOUNTING            NEW YORK
20      RESEARCH              DALLAS
30      SALES                 CHICAGO
40      OPERATIONS            BOSTON

SQL> EXIT
Disconnected from EnterpriseDB Database.
```

以下は、前の例で使用されたファイル dept_query.sql の内容です。

```
SET PAGESIZE 9999
SET ECHO ON
SELECT * FROM dept;
EXIT
```

SecureSocketsLayer (SSL) 接続の使用

AdvancedServer データベースへの EDB*Plus 接続は、SecureSocketsLayer (SSL) 接続を使用して実行できます。

SSL を使用するには、SSL 接続に関連するデータベースサーバーでさまざまな前提条件の構成手順を実行し、EDB*Plus を実行するホストで Java トラストストアとキーストアを作成する必要があります。

Java*truststore*は、Java クライアント (EDB*Plus) が SSL 接続を開始するサーバーの信頼性を検証するために使用する認証局 (CA) 証明書を含むファイルです。

Java*keystore*は、秘密鍵と公開鍵、およびそれらに対応する証明書を含むファイルです。キーストアは、EDB*Plus 接続に使用されるサーバーへのクライアント認証に必要です。

以下は、SSL 接続の設定に関するガイダンスとして参照できる資料です。

- AdvancedServer データベースへの SSL 接続の設定については、次の場所にある*PostgreSQL-CoreDocumentation*の (*ServerSetupandOperation*) の SSL によるセキュアな TCP 接続に関するセクションを参照してください。

<https://www.postgresql.org/docs/11/ssl-tcp.html>

- SSL を使用した JDBC クライアント接続の詳細については、次の場所にある*PostgreSQLJDBC インターフェース*のクライアント (Using SSL) の構成に関するセクションを参照してください

さい。

<https://jdbc.postgresql.org/documentation/head/ssl-client.html>

以下のセクションでは、SSL を使用する構成手順について説明します。

- AdvancedServer での SSL の構成
- EDB*Plus クライアントの SSL の構成
- AdvancedServer データベースへの SSL 接続の要求

4.1 AdvancedServer での SSL の構成

このセクションでは、データベースサーバーで SSL を構成して、EDB*Plus で SSL を使用する例を示します。この目的のために、自己署名証明書が使用されます。

ステップ 1: 証明書署名要求 (CSR) を作成します。

次の例では、生成された証明書署名要求ファイルは `server.csr` です。秘密鍵はファイル `server.key` として生成されます。

```
$ openssl req -new -nodes -text -out server.csr \
> -keyout server.key -subj "/CN=enterprisedb"
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'server.key'
-----
```

注釈: 証明書を作成する場合、共通名フィールドに指定する値（この例では `CN=enterprisedb` と指定）は、EDB*Plus に接続するときに指定したデータベースユーザー名である必要があります。

さらに、`pg_ident.conf` ファイルで定義されているユーザー名マップを使用して、共通名とデータベースユーザー名の柔軟性を高めることができます。手順 8 と 9 は、ユーザー名 maps の使用について説明しています。

ステップ 2: 自己署名証明書を生成します。

以下は、証明書署名要求ファイル `server.csr` と秘密鍵 `server.key` を入力として使用して、ファイル `server.crt` に自己署名証明書を生成します。

```
$ openssl x509 -req -days 365 -in server.csr -signkey server.key \  
> -out server.crt  
Signature ok  
subject=/CN=enterprisedb  
Getting Private key
```

ステップ 3: ルート認証局(CA)ファイル(`root.crt`)として使用するサーバー証明書(`server.crt`)のコピーを作成します。

```
$ cp server.crt root.crt
```

ステップ 4: 冗長な証明書署名リクエスト (`server.csr`) を削除します。

```
$ rm -f server.csr
```

ステップ 5: 証明書と秘密鍵ファイルを **AdvancedServer** データディレクトリ (たとえば、`/opt/edb/as10/data`) に移動またはコピーします。

```
$ mv root.crt /opt/edb/as10/data  
$ mv server.crt /opt/edb/as10/data  
$ mv server.key /opt/edb/as10/data
```

ステップ 6: 証明書ファイルと秘密鍵ファイルにファイルの所有権と権限を設定します。

所有権を、データベースサーバーの `data` サブディレクトリを所有するオペレーティングシステムアカウントに設定します。所有者以外のグループまたはアカウントがこれらのファイルにアクセスできないように、アクセス許可を設定します。

```
$ chown enterprisedb root.crt server.crt server.key  
$ chgrp enterprisedb root.crt server.crt server.key  
$ chmod 600 root.crt server.crt server.key  
$ ls -l  
total 152  
.  
.  
.  
-rw----- 1 enterprisedb enterprisedb 985 Aug 22 11:00 root.crt  
-rw----- 1 enterprisedb enterprisedb 985 Aug 22 10:59 server.crt  
-rw----- 1 enterprisedb enterprisedb 1704 Aug 22 10:58 server.key
```

ステップ 7: `postgresql.conf` ファイルで、次の変更を行います。

```
ssl = on
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ca_file = 'root.crt'
```

ステップ 8: `pg_hba.conf` ファイルを変更して、EDB*Plus が SSL 接続を行う目的のデータベースで SSL の使用を有効にします。

`pg_hba.conf` ファイルの `hostssl` タイプは、クライアント (EDB*Plus) からの SSL 接続試行を検証するためにエントリが使用されることを示します。

認証方法は、証明書の共通名 (この例では `enterprisedb`) を使用して認証が実行されるクライアントからの SSL 証明書を要求するために、オプション `clientcert=1` で `cert` に設定されています。

The `map=sslusers` option specifies that a mapping named `sslusers` defined in the `pg_ident.conf` file is to be used for authentication. This mapping allows a connection to the database if the common name from the certificate and the database user name attempting the connection match the SYSTEM-USERNAME/PG-USERNAME pair listed in the `pg_ident.conf` file.

以下は、データベース (`edb`) が SSL 接続を使用する必要がある場合の `pg_hba.conf` ファイルの設定例です。

```
# TYPE          DATABASE   USER        ADDRESS            METHOD
# "local" is for Unix domain socket connections only
local          all         all         md5
# IPv4 local connections:
hostssl        edb         all         192.168.2.0/24     cert clientcert=1
↪map=sslusers
```

ステップ 9: 以下は、`map=sslusers` オプションによって `pg_hba.conf` ファイルに関連する `pg_ident.conf` ファイル内のユーザー名マップを示しています。これらのユーザー名マップを使用すると、EDB*Plus に接続するときにデータベースユーザー名 `edbuser`、`postgres`、または `enterprisedb` を指定できます。

```
# MAPNAME      SYSTEM-USERNAME  PG-USERNAME
sslusers       enterprisedb     edbuser
sslusers       enterprisedb     postgres
sslusers       enterprisedb     enterprisedb
```

ステップ 10: 構成ファイルに変更を加えた後、データベースサーバーを再起動します。

4.1.1 EDB*Plus クライアントの SSL の構成

データベースサーバーで SSL を構成した後、次の手順では、EDB*Plus (JDBC クライアント) の証明書とキーストアファイルを生成する例を示します。

ステップ 1: データベースサーバーのデータサブディレクトリの下にあるファイル `server.crt` および `server.key` を使用して、これらのファイルのコピーを作成し、EDB*Plus を実行するホストに移動します。

これらのファイルを目的のディレクトリに保存して、次の手順で生成される信頼できる証明書とキーストアファイルを含めます。推奨される場所は、EDB*Plus を呼び出すホームユーザーアカウントの下に `.postgresql` サブディレクトリを作成することです。したがって、これらのファイルは、EDB*Plus を実行するユーザーアカウントの `~/.postgresql` ディレクトリの下にあります。

この例では、ファイル `edb.crt` は `server.crt` のコピーであり、`edb.key` は `server.key` のコピーであると想定しています。

ステップ 2: `edb.crt` の追加コピーを作成します。

```
$ cp edb.crt edb_root.crt
$ ls -l
total 12
-rw-r--r-- 1 user user 985 Aug 22 14:17 edb.crt
-rw-r--r-- 1 user user 1704 Aug 22 14:18 edb.key
-rw-r--r-- 1 user user 985 Aug 22 14:19 edb_root.crt
```

ステップ 3: ファイル `edb_root.crt` の `DistinguishedEncodingRules` (DER) 形式を作成します。このファイルの生成された DER 形式は `edb_root.crt.der` です。手順 4 で使用される `keytool` プログラムには、ファイルの DER 形式が必要です。

```
$ openssl x509 -in edb_root.crt -out edb_root.crt.der -outform der
$ ls -l
total 16
-rw-r--r-- 1 user user 985 Aug 22 14:17 edb.crt
-rw-r--r-- 1 user user 1704 Aug 22 14:18 edb.key
-rw-r--r-- 1 user user 985 Aug 22 14:19 edb_root.crt
-rw-rw-r-- 1 user user 686 Aug 22 14:21 edb_root.crt.der
```

ステップ 4: `keytool` プログラムを使用して、入力として `edb_root.crt.der` を使用してキーストアファイル (`postgresql.keystore`) を作成します。このプロセスにより、Postgres データベースサーバーの証明書がキーストアファイルに追加されます。

Note: The file name `postgresql.keystore` is recommended so that it can be accessed in its default directory location `~/.postgresql/postgresql.keystore`, which is under the home directory of the user account invoking EDB*Plus. Also note that the file name suffix can be `.jks` instead of `.keystore` (thus, file name `postgresql.jks`). However, in the remainder of these examples, file name `postgresql.keystore` is used.

Windows のみ: パスは `%APPDATA%/.postgresql/postgresql.keystore` です

`keytool` プログラムは、Java ランタイム環境インストールの `bin` サブディレクトリにあります。

新しいパスワードの入力を求められます。このパスワードは **PGSSLCERTPASS** 環境変数で指定する必要があるため、保存してください。

```
$ /usr/java/jdk1.8.0_131/jre/bin/keytool -keystore postgresql.keystore
> -alias postgresqlstore -import -file edb_root.crt.der
Enter keystore password:
Re-enter new password:
Owner: CN=enterprisedb
Issuer: CN=enterprisedb
Serial number: c60f40256b0e8d53
Valid from: Tue Aug 22 10:59:25 EDT 2017 until: Wed Aug 22 10:59:25 EDT
2018
Certificate fingerprints:
    MD5: 85:0B:E9:A7:6E:4F:7C:B0:9B:D6:3A:44:55:E2:E9:8E
    SHA1: DD:A6:71:24:0B:6C:F8:BC:7A:4C:89:9B:DC:22:6A:6C:B0:F5:3F:7C
    SHA256:
↳DC:02:64:E2:B0:E9:6F:1C:FC:4F:AE:E6:18:85:0B:79:57:43:C3:C5:AE:43:0D:37:49:53:6D:11:69:06
    Signature algorithm name: SHA1withRSA
    Version: 1
Trust this certificate? [no]: yes
Certificate was added to keystore
```

ステップ 5: ファイル `edb.crt` および `edb.key` を入力として使用して、キーストアファイル (`postgresql.p12`) の PKCS #12 形式を作成します。

注: ファイル名 `postgresql.p12` は、EDB*Plus を呼び出すユーザーアカウントのホームディレクトリの下にあるデフォルトディレクトリの場合 `~/.postgresql/postgresql.p12` からアクセスできるようにすることをお勧めします。

Windows のみ: パスは `% APPDATA %/. postgresql/postgresql.p12` です

新しいパスワードの入力を求められます。このパスワードは **PGSSLKEYPASS** 環境変数で指定する必

必要があるため、保存してください。

```
$ openssl pkcs12 -export -in edb.crt -inkey edb.key -out postgresql.p12

Enter Export Password:

Verifying - Enter Export Password:

$ ls -l

total 24

-rw-rw-r-- 1 user user 985 Aug 24 12:18 edb.crt
-rw-rw-r-- 1 user user 1704 Aug 24 12:18 edb.key
-rw-rw-r-- 1 user user 985 Aug 24 12:20 edb_root.crt
-rw-rw-r-- 1 user user 686 Aug 24 12:20 edb_root.crt.der
-rw-rw-r-- 1 user user 758 Aug 24 12:26 postgresql.keystore
-rw-rw-r-- 1 user user 2285 Aug 24 12:28 postgresql.p12
```

ステップ 6: postgresql.keystore および postgresql.p12 ファイルがまだ ~/.postgresql ディレクトリにない場合、それらをその場所に移動またはコピーします。

Windows のみ: ディレクトリは %APPDATA%\postgresql です。

ステップ 7: デフォルトの場所 ~/.postgresql が使用されていない場合、 postgresql.keystore ファイルへのフルパス（ファイル名を含む）は PGSSLCERT 環境変数で設定する必要があります、およびファイル postgresql.p12 へのフルパス（ファイル名を含む）は、EDB*Plus を呼び出す前に PGSSLKEY 環境変数で設定する必要があります。

さらに、ステップ 4 で生成されたファイルの名前が postgresql.keystore または postgresql.jks ではなかった場合は、PGSSLCERT 環境変数を使用してファイル名とその場所を指定します。同様に、ステップ 5 で生成されたファイルの名前が postgresql.p12 ではなかった場合は、PGSSLKEY 環境変数を使用してファイル名とその場所を指定します。

4.1.2 EDB*Plus と AdvancedServer データベース間の SSL 接続の要求

SSL 接続を実行するために、次のトピックに対処してください。

- 信頼できる証明書とキーストアファイルは、データベースサーバーと EDB*Plus を呼び出すクライアントホストの両方に対して生成されています。
- データベースサーバーの `postgresql.conf` ファイルには、更新された構成パラメーターが含まれています。
- データベースサーバーの `pg_hba.conf` ファイルには、SSL 接続を許可するために必要なエントリが含まれています。
- クライアントホストの場合、クライアントの証明書とキーストアファイルがユーザーアカウントの `~/.postgresql` ディレクトリに配置されているか、環境変数 `PGSSLCERT` および `PGSSLKEY` が EDB*Plus を呼び出す前に設定されています。
- `PGSSLCERTPASS` 環境変数にパスワードが設定されています。
- `PGSSLKEYPASS` 環境変数にパスワードが設定されている

EDB*Plus を呼び出すときは、`connectstring` オプションのように、データベース接続文字列に `?ssl=true` パラメータを含めます。

次の例は、データベースサーバーに対してリモートのホストから呼び出された EDB*Plus を示しています。

データベースサーバーの `postgresql.conf` ファイルには、以下の変更されたパラメーターが含まれています:

```
ssl = on

ssl_cert_file = 'server.crt'

ssl_key_file = 'server.key'

ssl_ca_file = 'root.crt'
```

データベースサーバーの `pg_hba.conf` ファイルには、リモートホストの EDB*Plus から接続するための次のエントリが含まれています。

```
# TYPE DATABASE USER ADDRESS METHOD
```

(次のページに続く)

(前のページからの続き)

```
# "local" is for Unix domain socket connections only

local all all md5

# IPv4 local connections:

hostssl edb all 192.168.2.24/32 cert clientcert=1
```

EDB*Plus が呼び出されるリモートホストでは、user という名前の Linux ユーザーアカウントの ~/.postgresql ディレクトリに証明書とキーストアファイルが含まれています。

```
[user@localhost ~]$ whoami

user

[user@localhost ~]$ cd .postgresql

[user@localhost .postgresql]$ pwd

/home/user/.postgresql

[user@localhost .postgresql]$ ls -l

total 8

-rw-rw-r-- 1 user user 758 Aug 24 12:37 postgresql.keystore

-rw-rw-r-- 1 user user 2285 Aug 24 12:37 postgresql.p12

Logged into Linux with the account named user, EDB*Plus is successfully
invoked with the ssl=true parameter:

$ export PGSSLCERTPASS=keypass

$ export PGSSLKEYPASS=exppass

$ cd /opt/edb/as10/edbplus

$ ./edbplus.sh enterprisedb/password@192.168.2.22:5444/edb?ssl=true

Connected to EnterpriseDB 10.0.1 (192.168.2.22:5444/edb) AS enterprisedb
```

(次のページに続く)

(前のページからの続き)

```
EDB*Plus: Release 10 (Build 36.0.0)

Copyright (c) 2008-2017, EnterpriseDB Corporation. All rights reserved.

SQL>
```

または、証明書とキーストアファイルを ~/.postgresql に配置せずに、別のディレクトリに配置して、EDB*Plus を次の方法で呼び出すことができます:

```
$ export PGSSLCERT=/home/user/ssl/postgresql.keystore

$ export PGSSLKEY=/home/user/ssl/postgresql.p12

$ export PGSSLCERTPASS=keypass

$ export PGSSLKEYPASS=exppass

$ cd /opt/edb/as10/edbplus

$ ./edbplus.sh enterprisedb/password@192.168.2.22:5444/edb?ssl=true

Connected to EnterpriseDB 10.0.1 (192.168.2.22:5444/edb) AS enterprisedb

EDB*Plus: Release 10 (Build 36.0.0)

Copyright (c) 2008-2017, EnterpriseDB Corporation. All rights reserved.

SQL>
```

どちらの場合も、EDB*Plus へのログインに使用されるデータベースユーザー名は `enterprisedb` であることに注意してください。これは、証明書の作成時に共通名フィールドに指定されたユーザーだからです。map オプション付きの `pg_hba.conf` ファイルと `pg_ident.conf` ファイルが使用されている場合、他のデータベースユーザー名を使用できます。

コマンドの要約

次のセクションには、EDB*Plus コマンドの要約が含まれています。

5.1 ACCEPT

ACCEPT コマンドはプロンプトを表示し、ユーザーのキーボード入力を待ちます。ユーザーが入力した値は、指定した変数に格納されます。

```
ACC[EPT ] variable
```

次の例では、**my_name** という名前の新しい変数を作成し、**JohnSmith** の値を受け入れ、**DEFINE** コマンドを使用して値を表示します。

```
SQL> ACCEPT my_name
Enter value for my_name: John Smith
SQL> DEFINE my_name
DEFINE MY_NAME = "John Smith"
```

5.2 APPEND

APPEND は、SQL バッファの現在の行の最後に指定されたテキストを追加する行エディターコマンドです。

```
A[PPEND ] text
```

次の例では、APPEND コマンドを使用して、SELECT コマンドが SQL バッファに組み込まれています。APPEND コマンドと WHERE 句の間に 2 つのスペースが置かれ、SQL バッファで 1 つのスペースで dept と WHERE を区切ることに注意してください。

```
SQL> APPEND SELECT * FROM dept
SQL> LIST
1 SELECT * FROM dept
SQL> APPEND WHERE deptno = 10
SQL> LIST
1 SELECT * FROM dept WHERE deptno = 10
```

5.3 CHANGE

CHANGE は、SQL バッファの現在の行で検索と置換を実行する行エディターコマンドです。

```
C[HANGE ] FROM [ TO ]
```

If TO/ is specified, the first occurrence of text FROM in the current line is changed to text TO. If TO/ is omitted, the first occurrence of text FROM in the current line is deleted.

次の一連のコマンドは、3 行目を現在の行にしてから、WHERE 句の部門番号を 20 から 30 に変更します。

```
SQL> LIST
1 SELECT empno, ename, job, sal, comm
2 FROM emp
3 WHERE deptno = 20
4* ORDER BY empno
SQL> 3
3* WHERE deptno = 20
SQL> CHANGE /20/30/
3* WHERE deptno = 30
SQL> LIST
1 SELECT empno, ename, job, sal, comm
```

(次のページに続く)

(前のページからの続き)

```
2 FROM emp
3 WHERE deptno = 30
4* ORDER BY empno
```

5.4 CLEAR

CLEAR コマンドは、SQL バッファの内容を削除するか、COLUMN コマンドで設定されたすべての列定義を削除するか、画面をクリアします。

```
CL[EAR ] [ BUFF[ER ] | SQL | COL[UMNS ] | SCR[EEN ] ] BUFFER | SQL
```

SQL バッファをクリアします。

COLUMNS

列定義を削除します。

SCREEN

画面をクリアします。これは、オプションが指定されていない場合のデフォルトです。

5.5 COLUMN

COLUMN コマンドは、出力フォーマットを制御します。COLUMN コマンドを使用して設定されたフォーマット属性は、現在のセッションの間のみ有効です。

```
COL[UMN ]
[ column
{ CLE[AR ] |
{ FOR[MAT ] spec |
HEA[DING ] text |
{ OFF | ON }
} [...]
}
]
```

後続のオプションなしで COLUMN コマンドを指定すると、セッションで有効な現在の列のフォーマットオプションが表示されます。

COLUMN コマンドの後に列名が続く場合、列名の後に次のいずれかが続く場合があります。

1. 他のオプションはありません
2. CLEAR
3. FORMAT、HEADING、および OFF または ON のいずれかの組み合わせ

column

Name of a column in a table to which subsequent column formatting options are to apply. If no other options follow column, then the current column formatting options if any, of column are displayed.

CLEAR

CLEAR オプションは、すべてのフォーマットオプションをデフォルトの column に戻します。CLEAR オプションを指定する場合は、それだけを指定する必要があります。

spec

column に適用するフォーマット指定。文字列の場合、spec は次の形式を取ります:

n

n is a positive integer that specifies the column width in characters within which to display the data. Data in excess of n will wrap around with the specified column width.

数値列の場合、spec は次の要素で構成されます。

表-数値列フォーマット要素

素子	説明
\$	先頭のドル記号を表示します。
,	指定された位置にコンマを表示します。
.	小数点の位置をマークします。
0	先行ゼロを表示します。
9	表示する有効桁数。

フォーマットのオーバーフローが原因で有効数字の損失が発生した場合、すべての # が表示されます。

text

column の列見出しに使用されるテキスト。

OFF | ON

OFF を指定すると、フォーマットオプションはデフォルトに戻りますが、セッション内では引き続き使用できます。ON を指定すると、セッション内の column に対して以前の COLUMN コマンドで指定されたフォーマットオプションが再度アクティブになります。

次の例は、ジョブ列の表示幅を変更した場合の影響を示しています。

```
SQL> SET PAGESIZE 9999
SQL> COLUMN job FORMAT A5
SQL> COLUMN job
COLUMN JOB ON
FORMAT A5
wrapped
SQL> SELECT empno, ename, job FROM emp;
```

EMPNO	ENAME	JOB
-----	-----	-----
7369	SMITH	CLERK
799	ALLEN	SALES
		MAN
7521	WARD	SALES
		MAN
7566	JONES	MANAG
		ER
7654	MARTIN	SALES
		MAN
7698	BLAKE	MANAG
		ER
7782	CLARK	MANAG
		ER
7788	SCOTT	ANALY
		ST
7839	KING	PRESI
		DENT
7844	TURNER	SALES
		MAN
7876	ADAMS	CLERK
7900	JAMES	CLERK
7902	FORD	ANALY
		ST
7934	MILLER	CLERK

(次のページに続く)

(前のページからの続き)

14 rows retrieved.

次の例では、sal 列にフォーマットを適用します。

```
SQL> COLUMN sal FORMAT $99,999.00
SQL> COLUMN
COLUMN JOB ON
FORMAT A5
wrapped
COLUMN SAL ON
FORMAT $99,999.00
wrapped
SQL> SELECT empno, ename, job, sal FROM emp;
```

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	\$800.00
7499	ALLEN	SALES	\$1,600.00
		MAN	
7521	WARD	SALES	\$1,250.00
		MAN	
7566	JONES	MANAG	\$2,975.00
		ER	
7654	MARTIN	SALES	\$1,250.00
		MAN	
7698	BLAKE	MANAG	\$2,850.00
		ER	
7782	CLARK	MANAG	\$2,450.00
		ER	
7788	SCOTT	ANALY	\$3,000.00
		ST	
7839	KING	PRESI	\$5,000.00
		DENT	
7844	TURNER	SALES	\$1,500.00
		MAN	
7876	ADAMS	CLERK	\$1,100.00
7900	JAMES	CLERK	\$950.00
7902	FORD	ANALY	\$3,000.00
		ST	
7934	MILLER	CLERK	\$1,300.00

14 rows retrieved.

5.6 CONNECT

データベース接続を別のユーザーに変更するか、別のデータベースに接続します。CONNECT コマンドに続くパラメータの間に空白があってはなりません。

```
CON[NECT] username [/password][@{ connectstring | variable }]
```

どこ:

username はデータベースへの接続に使用するデータベースのユーザー名です。

password は指定された username に関連付けられたパスワードです。password が提供されていないが、認証にパスワードが必要な場合、最初に EDB*Plus を呼び出す Linux オペレーティングシステムアカウントのホームディレクトリ（または %APPDATA%/postgresql/ Windows のディレクトリ）、次に PGPASSFILE 環境変数で指定された場所にあります。パスワードファイルは、Linux ホストでは .pgpass、Windows ホストでは pgpass.conf です。以下は、Windows ホストでの例です。

```
C:/Users/Administrator/AppData/Roaming/postgresql/pgpass.conf
```

パスワードファイルが見つからない場合、または EDB*Plus 接続パラメーターに一致するエントリがない場合、EDB*Plus はパスワードを要求します。パスワードファイルの詳細については、次の PostgreSQL コアドキュメントを参照してください。

<https://www.postgresql.org/docs/current/static/libpq-pgpass.html>

Note: When a password is not required, EDB*Plus does not prompt for a password such as when the trust authentication method is specified in the pg_hba.conf file. For more information about the pg_hba.conf file and authentication methods, see the PostgreSQL core documentation at:

<https://www.postgresql.org/docs/current/static/auth-pg-hba-conf.html>

connectstring はデータベース接続文字列です。データベース接続文字列の詳細については、セクション 2.2 を参照してください。

variable はデータベース接続文字列を含む login.sql ファイルで定義された変数です。login.sql ファイルは AdvancedServer ホームディレクトリの edbplus サブディレクトリにあります。

次の例では、データベース接続は、ユーザー名 smith のポート 5445 で localhost のデータベース edb に変更されます。

```
SQL> CONNECT smith/mypassword@localhost:5445/edb
Disconnected from EnterpriseDB Database.
Connected to EnterpriseDB 13.0.1 (localhost:5445/edb) AS smith
```

上記のセッション内から、接続はユーザー名 enterprisedb に変更されます。また、ホストのデフォルトは localhost、ポートのデフォルトは 5444（以前に使用したポートとは異なります）、データベースのデフォルトは edb です。

```
SQL> CONNECT enterprisedb/password
Disconnected from EnterpriseDB Database.
Connected to EnterpriseDB 13.0.1 (localhost:5444/edb) AS enterprisedb
```

5.7 DEFINE

DEFINE コマンドは、***ユーザー変数***（***置換変数***とも呼ばれます）の値を作成または置換します。

```
DEF[INE ] [ <variable> [ = <text> ] ]
```

パラメーターなしで DEFINE コマンドを指定すると、現在のすべての変数とその値が表示されます。

If DEFINE variable is given, only variable is displayed with its value.

DEFINE variable = text assigns text to variable. text may be optionally enclosed within single or double quotation marks. Quotation marks must be used if text contains space characters.

次の例では、dept と name の 2 つの変数を定義しています。

```
SQL> DEFINE dept = 20
SQL> DEFINE name = 'John Smith'
SQL> DEFINE
DEFINE EDB = "localhost:5445/edb"
DEFINE DEPT = "20"
DEFINE NAME = "John Smith"
```

注: 変数 EDB は、AdvancedServer ホームディレクトリの edbplus サブディレクトリにある login.sql ファイルから読み取られます。

5.8 DEL

DEL は、SQL バッファから 1 つ以上の行を削除する行エディターコマンドです。

```
DEL [ n | n m | n * | n L[AST ] | n | L[AST ] | L[AST ] ]
```

パラメーターは、SQL バッファから削除する行を指定します。2 つのパラメーターは、削除する行の

範囲の開始と終了を指定します。パラメーターなしで DEL コマンドを指定すると、現在の行が削除されます。

n

n is an integer representing the nth line

n m

n and m are integers where m is greater than n representing the nth through the mth lines

- 現在の行

LAST

最終行

次の例では、列 sal と comm をそれぞれ含む 5 行目と 6 行目が SQL バッファの SELECT コマンドから削除されます。

```
SQL> LIST
1 SELECT
2 empno
3 ,ename
4 ,job
5 ,sal
6 ,comm
7 ,deptno
8* FROM emp
SQL> DEL 5 6
SQL> LIST
1 SELECT
2 empno
3 ,ename
4 ,job
5 ,deptno
6\* FROM emp
```

5.9 DESCRIBE

DESCRIBE コマンドは以下を表示します。

- テーブルまたはビューの列、列のデータ型、および列の長さのリスト
- プロシージャまたは関数のパラメータのリスト
- プロシージャと関数、およびパッケージのそれぞれのパラメータのリスト。

DESCRIBE コマンドは、シノニムによって参照されるデータベースオブジェクトの構造も表示します。構文は次のとおりです。

```
DESC[RIBE] [ <schema>.] <object>
```

schema

説明するオブジェクトを含むスキーマの名前。

object

表示するテーブル、ビュー、プロシージャ、関数、パッケージの名前、またはオブジェクトのシノニム。

5.10 DISCONNECT

DISCONNECT コマンドは、現在のデータベース接続を閉じますが、EDB*Plus を終了しません。

```
DISC[ONNECT ]
```

5.11 EDIT

EDIT コマンドは、外部エディターを呼び出して、オペレーティングシステムファイルまたは SQL バッファの内容を編集します。

```
ED[IT ] [ <filename> [.<ext> ] ]
```

<filename>[. <ext>]

filename is the name of the file to open with an external editor. ext is the filename extension. If the filename extension is sql, then the .sql extension may be omitted when specifying filename. EDIT

always assumes a .sql extension on filenames that are specified with no extension. If the filename parameter is omitted from the EDIT command, the contents of the SQL buffer are brought into the editor.

5.12 EXECUTE

EXECUTE コマンドは、EDB*Plus から SPL プロシージャを実行します。

```
EXEC[UTE ] <spl_procedure> [ ([ <parameters> ] ) ]
```

spl_procedure

実行する SPL プロシージャの名前。

parameters

パラメータのカンマ区切りリスト。パラメータがない場合は、空の括弧のペアをオプションで指定できます。

5.13 EXIT

EXIT コマンドは、EDB*Plus セッションを終了し、制御をオペレーティングシステムに戻します。QUIT は EXIT の同義語です。パラメーターを指定しないことは、EXITSUCCESSCOMMIT と同等です。

```
{ EXIT | QUIT }  
[ SUCCESS | FAILURE | WARNING | <value> | <variable> ]  
[ COMMIT | ROLLBACK ]SUCCESS | FAILURE |WARNING]
```

成功した操作、失敗、または成功、失敗、警告の警告をそれぞれ示す、オペレーティングシステムに依存する戻りコードを返します。デフォルトは SUCCESS です。

value

戻りコードとして返される整数値。

variable

戻りコードとして値が返される DEFINE コマンドで作成された変数。

コミット|ロールバック

COMMIT が指定されている場合、コミットされていない更新は終了時にコミットされます。ROLLBACK が指定されている場合、コミットされていない更新は終了時にロールバックされます。デフォルトは COMMIT です。

5.14 GET

GET コマンドは、指定されたファイルの内容を SQL バッファにロードします。

```
GET filename[.<ext> ] [ LIS[T ] | NOL[IST ] ]
```

filename [。<ext>]

filename is the name of the file to load into the SQL buffer. ext is the filename extension. If the filename extension is sql, then the .sql extension may be omitted when specifying filename. GET always assumes a .sql extension on filenames that are specified with no extension.

LIST | NOLIST

LIST が指定されている場合、SQL バッファの内容は、ファイルのロード後に表示されます。NOLIST を指定すると、リストは表示されません。デフォルトは LIST です。

5.15 HELP

HELP コマンドは、トピックのインデックスまたは特定のトピックに関するヘルプを取得します。疑問符 (?) は、HELP を指定することと同義です。

```
{ HELP | ? } { INDEX | <topic> }
```

INDEX

利用可能なトピックのインデックスを表示します。

topic

ヘルプが必要な特定のトピックの名前 (EDB*Plus コマンドなど)。

5.16 HOST

HOST コマンドは、EDB*Plus からオペレーティングシステムコマンドを実行します。

```
HO[ST ] [<os_command>]
```

os_command

実行するオペレーティングシステムコマンド。オペレーティングシステムコマンドを指定しない場合、EDB*Plus は実行を一時停止し、新しいシェルプロンプトを開きます。シェルが終了すると、EDB*Plus は実行を再開します。

5.17 INPUT

INPUT 行エディターコマンドは、SQL バッファの現在の行の後にテキスト行を追加します。

```
I[NPUT ] <text>
```

次の一連の INPUT コマンドは、SELECT コマンドを作成します。

```
SQL> INPUT SELECT empno, ename, job, sal, comm
SQL> INPUT FROM emp
SQL> INPUT WHERE deptno = 20
SQL> INPUT ORDER BY empno
SQL> LIST
1 SELECT empno, ename, job, sal, comm
2 FROM emp
3 WHERE deptno = 20
4* ORDER BY empno
```

5.18 LIST

LIST は、SQL バッファの内容を表示する行エディターコマンドです。

```
L[IST] [ n | n m | n * | n L[AST] | * | * n | * L[AST] | L[AST] ]
```

バッファには、EDB*Plus コマンドの履歴は含まれません。

n

n はバッファ行番号を表します。

`n m`

`nm` は、`n` と `m` の間の行のリストを表示します。

`n *`

`n*` は、行 `n` から現在の行までの範囲の行のリストを表示します。

`n L[AST]`

`nL[AST]` は、バッファの行 `n` から最後の行までの範囲の行のリストを表示します。

`*`

`*` は現在の行を表示します。

`* n`

`* n` は、現在の行から `n` 行目までの範囲の行のリストを表示します。

`* L[AST]`

`* L[AST]` は、現在の行から最後の行までの範囲の行のリストを表示します。

`L[AST]`

`L[AST]` は最後の行を表示します。

5.19 PASSWORD

PASSWORD コマンドを使用して、データベースのパスワードを変更します。

`PASSW[ORD] [<user_name>]`

PASSWORD コマンドを使用して別のユーザーのパスワードを変更するには、十分な特権が必要です。次の例は、**PASSWORD** コマンドを使用して、`acctg` という名前のユーザーのパスワードを変更する方法を示しています。

```
SQL> PASSWORD acctg
Changing password for acctg
New password:
New password again:
Password successfully changed.
```

5.20 PAUSE

PAUSE コマンドはメッセージを表示し、ユーザーが Enter キーを押すのを待ちます。

```
PAU[SE] [<optional_text>]
```

`optional_text` specifies the text that will be displayed to the user. If the `optional_text` is omitted, Advanced Server will display two blank lines. If you double quote the `optional_text` string, the quotes will be included in the output.

5.21 PROMPT

PROMPT コマンドは、続行する前にユーザーにメッセージを表示します。

```
PRO[MPT] [<message_text>]
```

`message_text` はユーザーに表示するテキストを指定します。出力に引用符を含めるには、文字列を二重引用符で囲みます。

5.22 QUIT

QUIT コマンドはセッションを終了し、制御をオペレーティングシステムに戻します。QUIT は EXIT の同義語です。

```
QUIT  
[SUCCESS | FAILURE | WARNING | value> | <sub_variable>]  
[COMMIT | ROLLBACK]``
```

デフォルト値は QUITSUCCESSCOMMIT です。

5.23 REMARK

REMARK を使用して、スクリプトにコメントを含めます。

```
REM[ARK] [<optional_text>]
```

次の規則を使用してコメントを含めることもできます。

```
*
* This is an example of a three line comment.
*
```

5.24 SAVE

SAVE コマンドを使用して、SQL バッファをオペレーティングシステムファイルに書き込みます。

```
SAV[E] <file_name> [CRE[ATE] | REP[LACE] | APP[END]]
```

file_name

file_name は、バッファの内容が書き込まれるファイルの名前（パスを含む）を指定します。ファイル拡張子を指定しない場合、.sql がファイル名の末尾に追加されます。

CREATE

新しいファイルを作成するには、CREATE キーワードを含めます。指定した名前のファイルがまだ存在しない場合にのみ、新しいファイルが作成されます。これがデフォルトです。

REPLACE

AdvancedServer が既存のファイルを上書きするように指定するには、REPLACE キーワードを含めます。

APPEND

APPEND キーワードを含めて、AdvancedServer が SQL バッファの内容を指定されたファイルの最後に追加するように指定します。

次の例では、SQL バッファの内容を一時ディレクトリにある ``example.sql`` という名前のファイルに保存します。

```
SQL>SAVEC:/example.sqlCREATEFile/"example.sql/" が書き込まれました。
```

5.25 SET

SET コマンドを使用して、EDB*Plus の動作を制御するセッションレベル変数の値を指定します。次の形式の SET コマンドが有効です。

オートコミットを設定する

SETAUTOCOMMIT コマンドを使用して、AdvancedServer トランザクションのコミット動作を指定します。

```
SET AUTO[COMMIT] {ON | OFF | IMMEDIATE | <statement_count>}
```

EDB*Plus は常に自動的に DDL ステートメントをコミットすることに注意してください。

ON

自動コミット動作をオンにするには、ON を指定します。

OFF

自動コミット動作をオフにするには、OFF を指定します。

IMMEDIATE

IMMEDIATE は ON と同じ効果があります。

statement_count

成功した SQL ステートメントの指定された数の後にコミットを発行するよう EDB*Plus に指示するには、statement_count の値を含めます。

列区切りを設定

SETCOLUMNSEPARATOR コマンドを使用して、AdvancedServer が列の間に表示するテキストを指定します。

```
SET COLSEP <column_separator>
```

デフォルト値の column_separator は単一のスペースです。

エコーを設定

SETECHO コマンドを使用して、SQL および EDB*Plus スクリプトステートメントを実行時に画面に表示するかどうかを指定します。

```
SET ECHO {ON | OFF}
```

デフォルト値はオフです。

フィードバックを設定

SETFEEDBACK コマンドは、SQL ステートメントの実行後のインタラクティブ情報の表示を制御します。

```
SET FEED[BACK] {ON | OFF | row_threshold}
```

row_threshold

Specify an integer value for row_threshold. Setting row_threshold to 0 is same as setting FEEDBACK to OFF. Setting row_threshold equal 1 effectively sets FEEDBACK to ON.

フラッシュを設定

ディスプレイのバッファリングを制御するには、SETFLUSH コマンドを使用します。

```
SET FLU[SH] {ON | OFF}
```

ディスプレイのバッファリングを有効にするには、FLUSH を OFF に設定します。バッファリングを有効にすると、スクリプトが完了するまで、画面向けのメッセージが表示されない場合があります。FLUSH を OFF に設定すると、パフォーマンスが向上することに注意してください。

ディスプレイのバッファリングを無効にするには、FLUSH を ON に設定します。バッファリングを無効にすると、画面向けのメッセージがすぐに表示されます。

見出しを設定

SETHEADING 変数を使用して、AdvancedServer が SELECT ステートメントの列見出しを表示するかどうかを指定します。

```
SET HEA[DING] {ON | OFF}
```

ヘッドセパレータを設定

SETHEADSEP コマンドは、COLUMNHEADING コマンドで使用される新しい見出し区切り文字を設定します。デフォルトは''l'' です。

```
SET HEADS[EP]
```

ラインサイズを設定

行の幅を文字数で指定するには、SETLINESIZE コマンドを使用します。

```
SET LIN[ESIZE] <width_of_line>
```

width_of_line

width_of_lineのデフォルト値は 132 です。

SETNEWPAGE

SETNEWPAGE コマンドを使用して、改ページ後に印刷される空白行の数を指定します。

```
SET NEWP[AGE] <lines_per_page>
```

lines_per_page

lines_per_page のデフォルト値は 1 です。

NULL を設定

SETNULL コマンドを使用して、出力列に NULL 列値が表示されるときにユーザーに表示される文字列を指定します。

```
SET NULL <null_string>
```

ページサイズを設定

SETPAGESIZE コマンドを使用して、ページに収まる印刷行数を指定します。

```
SET PAGES[IZE] <line_count>
```

line_count パラメータを使用して、ページあたりの行数を指定します。

SQLCASE を設定する

SETSQLCASE コマンドは、サーバーに送信される SQL ステートメントを大文字または小文字に変換するかどうかを指定します。

```
SET SQLC[ASE] {MIX[ED] | UP[PER] | LO[WER]}
```

アップパー

コマンドテキストを大文字に変換するには、UPPER を指定します。

低い

コマンドテキストを小文字に変換するには、LOWER を指定します。

混合

SQL コマンドの大文字小文字を変更しない場合は、MIXED を指定します。デフォルトは MIXED です。

一時停止を設定

SETPAUSE コマンドは、スクリプトに含まれている場合に最も役立ちます。コマンドはプロンプトを表示し、ユーザーが Return キーを押すのを待ちます。

```
SET PAU[SE] {ON | OFF}
```

SETPAUSE が ON の場合、各コマンドが実行される前に、HitENTERtocontinue…というメッセージが表示されます。

スペースを設定

列間に表示するスペースの数を指定するには、SETSPACE コマンドを使用します。

```
SET SPACE <number_of_spaces>
```

SETSQLPROMPT

SETSQLPROMPT を使用して、ユーザーインタラクティブプロンプトの値を設定します。

```
SET SQLP[ROMPT] /"<prompt>/"
```

デフォルトでは、SQLPROMPT は"/"SQL>/""に設定されています

期限を設定

SETTERMOUT コマンドを使用して、コマンド出力を画面に表示するかどうかを指定します。

```
SET TERM[OUT] {ON | OFF}
```

タイミングを設定する

SETTIMING コマンドは、実行後に AdvancedServer が各 SQL ステートメントの実行時間を表示するかどうかを指定します。

```
SET TIMI[NG] {ON | OFF}
```

トリムスプールを設定

SETTRIMSPPOOL コマンドを使用して、SPOOL コマンドで指定された出力ファイルの各行から末尾のスペースを削除します。

```
SET TRIMS[POOL] {ON | OFF}
```

デフォルト値はオフです。

確認を設定

置換変数が検出されたときに、SQL ステートメントの古い値と新しい値の両方を表示するかどうかを指定します。

```
SET VER[IFY] { ON | OFF }
```

5.26 SHOW

SHOW コマンドを使用して、現在のパラメーター値を表示します。

```
SHO[W] {ALL | ``パラメータ名 ``}
```

ALL キーワードを含めることにより、現在のパラメーター設定を表示します。

```
SQL> SHOW ALL
autocommit OFF
colsep " "
define "&"
echo OFF
FEEDBACK ON for 6 row(s).
flush ON
heading ON
headsep "|"
linesize 78
newpage 1
null " "
pagesize 14
pause OFF
serveroutput OFF
spool OFF
sqlcase MIXED
sqlprompt "SQL> "
sqlterminator ";"
suffix ".sql"
termout ON
timing OFF
verify ON
USER is "enterprisedb"
HOST is "localhost"
PORT is "5444"
DATABASE is "edb"
VERSION is "13.0.0"
```

または、SHOW コマンドに parameter_name を含めて、特定のパラメータ設定を表示します。

```
SQL> SHOW VERSION
VERSION is "13.0.0"
```

5.27 SPOOL

SPOOL コマンドは、ディスプレイからの出力をファイルに送信します。

```
SP[OOL] <output_file> | OFF
```

output_file パラメータを使用して、出力ファイルのパス名を指定します。

5.28 START

START コマンドを使用して、EDB*Plus スクリプトファイルを実行します。START は@コマンドのエイリアスです。

```
STA[RT] <script_file>
```

script_file パラメータにスクリプトファイルの名前を指定します。

5.29 UNDEFINE

UNDEFINE コマンドは、DEFINE コマンドによって作成されたユーザー変数を消去します。

```
UNDEF[INE] <variable_name> [ <variable_name>...]
```

変数の名前を指定するには、variable_name パラメータを使用します。

5.30 WHENEVER SQLERROR

WHENEVERSQLERROR コマンドは、SQL エラーまたは PL/SQL ブロックエラーのエラー処理を提供します。構文は次のとおりです。

```
WHENEVER SQLERROR
{ CONTINUE [ COMMIT | ROLLBACK | NONE ]
```

(次のページに続く)

(前のページからの続き)

EXIT[SUCCESS FAILURE WARNING <n> <sub_variable>] [COMMIT ROLLBACK]}
--

AdvancedServer が SQL コマンドまたは PL/SQL ブロックの実行中にエラーを検出した場合、EDB*Plus は **WHENEVERSQLERROR** コマンドで指定されたアクションを実行します。

続行する前に指定されたアクションを実行するよう EDB*Plus に指示するには、**CONTINUE** 句を含めます。

終了または続行する前に現在のトランザクションを **COMMIT** するよう EDB*Plus に指示するには、**COMMIT** 句を含めます。

終了または続行する前に現在のトランザクションを **ROLLBACK** するよう EDB*Plus に指示するには、**ROLLBACK** 句を含めます。

トランザクションをコミットまたはロールバックせずに続行するよう EDB*Plus に指示するには、**NONE** 句を含めます。

EDB*Plus に指定したアクションを実行し、エラーが発生した場合は終了するように指示するには、**EXIT** 句を含めます。

次のオプションを使用して、EDB*Plus が終了する前に返すステータスコードを指定します。

```
[SUCCESS|FAILURE|WARNING|<n>|<sub_variable>]
```

EDB*Plus は置換変数をサポートしていますが、バインド変数はサポートしていないことに注意してください。

第 6

結論

EDB*Plus ユーザーガイド

Copyright©2007-2020EnterpriseDBCorporation。全著作権所有。

EnterpriseDB®Corporation34CrosbyDrive、Suite201、Bedford、MA01730、USA

電話番号 +17813573390

F+19784671307E

info@enterprisedb.com

www.enterprisedb.com

- EDB は、EDB 製品が存在する場合、ログオン UI の設計、コーディングのベストプラクティスの確立、レビュー、および検証を行います。EDB は、追加の入力コンポーネントについても同じアプローチに従いますが、製品の性質上、制限付きの検証が可能な信頼できるユーザーが入力および送信するフリーフォーム SQL、WMI、またはその他の文字列を受け入れることが必要になる場合があります。このような場合、ユーザーが誤った入力や危険な入力をするのを防ぐことはできません。
- EDB は、将来的に認証済みの信頼できるユーザーからのフリーフォーム SQL、WMI、またはその他の潜在的に危険な入力を受け入れる製品に機能を追加する権利を留保しますが、そのような

すべての機能が可能な限り最小限のリスクを提供するように設計およびテストされるようにします。可能、スーパーユーザーまたは同等の権限が必要です。

- EDB は、すべての潜在的な脅威を予測できること、または予測できることを保証するものではありません。したがって、私たちのプロセスでは、すべての潜在的な脆弱性が対処または検討されたことを完全に保証できません。

索引

``ACCEPT``, 28

APPEND, 29

CLEAR, 30

Configuring SSL for EDB*Plus Client, 22

CONNECT, 34

DEFINE, 35

DEL, 35

DESCRIBE, 37

DISCONNECT, 37

EDB*Plus, 2

EDB*Plus commands, 28

EDB*Plus の使用, 14

EDIT, 37

EXECUTE, 38

EXIT, 38

GET, 39

HELP, 39

HOST, 40

INPUT, 40

Installing EDB*Plus through graphical installer, 9

Installing EDB*Plus with an RPM, 5

LIST, 40

PASSWORD, 41

PROMPT, 42

QUIT, 42

REMARK, 42

SAVE, 43

SecureSocketsLayer (SSL) 接続の使用, 18

SET, 44

SHOW, 48

SPOOL, 49

SSL connection between EDB*PLUS and database, 25

START, 49

supported failover scenarios, 19, 29, 30, 42

UNDEFINE, 49

WHENEVER SQLERROR, 49

インストール EDB*プラス, 3

コマンドの要約, 28

結論, 51