



EDB

EDB ODBC Connector

リリース *13.2.0.02*

EnterpriseDB Corporation

2023 年 04 月 28 日

目次

第 1 章	Release notes	3
1.1	Version 13.02.0000.02	3
1.2	Version 13.02.0000.01	4
1.3	Version 13.01.0000.02	4
1.4	Version 13.01.0000.01	4
1.5	Version 13.00.0000.01	4
第 2 章	Product compatibility	5
第 3 章	Installing EDB ODBC Connector	7
3.1	Linux x86-64 (amd64)	7
3.1.1	Red Hat Enterprise Linux (RHEL) およびその派生物	7
3.1.2	SUSE Linux Enterprise (SLES)	7
3.1.3	Debian とその派生物	7
3.2	Linux Installing EDB ODBC Connector on Linux IBM Power (ppc64le)	8
3.2.1	Red Hat Enterprise Linux (RHEL) およびその派生物	8
3.2.2	SUSE Linux Enterprise (SLES)	8
3.3	Windows	8
3.3.1	Installing EDB ODBC Connector on Linux x86 (amd64)	8
3.3.2	Installing EDB ODBC Connector on Linux IBM Power (ppc64le)	16
3.3.3	Installing the ODBC Connector on Windows	18
3.3.4	Upgrading a Linux installation	20
第 4 章	Creating a data source	21
第 5 章	Connection properties	23
5.1	Windows でデータソース定義を追加する	30
5.2	Linux でデータソース定義を追加する	35
第 6 章	Driver functionality	37
6.1	SQLGetInfo()	37
6.2	接続属性	53
6.2.1	SQLGetConnectAttr()	53
6.2.2	SQLSetConnectAttr()	54
6.3	環境属性	56
6.3.1	SQLGetEnvAttr()	56
6.4	SQLSetEnvAttr()	57
6.5	ステートメント属性	58
6.5.1	SQLGetStmtAttr()	58
6.5.2	SQLSetStmtAttr()	59
6.6	エラー処理	60

6.6.1	SQLGetDiagRec()	60
6.7	サポートされている ODBC API 関数	61
6.8	サポートされているデータ型	64
6.9	ADO ユーザーの前提条件	64
6.10	スレッドセーフ	64
第 7 章	Scram compatibility	65

ODBC (Open Database Connectivity) は、クライアントアプリケーションが ODBC ドライバーを提供するデータベースに接続できるようにするプログラミングインターフェイスです。EDB ODBC Connector は、ODBC 準拠のクライアントアプリケーションが EDB Postgres Advanced Server データベースに接続できるようにするインターフェイスです。EDB ODBC Connector を使用すると、他のデータベースと連携するように設計されたアプリケーションを EDB Postgres Advanced Server で実行できます。ODBC コネクタは、クライアントアプリケーションが接続を確立し、クエリを送信し、EDB Postgres Advanced Server から結果を取得する方法を提供します。

ODBC Connector は一定のアプリケーションの移植性を提供しますが、移植性は制限されています。接続は提供しますが、コマンドの互換性は保証しません。別のデータベースで受け入れられるコマンドは、EDB Postgres Advanced Server では機能しない場合があります。

一般的な ODBC アプリケーションの主なコンポーネントは次のとおりです。

- ODBC のバインディングを持つ言語で書かれたクライアントアプリケーション
- Windows の名前付き接続または Linux の名前付き接続を含む `odbc.ini` を処理する ODBC アドミニストレーター
- データベース固有の ODBC ドライバー (ODBC Connector)
- ODBC 準拠サーバー (EDB Postgres Advanced Server)

ODBC のバインディングを持つ任意の言語でクライアントアプリケーションを作成できます。C、MS-Access、および C++ はほんの一部です。

第1章 Release notes

EDB ODBC コネクタのドキュメントには、EDB ODBC コネクタの最新バージョンが説明されています。

リリースノートでは、リリースの新機能について説明します。マイナーリリースまたはパッチリリースで新機能が導入されると、コンテンツ内のインジケータにより、新機能が導入されたバージョンが識別されます。

Version	Release date
<i>Version 13.02.0000.02</i>	2023 Feb 14
<i>Version 13.02.0000.01</i>	2022 May 17
<i>Version 13.01.0000.02</i>	2021 Dec 12
<i>Version 13.01.0000.01</i>	2021 Sep 14
<i>Version 13.00.0000.01</i>	2020 Nov 19

1.1 Version 13.02.0000.02

EDB ODBC Connector 13.02.0000.02 には、次の拡張機能が含まれています。

Type	Description
Enhancement	EDB Postgres Advanced Server v15 のサポートが追加されました。

!!!注以前の Ubuntu / Debian インストールからアップグレードする場合は、install コマンドを実行し、パッケージ名にバージョン番号を指定する必要があります。例：

```
sudo apt-get install edb-odbc=13.02.0000.02-1.bionic
```

ダウングレードに関するメッセージが表示された場合、無視できます。

1.2 Version 13.02.0000.01

EDB ODBC Connector 13.02.0000.01 には、次の機能強化が含まれています。

Type	Description
Upstream merge	アップストリームコミュニティドライババージョン 13.02.0000 とマージされました。詳細については、コミュニティ Release notes を参照してください。
Bug fix	MS SQL リンクサーバーデータベースの繰り返し [サポートチケット # 1044249]。

1.3 Version 13.01.0000.02

EDB ODBC Connector 13.01.0000.02 には、次の機能強化が含まれています。

Type	Description
Enhancement	EDB Postgres Advanced Server バージョン 14.1.0 のサポート。

1.4 Version 13.01.0000.01

EDB ODBC Connector 13.01.0000.01 には、次のアップストリームマージが含まれています。

Type	Description
Upstream merge	アップストリームコミュニティドライババージョン 13.01.0000 とマージされました。詳細については、コミュニティ Release notes を参照してください。

1.5 Version 13.00.0000.01

EDB ODBC Connector 13.00.0000.01 には、次のアップストリームマージが含まれます。

Type	Description
Upstream merge	アップストリームコミュニティドライババージョン 13.00.0000 とマージされました。詳細については、コミュニティ Release notes を参照してください。

第2章 Product compatibility

この表は、最新の ODBC Connector バージョンと、サポートされている対応する EDB Postgres Advanced Server (EPAS) バージョンをリストしています。詳細は [Product compatibility](#) を参照してください。

ODBC Connector	EPAS 15	EPAS 14	EPAS 13	EPAS 12	EPAS 11
13.02.0.02	Y	Y	Y	Y	Y
13.02.0.01	N	Y	Y	Y	Y
13.01.0.02	N	Y	Y	Y	Y
13.01.0.01	N	N	Y	Y	Y
13.00.0.01	N	N	Y	Y	Y

第3章 Installing EDB ODBC Connector

リンクを選択して、該当するインストール手順にアクセスします。

3.1 Linux x86-64 (amd64)

3.1.1 Red Hat Enterprise Linux (RHEL) およびその派生物

- *Installing EDB ODBC Connector on RHEL 8 or OL 8 x86_64*、*Installing EDB ODBC Connector on RHEL 7 or OL 7 x86_64*
- Oracle Linux (OL) 8、Oracle Linux (OL) 7
- *Installing EDB ODBC Connector on AlmaLinux 8 or Rocky Linux 8 x86_64*
- *Installing EDB ODBC Connector on AlmaLinux 8 or Rocky Linux 8 x86_64*
- *Installing EDB ODBC Connector on CentOS 7 x86_64*

3.1.2 SUSE Linux Enterprise (SLES)

- *Installing EDB ODBC Connector on SLES 15 x86_64*、*Installing EDB ODBC Connector on SLES 12 x86_64*

3.1.3 Debian とその派生物

- *Installing EDB ODBC Connector on Ubuntu 22.04 x86_64*、*Installing EDB ODBC Connector on Ubuntu 20.04 x86_64*、*Installing EDB ODBC Connector on Ubuntu 18.04 x86_64*
- *Installing EDB ODBC Connector on Debian 11 x86_64*、*Installing EDB ODBC Connector on Debian 10 x86_64*

3.2 Linux Installing EDB ODBC Connector on Linux IBM Power (ppc64le)

3.2.1 Red Hat Enterprise Linux (RHEL) およびその派生物

- *Installing EDB ODBC Connector on RHEL 8 or OL 8 x86_64*

3.2.2 SUSE Linux Enterprise (SLES)

- *Installing EDB ODBC Connector on SLES 15 x86_64* 、 *Installing EDB ODBC Connector on SLES 12 x86_64*

3.3 Windows

- *Windows Server 2019*

3.3.1 Installing EDB ODBC Connector on Linux x86 (amd64)

リポジトリへのアクセスを含む、オペレーティングシステム固有のインストール手順については、以下を参照してください。

Red Hat Enterprise Linux (RHEL) およびその派生物

- *Installing EDB ODBC Connector on RHEL 8 or OL 8 x86_64*
- *Installing EDB ODBC Connector on RHEL 7 or OL 7 x86_64*
- Oracle Linux (OL) 8
- Oracle Linux (OL) 7
- *Installing EDB ODBC Connector on AlmaLinux 8 or Rocky Linux 8 x86_64*
- *Installing EDB ODBC Connector on AlmaLinux 8 or Rocky Linux 8 x86_64*
- *Installing EDB ODBC Connector on CentOS 7 x86_64*

SUSE Linux Enterprise (SLES)

- *Installing EDB ODBC Connector on SLES 15 x86_64*
- *Installing EDB ODBC Connector on SLES 12 x86_64*

Debian とその派生物

- *Installing EDB ODBC Connector on Ubuntu 22.04 x86_64*
- *Installing EDB ODBC Connector on Ubuntu 20.04 x86_64*
- *Installing EDB ODBC Connector on Ubuntu 18.04 x86_64*
- *Installing EDB ODBC Connector on Debian 11 x86_64*
- *Installing EDB ODBC Connector on Debian 10 x86_64*

Installing EDB ODBC Connector on RHEL 8 or OL 8 x86_64

前提条件

インストールプロセスを開始する前に：

• EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- EPEL リポジトリをインストールします

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.  
→noarch.rpm
```

パッケージをインストールする

```
sudo dnf -y install edb-odbc
sudo dnf -y install edb-odbc-devel
```

Installing EDB ODBC Connector on AlmaLinux 8 or Rocky Linux 8 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

[EDB repositories](#) に移動し、そこに記載されている指示に従います。

- EPEL リポジトリをインストールします

```
sudo dnf -y install epel-release
```

- 追加のリポジトリを有効にして依存関係を解決します

```
sudo dnf config-manager --set-enabled powertools
```

パッケージをインストールする

```
sudo dnf -y install edb-odbc
sudo dnf -y install edb-odbc-devel
```

Installing EDB ODBC Connector on RHEL 7 or OL 7 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- EPEL リポジトリをインストールします

```
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.
→noarch.rpm
```

- 追加のリポジトリを有効にして依存関係を解決します

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-
→rpms" --enable "rhel-ha-for-rhel-*-server-rpms"
```

パッケージをインストールする

```
sudo yum -y install edb-odbc
sudo yum -y install edb-odbc-devel
```

Installing EDB ODBC Connector on CentOS 7 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- EPEL リポジトリをインストールします

```
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.
→noarch.rpm
```

パッケージをインストールする

```
sudo yum -y install edb-odbc
sudo yum -y install edb-odbc-devel
```

Installing EDB ODBC Connector on SLES 15 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- その他の前提条件に対処する

```
# Activate the required SUSE module
sudo SUSEConnect -p PackageHub/15.3/x86_64

# Refresh the metadata
sudo zypper refresh
```

パッケージをインストールする

```
sudo zypper -n install edb-odbc
sudo zypper -n install edb-odbc-devel
```

Installing EDB ODBC Connector on SLES 12 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- その他の前提条件に対処する

```
# Activate the required SUSE module
sudo SUSEConnect -p PackageHub/12.5/x86_64
sudo SUSEConnect -p sle-sdk/12.5/x86_64

# Refresh the metadata
sudo zypper refresh
```

パッケージをインストールする

```
sudo zypper -n install edb-odbc
sudo zypper -n install edb-odbc-devel
```

Installing EDB ODBC Connector on Ubuntu 22.04 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

パッケージをインストールする

```
sudo apt-get -y install edb-odbc
sudo apt-get -y install edb-odbc-dev
```

Installing EDB ODBC Connector on Ubuntu 20.04 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

パッケージをインストールする

```
sudo apt-get -y install edb-odbc
sudo apt-get -y install edb-odbc-dev
```

Installing EDB ODBC Connector on Ubuntu 18.04 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

パッケージをインストールする

```
sudo apt-get -y install edb-odbc
sudo apt-get -y install edb-odbc-dev
```

Installing EDB ODBC Connector on Debian 11 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

パッケージをインストールする

```
sudo apt-get -y install edb-odbc
sudo apt-get -y install edb-odbc-dev
```

Installing EDB ODBC Connector on Debian 10 x86_64

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

パッケージをインストールする

```
sudo apt-get -y install edb-odbc
sudo apt-get -y install edb-odbc-dev
```

3.3.2 Installing EDB ODBC Connector on Linux IBM Power (ppc64le)

リポジトリへのアクセスを含む、オペレーティングシステム固有のインストール手順については、以下を参照してください。

Red Hat Enterprise Linux (RHEL)

- *Installing EDB ODBC Connector on RHEL 8 or OL 8 x86_64*

SUSE Linux Enterprise (SLES)

- *Installing EDB ODBC Connector on SLES 15 x86_64*
- *Installing EDB ODBC Connector on SLES 12 x86_64*

Installing EDB ODBC Connector on RHEL 8 ppc64le

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- EPEL リポジトリをインストールします

```
sudo dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.
↪noarch.rpm
```

- キャッシュを更新する

```
sudo dnf makecache
```

パッケージをインストールする

```
sudo dnf -y install edb-odbc  
sudo dnf -y install edb-odbc-devel
```

Installing EDB ODBC Connector on SLES 15 ppc64le

前提条件

インストールプロセスを開始する前に：

- EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは1回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- その他の前提条件に対処する

```
# Activate the required SUSE module  
sudo SUSEConnect -p PackageHub/15.3/ppc64le  
  
# Refresh the metadata  
sudo zypper refresh
```

パッケージをインストールする

```
sudo zypper -n install edb-odbc  
sudo zypper -n install edb-odbc-devel
```

Installing EDB ODBC Connector on SLES 12 ppc64le

前提条件

インストールプロセスを開始する前に：

• EDB Postgres Advanced Server を、接続文字列で接続可能なホストにインストールします。同じホスト上にある必要はありません。

Installing EDB Postgres Advanced Server を参照してください。

- リポジトリをセットアップする

リポジトリのセットアップは 1 回限りのタスクです。リポジトリをすでに設定している場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

EDB repositories に移動し、そこに記載されている指示に従います。

- その他の前提条件に対処する

```
# Activate the required SUSE module
sudo SUSEConnect -p PackageHub/12.5/ppc64le
sudo SUSEConnect -p sle-sdk/12.5/ppc64le

# Refresh the metadata
sudo zypper refresh
```

パッケージをインストールする

```
sudo zypper -n install edb-odbc
sudo zypper -n install edb-odbc-devel
```

3.3.3 Installing the ODBC Connector on Windows

EDB は、Windows のグラフィカルな対話型インストーラーを提供します。次の 2 つの方法でアクセスできます。

- [Downloads page](#) からグラフィカルインストーラーをダウンロードし、インストーラーを直接起動します。[直接インストール](#) を参照してください。
- Stack Builder (PostgreSQL を使用) または StackBuilder Plus (EDB Postgres Advanced Server を使用) を使用して、EDB インストーラーパッケージをダウンロードし、グラフィカルインストーラーを起動します。

[Stack Builder または StackBuilder Plus の使用](#) を参照してください。

直接インストール

グラフィカルインストーラーをダウンロードした後、インストールウィザードを起動して、十分な特権（スーパーユーザーまたは管理者）を引き受け、インストーラーアイコンをダブルクリックします。プロンプトが表示されたら、パスワードを入力します。

Windows の一部のバージョンでは、管理者権限でインストーラーを起動するには、インストーラーアイコンを右クリックし、コンテキストメニューから **RunasAdministrator** を選択する必要があります。

[グラフィカルインストーラーの使用](#) に進みます。

Stack Builder または StackBuilder Plus の使用

PostgreSQL を使用している場合は、Stack Builder でグラフィカルインストーラーを呼び出すことができます。

[Stack Builder または StackBuilder Plus の使用](#) を参照してください。

EDB Postgres Advanced Server を使用している場合、StackBuilder Plus でグラフィカルインストーラーを呼び出すことができます。

Using StackBuilder Plus を参照してください。

1. Stack Builder または StackBuilder Plus で、モジュール選択ページが表示されるまでプロンプトに従います。

Welcome ページで、使用可能なサーバーのリストからターゲットサーバーのインストールを選択します。ネットワークでプロキシサーバーを使用してインターネットにアクセスする必要がある場合は、[**プロキシサーバー**] を選択してサーバーを指定します。 **Next** を選択します。

1. データベースドライバ ノードを展開し、次のいずれかを実行します。

- Stack Builder で、 **pgODBC** を選択します。
- StackBuilder Plus で、 **EnterpriseDBODBCConnector** を選択します。

1. [グラフィカルインストーラーの使用](#) に進みます。

グラフィカルインストーラーの使用

1. インストール言語を選択し、 **OK** を選択します。

1. [Setup ODBC] ページで、[**Next**] を選択します。

2. ODBC をインストールするディレクトリを参照するか、ディレクトリ設定をデフォルトの場所のままにします。 **Next** を選択します。

3. [Ready to Install] ページで、[**Next**] を選択します。

情報ボックスには、選択したコンポーネントのインストールの進行状況が表示されます。

1. インストールが完了したら、 **Finish** を選択します。

3.3.4 Upgrading a Linux installation

RPM インストールからのアップグレード

既存の EDB ODBC コネクタ RPM インストールがある場合は、yum または dnf を使用してリポジトリ構成ファイルをアップグレードし、より新しい製品バージョンに更新できます。edb.repo ファイルを更新するには、スーパーユーザー権限を引き受けて、次のように入力します。

- RHEL または CentOS 7 の場合：

```
yum upgrade edb-repo
```

- RHEL または Rocky Linux または AlmaLinux 8 の場合：

```
dnf upgrade edb-repo
```

yum または dnf は edb.repo ファイルを更新して、edb.repo ファイルで指定された資格情報で接続するように構成された現在の EDB リポジトリへのアクセスを有効にします。次に、yum または dnf を使用して、インストールされているパッケージをアップグレードできます。

- RHEL または CentOS 7 の場合：

```
yum upgrade edb-odbc
```

```
yum upgrade edb-odbc-devel
```

- RHEL または Rocky Linux または AlmaLinux 8 の場合：

```
dnf upgrade edb-odbc
```

```
dnf upgrade edb-odbc-devel
```

Debian / Ubuntu インストールからのアップグレード

以前の Debian / Ubuntu インストールからアップグレードする場合は、install コマンドを実行し、パッケージ名にバージョン番号を指定します。例：

```
sudo apt-get install edb-odbc=13.02.0000.02-1.bionic
```

ダウングレードに関するメッセージが表示された場合、無視できます。

第4章 Creating a data source

クライアントアプリケーションがサーバーとの接続を確立しようとするとき、通常、データソース名 (DSN) を提供します。ドライバーマネージャは、ODBC 構成データベースを調べて、アプリケーションが提供する DSN と名前が一致するデータソースを探します。

Linux または Unix ホストでは、データソースは通常 `/etc/odbc.ini` という名前のファイルで定義されます。名前と場所は異なる場合があります。次のコマンドを使用して、unixODBC がデータソース定義を検索している場所を見つけます。

```
$ odbc_config --odbcini --odbcinstini
```

Windows ホストでは、データソースは通常、Windows レジストリで定義されます。

ファイル DSN と呼ばれるデータソース定義をプレーンテキストファイルに保存することもできます。EDB-ODBC ドライバの一般的なデータソース定義は次のようになります。

```
$ cat /etc/odbc.ini
[EnterpriseDB]
Description = EnterpriseDB DSN
Driver = EnterpriseDB
Trace = yes
TraceFile = /tmp/odbc.log
Database = edb
Servername = localhost
UserName = enterprisedb
Password = manager
Port = 5444
```

データソースの最初の行はデータソース名です。名前は、角括弧で囲まれた一意の識別子です。データソース名の後には、データソースを構成する個々の接続プロパティを識別する一連の **keyword=value** ペアが続きます。

Windows の ODBC アドミニストレータユーティリティは、ODBC 接続用の名前付きデータソースを作成します。Windows を使用している場合、このツールは ODBC データソースアドミニストレータと呼ばれます。Linux を使用している場合、選択したテキストエディターで `odbc.ini` ファイルを使用して、名前付きデータソースを作成できます。

Windows でデータソース定義を追加する および *Linux でデータソース定義を追加する* は、

Windows でグラフィカルツールを使用して、Linux でコマンドラインを使用してデータソースを追加する手順を説明します。データソースを定義するプロセス中に、一連の接続プロパティを指定する必要があります。EDB-ODBC 接続プロパティには、オプションのデータソース接続プロパティに関する情報が含まれ

ています。グラフィカルツール（Windows の場合）を使用して接続プロパティを指定するか、テキストエディター（Linux の場合）で `odbc.ini` ファイルを編集できます。

第5章 Connection properties

次の表では、グラフィカル接続マネージャーツールのダイアログボックス、または名前付きデータソースを定義する `odbc.ini` ファイルで指定できる接続プロパティについて説明します。列は、接続プロパティ (ODBC アドミニストレータのダイアログに表示される)、対応するキーワード (`odbc.ini` ファイルに表示される)、プロパティのデフォルト値、および接続プロパティの説明を識別します。

Property	Keyword name	Default value	Description
Database	Database	None	接続しているデータベースの名前。
Driver	Driver	EDB-ODBC	ODBC ドライバーの名前。
Server	Servename	Localhost	接続しているサーバーの名前または IP アドレス。
dbms_name	dbms_name	EnterpriseDB	データベースシステム。 EnterpriseDB または PostgreSQL のいずれか。
Description	Description		データソースの説明的な名前。
User Name	Username		このデータソースがサーバーへの接続に使用するユーザーの名前。
Password	Password		この名前付きデータソースに関連付けられたユーザーのパスワード。
CPTimeout	CPTimeout	0	接続がタイムアウトするまでの秒数 (接続プーリング環境)。
Port	Port	5444	postmaster がリッスンしている TCP ポート。
Protocol	Protocol	7.4	指定すると、ドライバは指定されたプロトコルバージョンを使用します。
Level of Rollback on Errors	Use the <i>Protocol</i> option to specify rollback behavior.	Transaction Level	ドライバがエラーを処理する方法を指定します。 0-ロールバックしない 1-トランザクションをロールバックする 2-ステートメントをロールバックします
Usage Count	UsageCount	1	このドライバを使用したインストールの数。
Read Only	ReadOnly	No	接続が READONLY であることを指定します。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Show System Tables	ShowSystemTables	No	有効にすると、ドライバーは `SQLTables()` 関数の結果セットでシステムテーブルを報告します。
OID Options: Show Column	ShowOidColumn	No	有効になっている場合、 `SQLColumns()` 関数は OID 列を報告します。
OID Options: Fake Index	FakeOidIndex	No	有効にすると、 `SQLStatistics()` 関数は、各 OID 列に一意のインデックスが存在することを報告します。
Keyset Query Optimization	Ksqo	On	有効にすると、キーセットクエリ (MS Jet データベースエンジンによって生成される) のサーバー側サポートが強制されます。
Recognize Unique Indexes	UniqueIndex	On	有効にすると、 `SQLStatistics()` 関数は一意のインデックスを報告します。有効になっていない場合、 `SQLStatistics()` 関数は、インデックスが値の重複を許可することを報告します。
Use Declare/Fetch	UseDeclareFetch	Off	有効にすると、ドライバーはサーバー側カーソルを使用します。 `UseDeclareFetch` を有効にするには、 `1` の値を指定します。 `UseDeclareFetch` を無効にするには、 `0` の値を指定します。
CommLog	CommLog	Off	有効にすると、すべてのクライアント/サーバートラフィックをログファイルに記録します。
Parse Statements	Parse	Off	有効にすると、ドライバーは `SQLNumResultCols()`、 `SQLDescribeCol()`、または `SQLColAttributes()` 関数を呼び出すときに、単純な `SELECT` ステートメントを解析します。
Cancel as FreeStmt	CancelAsFreeStmt	Off	有効にすると、 `SQLCancel()` フังก์ションはあなたに代わって `SQLFreeStmt(SQL_Close)` を呼び出します。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
MyLog	Debug	Off	有効にすると、ドライバーはその作業をログファイルに記録します。Windows では、ファイル名は `C:\mylog<process-id>` です。Linux では、ファイル名は `/tmp/mylog<username><process-id>.log` です。
Unknown Sizes	UnknownSizes	Maximum	`SQLDescribeCol()` および `SQLColAttributes()` 関数が列のサイズを計算する方法を決定します。 `0` を指定して、ドライバーにタイプに許可されている最大サイズを報告させます。 `1` を指定してドライバーに不明な長さを報告させるか、 `2` を指定してドライバーに結果セットを検索させて最長値を見つけさせます。 `UseDeclareFetch` を有効にした場合、 `2` を指定しないでください。
Text as Long Varchar	TextAsLongVarChar	8190	有効にすると、ドライバーは TEXT 列を `SQL_LONGVARCHAR` 型であるかのように扱います。無効にすると、ドライバーは TEXT 列を `SQL_VARCHAR` 値として扱います。
Unknown as Long Varchar	LongVarChar	False	有効にすると、ドライバーは不明なタイプの値を `SQL_LONGVARCHAR` 値として扱います。無効にすると、ドライバーは不明なタイプの値を `SQL_VARCHAR` 値として扱います。デフォルトでは、不明な型の値は Y 値として扱われます。
Bools as Char	BoolsAsChar	On	有効にすると、ドライバーは BOOL 列を `SQL_CHAR` 値として扱います。無効にすると、BOOL 列は `SQL_BIT` 値として扱われます。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Max Varchar	MaxVarcharSize	255	有効にすると、ドライバーは `MaxVarCharSize` よりも長い `VARCHAR` および `BPCHAR` 値を `SQL_LONGVARCHAR` 値として扱います
Max Long Varchar Size	MaxLongVarchar-Size	8190	`TextAsLongVarChar` がオンの場合、ドライバーは TEXT 値が `MaxLongVarcharSize` バイト長であると報告します。

 `UnknownAsLongVarChar` がオンの場合、不明な型の列は `MaxLongVarcharSize` バイト長です。それ以外の場合、長さは `MaxVarcharSize` バイトであると報告されます。
Cache Size	Fetch	100	`UseDeclareFetch` が有効になっているときにドライバーによってフェッチされる行数を決定します。
SysTable Prefixes	ExtraSysTablePrefixes	<i>Windows</i> でデータソース定義を追加する ;	<i>SysTablePrefixes</i> フィールドを使用して、テーブルがシステムテーブルであることを示すプレフィックスのセミコロン区切りリストを指定します。デフォルトでは、リストには <i>Windows</i> でデータソース定義を追加する ; が含まれます。
Cumulative Row Count for Insert	MapSqlParcNo-Batch	Off/0	有効になっている場合、`SQLRowCount()` 関数は、`INSERT` ステートメントのパラメーター設定の配列全体の単一の累積行カウントを返します。無効にすると、パラメーター設定ごとに個別の行カウントが返されます。デフォルトでは、このオプションは無効になっています。
LF<-> CR/LF conversion	LFConversion	System Dependent	LF<->CR/LF 変換オプションは、サーバーから文字値をフェッチするときにラインフィード文字をキャリッジリターン/ラインフィードのペアに変換し、文字値をサーバー。デフォルトでは、このオプションは有効になっています。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Updatable Cursors	UpdatableCursors	Off	`SQLSetPos()`または`SQLBulkOperations()`関数を使用して、位置指定`UPDATE`および`DELETE`操作を許可します。
Bytea as Long VarBinary	ByteaAsLong-VarBinary	Off	有効にすると、ドライバーは BYTEA 値を`SQL_LONGVARBINARY`型であるかのように扱います。無効にすると、BYTEA 値は`SQL_VARBINARY`値として扱われます。
Bytea as LO	ByteaAsLO	False	有効にすると、ドライバーは BYTEA 値をラージオブジェクトであるかのように扱います。
Row versioning	RowVersioning	Off	<i>Row Versioning</i> オプションは、テーブル内の列を報告するときにドライバーが`xmin`列を含めるかどうかを指定します。`xmin`値は、行を作成したトランザクションの ID です。`SQL_CONCURRENCY = SQL_CONCUR_ROWVER.1`でカーソルを作成する予定がある場合は、行バージョン管理を使用する必要があります。
Disallow Premature	DisallowPremature	No/0	クエリを実行せずにクエリに関する情報を取得しようとした場合のドライバーの動作を決定します。`Yes`の場合、ドライバーはクエリのカーソルを宣言し、カーソルからメタデータをフェッチします。`No`の場合、ドライバーはメタデータを要求するとすぐにコマンドを実行します。
True is -1	TrueIsMinus1	Off/0	<i>TrueIsMinus1</i> は、ドライバーに`-1`として TRUE の BOOL 値を返すように指示します。このオプションが有効になっていない場合、ドライバーは`1`として TRUE の BOOL 値を返します。ドライバーは、常に FALSE の BOOL 値を`0`として返します。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Server side prepare	UseServerSidePre- pare	No/0	有効にすると、ドライバーは`PRE-PARE`および`EXECUTE`コマンドを使用して準備/実行モデルを実装します。
Use GSSAPI for GSS re- quest	GssAuthUseGSS	False/0	`True/1`に設定されている場合、ドライバーは GSSAPI 認証要求をサーバーに送信します。Windows のみ。
Int8 As	BI	0	<i>BI</i> の値は、ドライバーが <i>BIGINT</i> の値をどのように扱うかを決定します。 `-5 as a SQL_BIGINT`、 `2 as a SQL_NUMERIC`、 `8 as a SQL_DOUBLE`、 `4 as a SQL_INTEGER`、 `12 as a SQL_VARCHAR`、 `0` (MS Jet クライアント上) 、`SQL_NUMERIC`として、 他のクライアントで `0` の場合、`SQL_BIGINT`として。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Extra options />>Connect Settings	AB />>ConnSettings	0x0	0x1-短い長さの形式の接続文字列の出力を強制します。 MFC CDatabase クラスを使用している場合は、このオプションを指定します。 0x2 - MS Access が PostgreSQL のシリアルタイプをオートナンバー型として認識できるようにします。 0x4 - ANSI 文字タイプを返しますアプリケーションからの問い合わせ。 Unicode データの処理が難しいアプリケーションの場合は、このオプションを指定します。 0x8 - 設定すると、NULL 日付は空の文字列として報告され、空の文字列は入力時に NULL 日付として解釈されます。 0x10- 'SQLGetInfo`がすべてのテーブルに関する情報を返すか、アクセス可能なテーブルのみを返すかを決定します。設定されている場合、アクセス可能なテーブルの情報のみが返されます。 0x20-設定されている場合、各 SQL コマンドは個別のネットワークラウンドトリップで処理されます。それ以外の場合、SQL コマンドは可能な限り少ないラウンドトリップにグループ化され、ネットワークの待機時間が短縮されます。ドライバーがサーバーに接続するときに実行される SQL コマンドのセミコロン区切りのリストが含まれています。ドライバーがクライアントに接続するために使用するバッファサイズを指定します。
	Socket	4096	
	Lie	Off	有効にすると、ドライバーはサポートされていない ODBC 機能をサポートしていると主張します。
Lowercase Identifier	LowerCaseIdentifier	Off	有効にすると、ドライバーは識別子を小文字に変換します。

次のページに続く

表 1 – 前のページからの続き

Property	Keyword name	Default value	Description
Disable Genetic Optimizer	Optimizer	Yes/1	遺伝的クエリオプティマイザーを無効にします。
Allow Keyset	UpdatableCursors	Yes/1	キーセットドリブンカーソルを許可する
SSL mode	SSLMode	Disabled	libpq（およびその依存関係）が EDB-ODBC ドライバーと同じディレクトリにインストールされている場合、SSL モードを有効にすると、SSL およびその他のユーティリティを使用できます。
Force Abbreviated Connection String	CX	No/0	接続文字列の省略形を強制するオプションを有効にします。
Fake MSS	FakeOidIndex	No/0	MS SQL Server になります、MS Access が PostgreSQL のシリアルタイプをオートナンバータイプとして認識できるようにします。
BDE Environment	BDE	No/0	このオプションを有効にすると、EDB-ODBC が Borland Database Engine 準拠の出力 (Unicode 関連) に対応するように調整されます。
XA_Opt	INI_XAOPT	Yes/1	有効になっている場合、`SQL_TABLES` への呼び出しには、ユーザーがアクセス可能なテーブルのみが含まれます。

5.1 Windows でデータソース定義を追加する

Windows ODBC データソースアドミニストレーターは、名前付きデータソースを作成するグラフィカルインターフェイスです。ODBC データソースアドミニストレーターを開くには、コントロールパネルで、**管理ツール**メニューを開き、適切な ODBC データソースアイコン（32 ビットまたは 64 ビット）をダブルクリックします。

追加を選択して、[新しいデータソースの作成] ダイアログボックスを開きます。ドライバーのリストから **EnterpriseDB(ANSI)** または **EnterpriseDB(UNICODE)** を選択し、**Finish** を選択します。

EnterpriseDB ODBC Driver ダイアログボックスのフィールドを使用して、名前付きデータソースを定義します。

- **データベース** フィールドにデータベース名を入力します。
- **Server** フィールドに EDB Postgres Advanced Server のホスト名または IP アドレスを入力します。
- **UserName** フィールドにユーザーの名前を入力します。
- **Description** フィールドに、名前付きデータソースの説明的な名前を入力します。

- libpq が EDB-ODBC ドライバーと同じディレクトリにインストールされている場合、**SSLMode** ラベルの横にあるリストがアクティブになり、SSL および他の EDB Postgres Advanced Server ユーティリティを使用できます。
- デフォルトのポート番号 (5444) を受け入れるか、**ポート** フィールドに別の番号を入力します。
- **Password** フィールドにユーザーのパスワードを入力します。

(オプションボックスにあります) **Datasource** を選択して [Advanced Options] ダイアログボックスを開き、接続プロパティを指定します。

Global を選択して、EDB-ODBC ドライバー (データソースではなくドライバー) のログオプションを指定できるダイアログを開きます。

- **GeneticOptimizer** を無効にするを選択して、ジェネティッククエリオプティマイザーを無効にします。デフォルトでは、クエリオプティマイザーはオンになっています。
- **KSQO (キーセットクエリ最適化)** を選択して、キーセットクエリのサーバー側サポートを有効にします。デフォルトでは、**キーセットクエリの最適化** はオンです。
- **RecognizeUniqueIndexes** を選択して、SQLStatistics() ファンクションに一意のインデックスを報告させます。このオプションが選択されていない場合、SQLStatistics() 関数は、すべてのインデックスが重複値を許可することを報告します。デフォルトでは、**RecognizeUniqueIndexes** はオンになっています。
- **UseDeclare/Fetch** を選択して、アプリケーションが SELECT コマンドを実行するたびにドライバーがサーバー側カーソルを使用するように指定します。デフォルトでは、**UseDeclare/Fetch** はオフになっています。
- ****CommLog(C::raw-latex:\psqlodbc_xxxx.log)**** を選択して、すべてのクライアント/サーバートラフィックをログファイルに記録します。デフォルトでは、ログ記録はオフになっています。
- **ParseStatements** を選択して、ドライバー (サーバーではなく) に指定し、SQLNumResultCols()、SQLDescribeCol()、または SQLColAttributes() 関数を呼び出すときに単純な SELECT ステートメントの解析を試みます。デフォルトでは、このオプションはオフです。
- **CancelasFreeStmt (Exp)** を選択して、SQLCancel() 関数があなたに代わって SQLFreeStmt(SQLClose) を呼び出すように指定します。デフォルトでは、このオプションはオフです。
- ****MyLog(C::raw-latex:\mylog_xxxx.log)**** を選択して、ドライバーアクティビティの詳細記録をログファイルに記録します。ログファイルの名前は c:\mylog_\ *process-id*.log です。デフォルトでは、ログ記録はオフになっています。

UnknownSizes ボックスのラジオボタンは、SQLDescribeCol() および SQLColAttributes() 関数が不明な型の列のサイズを計算する方法を指定します。

- **Maximum** を選択して、ドライバーが VARCHAR または LONGVARCHAR に許可される最大サイズを報告するように指定します (**UnknownsasLongVarChar** 設定に依存します)。UnknownsasLongVarChar が有効になっている場合、ドライバーは LONGVARCHAR の最大サイズを返します (**Miscellaneous** ボックスの **MaxLongVarChar** フィールドで指定されます)。UnknownsasLongVarChar がクリアされている場合、ドライバーは **Miscellaneous** ボックスの **MaxVarChar** フィールドで指定されたサイズを返します。

- **Don't know** を選択して、ドライバーが **不明** の長さを報告するように指定します。
- **最長** を選択して、ドライバーが結果セットを検索し、見つかった最長値を報告するように指定します。**UseDeclareFetch** が有効になっている場合は、**Longest** を指定しないでください。）

DataTypeOptions ボックスのプロパティは、ドライバーが特定の種類の列を処理する方法を決定します。

- **TextasLongVarChar** を選択して、TEXT 値を SQL_LONGVARCHAR 型であるかのように扱います。クリアすると、ドライバーは TEXT 値を SQL_VARCHAR 値として扱います。デフォルトでは、TEXT 値は SQL_LONGVARCHAR 値として扱われます。
- **UnknownsasLongVarChar** を選択して、ドライバーが不明なタイプの値を SQL_LONGVARCHAR 値として扱うように指定します。クリアすると、ドライバーは不明なタイプの値を SQL_VARCHAR 値として扱います。デフォルトでは、不明な型の値は SQL_VARCHAR 値として扱われます。
- **BoolsasChar** を選択して、ドライバーが BOOL 値を SQL_CHAR 値として扱うように指定します。クリアすると、BOOL 値は SQL_BIT 値として扱われます。デフォルトでは、BOOL 値は SQL_CHAR 値として扱われます。

Miscellaneous ボックスのフィールドで、名前付けデータソースに関連付けられたプロパティの一部の値を指定できます。

- **MaxVarChar** フィールドの VARCHAR 値に許可される最大長を示します。デフォルトでは、この値は 255 に設定されています。
- **MaxLongVarChar** フィールドに LONGVARCHAR 値に許可される最大長を入力します。デフォルトでは、この値は 8190 に設定されています。
- **CacheSize** フィールドで、ドライバー（UseDeclareFetch が有効な場合）によってフェッチされる行数を指定します。デフォルト値は 100 です。
- **SysTablePrefixes** フィールドを使用して、テーブルがシステムテーブルであることを示すプレフィックスのセミコロン区切りリストを指定します。デフォルトでは、リストには **dd_;** が含まれます。

Defaults を選択すると、このダイアログボックスの値をデフォルト設定にリセットできます。

適用 ボタンを選択して、データソースのプロパティに変更を適用します。**OK** を選択して変更を適用し、終了します。

（[Advanced Options] ダイアログボックスの左上隅にある） **Page2** を選択して、2 番目の詳細オプションセットにアクセスします。

- **ReadOnly** を選択して、ドライバーが次のコマンドを実行しないようにします：INSERT、UPDATE、DELETE、CREATE、ALTER、DROP、GRANT、REVOKE または LOCK。**ReadOnly** オプションを呼び出すと、ODBC のプロシージャコールエスケープ構文（call=procedure-name?）を使用する呼び出しも防止されます。デフォルトでは、このオプションはオフです。
- **ShowSystemTables** を選択して、SQLTables() 関数の結果セットにシステムテーブルを含めます。このオプションが有効になっている場合、ドライバーには、名前が pg_ で始まるテーブル、または [高度なオプション] ダイアログボックスの 1 ページの **SysTablePrefixes** フィールドにリストされているプレフィックスのいずれかが含まれます。デフォルトでは、このオプションはオフです。
- **Showsys/dboTables[Access]** を選択して、ODBC データソースを介して sys スキーマおよび dbo スキーマのオブジェクトにアクセスします。デフォルトでは、このオプションはオンです。

- **Showsys/SQLTables** を選択して、MS SQL でリンクサーバーのデータベーステーブルにアクセスできるようにします。
- **挿入の累積行数** を選択すると、`SQLRowCount()` メソッドへの呼び出しが実行されたときに `INSERT` ステートメントのパラメーター設定の配列全体に対して単一の累積行数が返されます。このオプションがクリアされている場合、配列内のパラメーター設定ごとに個々の行カウントを使用できるため、`SQLRowCount()` を呼び出すと、最後に挿入された行のカウントが返されます。
- **LF<->CR/LF** 変換を選択して、サーバーから文字値をフェッチするときにラインフィード文字をキャリッジリターン/ラインフィードのペアに変換し、文字値をサーバーに送信します。デフォルトでは、このオプションはオンです。
- **UpdatableCursors** を選択して、ドライバーが `SQLSetPos()` または `SQLBulkOperations()` 関数での位置付け `UPDATE` および `DELETE` 操作を許可するように指定します。デフォルトでは、このオプションはオンです。
- **byteasLO** を選択して、ドライバーが `BYTEA` 値を `SQL_LONGVARBINARY` 値であるかのように扱うように指定します。オフにすると、EDB-ODBC は `BYTEA` 値を `SQL_VARBINARY` 値であるかのように扱います。デフォルトでは、`BYTEA` 値は `SQL_VARBINARY` 値として扱われます。
- **RowVersioning** を選択して、テーブル内の列をレポートするときに `xmin` 列を含めます。`xmin` 列は、行を作成したトランザクションの ID です。`SQL_CONCURRENCY = SQL_CONCUR_ROWVER.1` でカーソルを作成する予定がある場合は、行バージョン管理を使用する必要があります。デフォルトでは、このオプションはオフです。
- **DisallowPremature** を選択して、実際にクエリを実行せずにクエリに関するメタデータ（結果セット内の列数または列タイプ）をドライバーが取得するように指定します。このオプションがオフの場合、クエリに関するメタデータを要求すると、ドライバーはクエリを実行します。デフォルトでは、このオプションはオフです。
- **Trueis-1** を選択して、ドライバーに `True` の `BOOL` 値を -1 として返すように指示します。このオプションがクリアされている場合、ドライバーは `True` の `BOOL` 値を 1 として返します。ドライバーは常に `False` の `BOOL` 値を 0 として返します。
- **Serversideprepare** を選択して、`PREPARE` および `EXECUTE` コマンドを使用して `Prepare/Execute` モデルを実装するようにドライバーに指示します。デフォルトでは、このオプションはオンです。
- **GSS 要求に gssapi を使用する** を選択して、サーバーに GSSAPI 接続要求を送信するようにドライバーに指示します。
- **dbms_name** フィールドにデータベースシステム（EnterpriseDB または PostgreSQL のいずれか）を入力します。ここで入力した値は、`SQLGetInfo()` 関数が呼び出されたときに `SQL_DBMS_NAME` 引数に返されます。デフォルトは EnterpriseDB です。

Int8As ボックスのオプションを使用して、ドライバーが `BIGINT` 値をクライアントに返す方法を指定します。クライアントが MS Jet の場合、**default** を選択して、**NUMERIC** のデフォルトタイプを指定します。クライアントが他の ODBC クライアントの場合は、**BIGINT** を選択します。オプションで、ドライバーが `BIGINT` 値を `bigint` (`SQL_BIGINT`)、`numeric` (`SQL_NUMERIC`)、`varchar` (`SQL_VARCHAR`)、`double` (`SQL_DOUBLE`)、または `int4` (`SQL_INTEGER`) として返すように指定できます。

ExtraOpts フィールドのデフォルト値は `0x0` です。**ExtraOpts** の場合、表に示されているオプションを指定できます。

Option	Specifies
0x1	短い長さの形式の接続文字列の出力を強制します。MFC CDatabase クラスを使用している場合は、このオプションを選択します。
0x2	MS Access が PostgreSQL のシリアルタイプを AutoNumber タイプとして認識できるようにします。
0x4	アプリケーションからの問い合わせに対して ANSI 文字タイプを返します。Unicode データの処理が難しいアプリケーションの場合は、このオプションを選択します。
0x8	設定されている場合、NULL の日付は空の文字列として報告され、空の文字列は入力時に NULL の日付として解釈されます。
0x10	`SQLGetInfo` がすべてのテーブルに関する情報を返すか、アクセス可能なテーブルのみを返すかを決定します。設定されている場合、アクセス可能なテーブルの情報のみが返されます。
0x20	設定されている場合、各 SQL コマンドは個別のネットワークラウンドトリップで処理されます。それ以外の場合、SQL コマンドは可能な限り少ないラウンドトリップにグループ化され、ネットワークの待機時間を短縮します。

プロトコルボックスには、特定のフロントエンド/バックエンドプロトコルバージョンを使用してサーバーと対話するようにドライバーに指示するオプションが含まれています。デフォルトでは、選択されたプロトコルは **7.4+** です。オプションで、バージョン **6.4+**、**6.3**、または **6.2** から選択できます。

LevelofRollbackonErrors ボックスには、ドライバーがエラー処理を処理する方法を指定するオプションが含まれています。

Option	Specifies
Transaction	ドライバーでエラーが発生した場合、現在のトランザクションをロールバックします。
Statement	ドライバーでエラーが発生した場合、現在のステートメントをロールバックします。
Nop	ドライバーでエラーが発生した場合、アプリケーションを続行するには、現在のトランザクションを手動でロールバックする必要があります。

OIDOptions ボックスには、ドライバーがいくつかのテーブルに含まれる OID 列を公開する方法を制御するオプションが含まれています。

- **ShowColumn** を選択して、SQLColumns() ファンクションの結果セットに OID 列を含めます。クリアすると、OID 列は SQLColumns() から非表示になります。
- **FakeColumns** を選択して、SQLStatistics() ファンクションに指定し、各 OID 列に一意のインデックスが存在することを報告します。

ConnectSettings フィールドを使用して、この接続を開くときに使用するドライバーのパラメーター割り当てのリストを指定します。SET ステートメントで変更できる構成パラメーターは、セミコロン区切りのリストに含めることができます。例：

```
set search_path to company1,public;
```

名前付きデータソースの接続プロパティを定義した後、**適用**を選択してオプションを適用します。**OK**を選択してオプションを保存し、終了します。

(EnterpriseDB ODBC Driver ダイアログボックスで) **Global** を選択して、Global Settings ダイアログボックスを開きます。このダイアログボックスのオプションは、EDB-ODBC ドライバーのログオプションを制御します。このダイアログボックスを使用して、名前付きデータソースなしでドライバーが使用される場合、または接続文字列が解析される前に発生するドライバー操作のログを記録する場合にログを強制します。

- **CommLog** を選択して、すべてのクライアント/サーバートラフィックをログファイルに記録します。ログファイルの名前は `C:\psqlodbc_<process-id>` です。<process-id> は、使用中のプロセスの名前です。
- **Mylog** を選択して、ドライバーのアクティビティのログファイルを保持します。ログファイルの名前は `c:\mylog_<process-id>` です。<process-id> は、使用中のプロセスの名前です。
- **ログファイルのフォルダー** フィールドにログファイルの場所を指定します。名前付きデータソースの接続情報を入力した後、**テスト** を選択して、ドライバーマネージャーが定義されたデータソースに接続できることを確認します。

OK を選択して、[接続テスト] ダイアログボックスを終了します。接続が成功したら、**Save** を選択して名前付けたデータソースを保存します。接続の確立に問題がある場合は、パラメーターを調整して再度テストします。

5.2 Linux でデータソース定義を追加する

Linux で名前付きデータソースをテキストファイルで定義して、ドライバーマネージャーが読み取ってデータベースに接続する方法を決定できます。ドライバーマネージャーは通常、名前付きデータソースを2つの場所で検索します。

- `/user/.odbc.ini` —現在のユーザーのみが使用できる名前付きデータソース。
- `/etc/odbc.ini` —すべてのユーザーが使用できる名前付きデータソース。

これらのファイルの構造に違いはなく、両方のファイルで同じ接続プロパティを使用できます。2つのファイルの場所のみが異なります。システムで両方のファイルを利用できる場合、ユーザーデータソース `/user/.odbc.ini` はシステムデータソース `/etc/odbc.ini` をオーバーライドします。

データソース定義には、角かっこで囲まれたデータソース名と、ドライバーと接続のプロパティのリストが、プロパティ名の後に等号 (=) が続き、プロパティ値が続く形式で含まれます。

EDB-ODBC ドライバーの一般的なユーザーまたはシステムデータソース定義：

```
[EnterpriseDB]
Description = EnterpriseDB DSN
Driver = /usr/edb/odbc/lib/edb-odbc.so
Trace = yes
TraceFile = /tmp/odbc.log
Database = edb
Servername = localhost
```

(次のページに続く)

```

UserName = enterprisedb
Password = manager
Port = 5444

```

可能なプロパティは次のとおりです。

Property	Description
[EnterpriseDB]	データソースの名前。任意の名前を使用できます。
Description	名前付きデータソースの説明。
Driver	EDB-ODBC ドライバーライブラリファイル (edb-odbc.so) のパス。
Trace	`Yes`は、ODBC アプリケーションからログファイルに行われた呼び出しのシーケンスを記録する unixODBC ドライバーのトレースユーティリティをオンにします。trace ユーティリティを使用すると、アプリケーションの速度が低下する場合があります。
TraceFile	トレースユーティリティが返す情報を受け取るファイル。
Database	EDB Postgres Advanced Server データベース。
Servename	EDB Postgres Advanced Server のホスト名または IP アドレス。
Username	ユーザーの名前。
Password	ユーザーのパスワード。
Port	ポート番号。
Protocol	フロントエンド/バックエンドのプロトコルバージョン。デフォルト値は 7.4 です。オプションで、プロトコルバージョン 7.4、6.4、6.3、または 6.2 を指定できます。
ReadOnly	<i>Yes</i> は、ドライバが次のコマンドを実行できないようにします。INSERT、UPDATE、DELETE、CREATE、ALTER、DROP、GRANT、REVOKE、または LOCK。また、ODBC プロシージャコールエスケープ構文 (<i>call=procedure-name?</i>) を使用する呼び出しを防ぎます。デフォルト値は <i>No</i> です。
RowVersioning	<i>Yes</i> は、テーブル内の列をレポートするときに <i>xmin</i> 列を含みます。 <i>xmin</i> 列は、行を作成したトランザクションの ID です。 <i>SQL_CONCURRENCY = SQL_CONCUR_ROWVER.1</i> でカーソルを作成する予定がある場合は、行バージョン管理を使用する必要があります。デフォルト値は <i>No</i> です。
ShowSystemTables	`Yes`は、`SQLTables()`ファンクションの結果セットにシステムテーブルを含めます。デフォルト値は <i>No</i> です。
ShowOidColumn	`Yes`は、`SQLColumns()`ファンクションの結果セットに <i>OID</i> 列を含めます。`ShowOidColumn`が <i>No</i> に設定されている場合、 <i>OID</i> 列は <i>SQLColumns()</i> から隠されます。デフォルト値は <i>No</i> です。
FakeOidIndex	`Yes`は、`SQLStatistics()`ファンクションを指定して、各 <i>OID</i> 列に一意のインデックスが存在することを報告します。これは、アプリケーションに一意の識別子が必要で、テーブルに含まれていない場合に役立ちます。デフォルト値は <i>No</i> です。
ConnSettings	この接続を開くときに使用するドライバのパラメーター割り当てのリスト。

第6章 Driver functionality

ODBC 関数を使用して、EDB-ODBC とサーバー間の接続のさまざまな属性に関する特定の情報について ODBC にクエリを実行できます。

- `SQLGetInfo()` は、EDB-ODBC ドライバーと EDB Postgres Advanced Server に関する情報を返します。
- `SQLGetEnvAttr()` は、ODBC 環境属性に関する情報を返します。
- `SQLGetConnectAttr()` は、個々の接続に固有の属性に関する情報を返します。
- `SQLGetStmtAttr()` は、個々のステートメントに固有の属性に関する情報を返します。

ODBC 関数を使用して、ODBC とのインターフェイスに使用するオブジェクトの属性を設定することもできます。

- `SQLSetConnectAttr()` 関数を使用して、接続属性を設定します。
- `SQLSetEnvAttr()` 関数を使用して、環境属性を設定します。
- `SQLSetStmtAttr()` 関数を使用して、ステートメントの属性を設定します。

6.1 SQLGetInfo()

ODBC `SQLGetInfo()` 関数は、EDB-ODBC ドライバーと EDB Postgres Advanced Server に関する情報を返します。`SQL_ODBC_VER` を `info_type` として指定しない限り、`SQLGetInfo()` を呼び出すには、開いている接続が必要です。`SQLGetInfo()` の署名は次のとおりです。

```
SQLRETURN SQLGetInfo
(
    SQLHDBC conn_handle , // Input
    SQLUSMALLINT info_type , // Input
    SQLPOINTER info_pointer , // Output
    SQLSMALLINT buffer_len , // Input
    SQLSMALLINT * string_length_pointer // Output
);
```

- `conn_handle` — 接続ハンドル。
- `info_type` — `SQLGetInfo()` が取得している情報のタイプ。
- `info_pointer` — 取得した値を保持するメモリバッファへのポインター。

`info_type` 引数が `SQL_DRIVER_HDESC` または `SQL_DRIVER_HSTMT` の場合、`info_pointer` 引数は `Input` と `Output` の両方です。

- **buffer_len** — **info_pointer** が指す割り当てられたメモリバッファの長さ。 **info_pointer** が NULL の場合、 **buffer_len** は無視されます。戻り値が固定サイズの場合、 **buffer_len** は無視されます。 **buffer_len** は、要求された値が文字列の形式で返される場合にのみ使用されます。
- **string_length_pointer** — **SQLSMALLINT** 値へのポインター。 **SQLGetInfo()** は、要求された値のサイズをこの整数に書き込みます。

典型的な使用法は、NULL **info_pointer** で **SQLGetInfo()** を呼び出して、要求された値の長さを取得し、必要なバイト数を割り当ててから、**SQLGetInfo()** を再度呼び出し（新しく割り当てられたバッファのアドレスを提供して）実際の値を取得します。最初の呼び出しは、値を保持するために必要なバイト数を取得します。2 番目の呼び出しは、値を取得します。

戻り値のサイズが **buffer_len** を超える場合、情報は切り捨てられ、NULL は終了します。戻り値が固定サイズの場合、 **string_length** は無視され、要求された値のサイズは **SQLGetInfo()** によって提供されません。

SQLGetInfo() は、次のいずれかの形式で情報を書き込みます。

- **SQLINTEGER** ビットマスク
- **SQLINTEGER** フラグ
- **SQLINTEGER** バイナリ値
- **SQLSMALLINT** 値
- NULL 終了文字列

SQLGetInfo() は、**SQL_SUCCESS**、**SQL_SUCCESS_WITH_INFO**、**SQL_ERROR**、または **SQL_INVALID_HANDLE** を返します。

次の表は、EDB Postgres Advanced Server 接続に関して EDB-ODBC によって返される情報をリストしています。

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_ACCESSIBLE_PROCEDURES : Indicates if procedures returned by <i>SQLProcedures()</i> can be executed by the application.	Returns <i>N</i> . Some procedures executed by the <i>SQLProcedures()</i> function might be executed by the application.
SQL_ACCESSIBLE_TABLES : Indicates if the user has SELECT privileges on all table names returned by <i>SQLTables()</i> .	Returns <i>N</i> . The user might not have select privileges on one or more tables returned by the <i>SQLTables()</i> function.
SQL_ACTIVE_CONNECTIONS prev. SQL_MAX_DRIVER_CONNECTIONS : Indicates the maximum number of connections EDB-ODBC can support.	Returns 0. There's no specified limit to the number of connections allowed.
SQL_ACTIVE_ENVIRONMENTS : The number of active environments EDB-ODBC can support.	Returns 0. There's no specified limit to the number of environments allowed.
SQL_ACTIVE_STATEMENTS prev. SQL_MAX_CONCURRENT_ACTIVITIES : Indicates the maximum number of active statements EDB-ODBC can support.	Returns 0. There's no specified limit to the number of active statements allowed.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_AGGREGATE_FUNCTION: Identifies the aggregate functions supported by the server and driver.	Returns <i>SQL_AF_ALL</i> .
SQL_ALTER_DOMAIN: Identifies the <i>ALTER DOMAIN</i> clauses supported by the server.	Returns 0. <i>ALTER DOMAIN</i> clauses aren't supported.
SQL_ALTER_TABLE: Identifies the <i>ALTER TABLE</i> clauses supported by the server.	Returns <i>SQL_AT_ADD_COLUMN</i> , <i>SQL_AT_DROP_TABLE_CONSTRAINT_CASCADE</i> , <i>SQL_AT_DROP_TABLE_CONSTRAINT</i> , <i>SQL_AT_CONSTRAINT_INITIALLY_DEFERRED</i> , <i>SQL_AT_CONSTRAINT_INITIALLY_IMMEDIATE</i> , <i>SQL_AT_CONSTRAINT_DEFERRABLE</i>
SQL_ASYNC_MODE: Level of asynchronous mode supported by EDB-ODBC.	Returns <i>SQL_AM_NONE</i> . Asynchronous mode isn't supported.
SQL_BATCH_ROW_COUNT: Indicates how the driver returns row counts.	Returns <i>SQL_BRC_EXPLICIT</i> . Row counts are available when executed by calling <i>SQLExecute</i> or <i>SQLExecDirect</i> .
SQL_BATCH_SUPPORT: Indicates support for batch statement execution.	Returns <i>SQL_BS_SELECT_EXPLICIT</i> , <i>SQL_BS_ROW_COUNT_EXPLICIT</i> . The driver supports explicit batches with result set and row count generating statements.
SQL_BOOKMARK_PERSISTENCE: Indicates level of support for bookmarks.	Returns <i>SQL_BP_DELETE</i> , <i>SQL_BP_TRANSACTION</i> , <i>SQL_BP_UPDATE</i> , <i>SQL_BP_SCROLL</i> .
SQL_CATALOG_LOCATION	Now
SQL_QUALIFIER_LOCATION: Indicates the position of the catalog in a qualified table name.	Returns <i>SQL_CL_START</i> . The catalog portion of a qualified table name is at the beginning of the name.
SQL_CATALOG_NAME	Now
SQL_QUALIFIER_NAME: Indicates support for catalog names.	Returns Y. The server supports catalog names.
SQL_CATALOG_NAME_SEPARATOR	Now
SQL_QUALIFIER_NAME_SEPARATOR: Character separating the catalog name from the adjacent name element.	Returns '.' The server expects a '.' character between the qualifier and the table name.
SQL_CATALOG_TERM	Now
SQL_QUALIFIER_TERM: The term used to describe a catalog.	Returns catalog.
SQL_CATALOG_USAGE	Now
SQL_QUALIFIER_USAGE: Indicates the SQL statements that can refer to catalogs.	Returns <i>SQL_CU_DML_STATEMENTS</i> . Catalog names can be used in <i>SELECT</i> , <i>INSERT</i> , <i>UPDATE</i> , <i>DELETE</i> , <i>SELECT FOR UPDATE</i> and positioned <i>UPDATE</i> and <i>DELETE</i> statements.
SQL_COLLATION_SEQ: Returns the name of the collation sequence.	Returns an empty string. The name of the default collation is unknown.
SQL_COLUMN_ALIAS: Indicates server support for column aliases.	Returns Y. The server supports column aliases.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_CONCAT_NULL_BEHAVIOR: Indicates how the server handles concatenation of NULL values.	Returns <i>SQL_CB_NON_NULL</i> . Concatenating a NULL value and a non-NULL value results in a NULL value.
SQL_CONVERT_BIGINT: Indicates conversion support from the <i>BIGINT</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_BINARY: Indicates conversion support from the <i>BINARY</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_BIT: Indicates conversion support from the <i>BIT</i> type using the <i>CONVERT</i> function.	Returns <i>SQL_CVT_INTEGER</i> , <i>SQL_CVT_BIT</i> .
SQL_CONVERT_CHAR: Indicates conversion support from the <i>CHAR</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_DATE: Indicates conversion support from the <i>DATE</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_DECIMAL: Indicates conversion support from the <i>DECIMAL</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_DOUBLE: Indicates conversion support from the <i>DOUBLE</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_FLOAT: Indicates conversion support from the <i>FLOAT</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_FUNCTIONS: Lists the scalar conversion functions supported by the server and driver using the <i>CONVERT</i> function.	Returns <i>SQL_FN_CVT_CONVERT</i> .
SQL_CONVERT_INTEGER: Lists the conversion support from the <i>INTEGER</i> type using the <i>CONVERT</i> function.	Returns <i>SQL_CVT_INTEGER</i> , <i>SQL_CVT_BIT</i> .
SQL_CONVERT_INTERVAL_DAY_TIME: Indicates conversion support from the <i>INTERVAL_DAY_TIME</i> type using the <i>CONVERT</i> function.	This info_type isn't currently supported.
SQL_CONVERT_INTERVAL_YEAR_MONTH: Indicates conversion support from the <i>INTERVAL_YEAR_MONTH</i> type using the <i>CONVERT</i> function.	This info_type isn't currently supported.
SQL_CONVERT_LONGVARBINARY: Indicates conversion support for the <i>LONG_VARBINARY</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_LONGVARCHAR: Indicates conversion support for the <i>LONGVARCHAR</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_NUMERIC: Indicates conversion support for the <i>NUMERIC</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_REAL: Indicates conversion support for the <i>REAL</i> type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_SMALLINT: Indicates conversion support for the <i>SMALLINT</i> type using the <i>CONVERT</i> function.	Returns <i>SQL_CVT_INTEGER</i> , <i>SQL_CVT_BIT</i> .

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_CONVERT_TIME: Indicates conversion support for TIME type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CVT_TIMESTAMP: Indicates conversion support for TIMESTAMP type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_TINYINT: Indicates conversion support for the TINYINT type using the <i>CONVERT</i> function.	Returns <i>SQL_CVT_INTEGER</i> , <i>SQL_CVT_BIT</i> .
SQL_CONVERT_VARBINARY: Indicates conversion support for the VARBINARY type using the <i>CONVERT</i> function.	Returns 0. The server doesn't support conversion.
SQL_CONVERT_VARCHAR: Indicates conversion support for VARCHAR type using the <i>CONVERT</i> function.	Returns <i>SQL_CVT_INTEGER</i> , <i>SQL_CVT_BIT</i> .
SQL_CONVERT_WCHAR: Indicates conversion support for the WCHAR type using the <i>CONVERT</i> function.	This info_type is valid only when using the Unicode driver. Returns 0. The server doesn't support conversion.
SQL_CONVERT_WLONGVARCHAR: Indicates conversion support for the WLONGVARCHAR type using the <i>CONVERT</i> function.	This info_type is valid only when using the Unicode driver. Returns 0. The server doesn't support conversion.
SQL_CONVERT_WVARCHAR: Indicates conversion support for the WVARCHAR type using the <i>CONVERT</i> function.	This info_type is valid only when using the Unicode driver. Returns 0. The server doesn't support conversion.
SQL_CORRELATION_NAME: Indicates server support for correlation names.	Returns <i>SQL_CN_ANY</i> . Correlation names are supported and can be any valid name.
SQL_CREATE_ASSERTION: Indicates support for the <i>CREATE ASSERTION</i> statement.	Returns 0. The <i>CREATE ASSERTION</i> statement isn't supported.
SQL_CREATE_CHARACTER_SET: Indicates support for the <i>CREATE CHARACTER</i> statement.	Returns 0. The <i>CREATE CHARACTER</i> statement isn't supported.
SQL_CREATE_COLLATION: Indicates support for the <i>CREATE COLLATION</i> statement.	Returns 0. The <i>CREATE COLLATION</i> statement isn't supported.
SQL_CREATE_DOMAIN: Indicates support for the <i>CREATE DOMAIN</i> statement.	Returns 0. The <i>CREATE DOMAIN</i> statement isn't supported.
SQL_CREATE_SCHEMA: Indicates support for the <i>CREATE SCHEMA</i> statement.	Returns <i>SQL_CS_CREATE_SCHEMA</i> , <i>SQL_CS_AUTHORIZATION</i> .
SQL_CREATE_TABLE: Indicates support for the <i>CREATE TABLE</i> statement.	Returns <i>SQL_CT_CREATE_TABLE</i> , <i>SQL_CT_GLOBAL_TEMPORARY</i> , <i>SQL_CT_CONSTRAINT_INITIALLY_DEFERRED</i> , <i>SQL_CT_CONSTRAINT_INITIALLY_IMMEDIATE</i> , <i>SQL_CT_CONSTRAINT_DEFERRABLE</i> , <i>SQL_CT_COLUMN_CONSTRAINT</i> , <i>SQL_CT_COLUMN_DEFAULT</i> , <i>SQL_CT_TABLE_CONSTRAINT</i> , <i>SQL_CT_CONSTRAINT_NAME_DEFINITION</i>

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_CREATE_TRANSLATION: Indicates support for the <i>CREATE TRANSLATION</i> statement.	Returns 0. The <i>CREATE TRANSLATION</i> statement isn't supported.
SQL_CREATE_VIEW: Indicates support for the <i>CREATE VIEW</i> statement.	Returns <i>SQL_CV_CREATE_VIEW</i> .
SQL_CURSOR_COMMIT_BEHAVIOR: Indicates how a <i>COMMIT</i> operation affects the cursor.	Returns <i>SQL_CB_PRESERVE</i> . Cursors are unchanged and can continue to fetch data.
SQL_CURSOR_ROLLBACK_BEHAVIOR: Indicates the server behavior after a <i>ROLLBACK</i> operation.	Returns <i>SQL_CB_PRESERVE</i> . Cursors are unchanged and can continue to fetch data.
SQL_CURSOR_SENSITIVITY: Indicates how the server synchronizes changes to a result set.	This info_type isn't currently supported.
SQL_DATA_SOURCE_NAME: Returns the server name used during connection.	The value returned is determined by the connection properties.
SQL_DATA_SOURCE_READ_ONLY: Indicates if the connection is in READ ONLY mode.	The value returned is determined by the connection properties.
SQL_DATABASE_NAME: Returns the name of the database.	The value returned is determined by the connection properties.
SQL_DATETIME_LITERALS: Indicates the DATETIME LITERALS supported by the server.	This info_type is not supported.
SQL_DBMS_NAME: Returns the name of the DBMS system.	Returns the value given by the <i>dbms_name</i> parameter from the <i>odbc.ini</i> file on Linux or the <i>dbms_name</i> field of page 2 of the Advanced Options dialog box when defining a data source in Windows. The default is <i>EnterpriseDB</i> .
SQL_DBMS_VER: Returns the server version.	Determined by the server.
SQL_DDL_INDEX: Indicates support for creating and dropping indexes.	Returns <i>SQL_DI_CREATE_INDEX</i> , <i>SQL_DI_DROP_INDEX</i> .
SQL_DEFAULT_TXN_ISOLATION: Indicates support for transaction isolation by the server.	Returns <i>TXN_READ_COMMITTED</i> . Nonrepeatable or phantom reads are possible. Dirty reads aren't.
SQL_DESCRIBE_PARAMETER: Indicates support for the <i>DESCRIBE INPUT</i> statement.	Returns <i>N</i> . The <i>DESCRIBE INPUT</i> statement isn't supported.
SQL_DM_VER: The version of the driver manager.	Determined by driver manager.
SQL_DRIVER_HDBC: The driver's connection handle.	Returns an <i>SQLULEN</i> value that contains the driver's connection handle.
SQL_DRIVER_HDESC: The driver descriptor handle.	Returns an <i>SQLULEN</i> value that contains driver's descriptor handle.
SQL_DRIVER_HENV: The driver's environment handle.	Returns an <i>SQLULEN</i> value that contains the driver's environment handle.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_DRIVER_HLIB: The driver handle.	Returns an <i>SQLULEN</i> value that contains the library handle (returned to the ODBC driver manager when the manager loaded the driver).
SQL_DRIVER_HSTMT: The driver's statement handle.	Returns an <i>SQLULEN</i> value that contains the driver's statement handle.
SQL_DRIVER_NAME: The name of the driver.	Returns <i>EDB-ODBC.DLL</i>
SQL_DRIVER_ODBC_VER: Identifies the ODBC version that the driver supports.	Returns <i>03.50</i>
SQL_DRIVER_VER: Identifies the driver version.	Returns <i>9.0.0.6</i>
SQL_DROP_ASSERTION: Lists the <i>DROP ASSERTION</i> clauses supported by the server.	Returns <i>0</i> .
SQL_DROP_CHARACTER_SET: Lists the <i>DROP CHARACTER</i> clauses supported by the server.	Returns <i>0</i> .
SQL_DROP_COLLATION: Lists the <i>DROP COLLATION</i> clauses supported by the server.	Returns <i>0</i> .
SQL_DROP_DOMAIN: Lists the <i>DROP DOMAIN</i> clauses supported by the server.	Returns <i>0</i> .
SQL_DROP_SCHEMA: Lists the <i>DROP SCHEMA</i> clauses supported by the server.	Returns <i>SQL_DS_DROP_SCHEMA</i> , <i>SQL_DS_RESTRICT</i> , <i>SQL_DS_CASCADE</i> .
SQL_DROP_TABLE: Lists the <i>DROP TABLE</i> clauses supported by the server.	Returns <i>SQL_DT_DROP_TABLE</i> , <i>SQL_DS_RESTRICT</i> , <i>SQL_DS_CASCADE</i> .
SQL_DROP_TRANSLATION: Lists the <i>DROP TRANSLATION</i> clauses supported by the server.	Returns <i>0</i> .
SQL_DROP_VIEW: Lists the <i>DROP VIEW</i> clauses supported by the server.	Returns <i>SQL_DV_DROP_VIEW</i> , <i>SQL_DS_RESTRICT</i> , <i>SQL_DS_CASCADE</i> .
SQL_DYNAMIC_CURSOR_ATTRIBUTES1: Describes the first set of dynamic cursor attributes supported by the driver.	Returns <i>0</i> .
SQL_DYNAMIC_CURSOR_ATTRIBUTES2: Describes the second set of dynamic cursor attributes supported by the driver.	Returns <i>0</i> .
SQL_EXPRESSIONS_IN_ORDERBY: Indicates server support for <i>ORDER BY</i> .	Returns <i>Y</i> .
SQL_FETCH_DIRECTION: Indicates <i>FETCH</i> order options (deprecated in ODBC 3.0).	Returns <i>SQL_FD_FETCH_NEXT</i> , <i>SQL_FD_FETCH_FIRS</i> , <i>SQL_FD_FETCH_LAST</i> , <i>SQL_FD_FETCH_PRIOR</i> , <i>SQL_FD_FETCH_ABSOLUTE</i> , <i>SQL_FD_FETCH_RELATIVE</i> , <i>SQL_FD_FETCH_BOOKMARK</i> .

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_FILE_USAGE: Indicates how a single-tier driver treats files on the server.	Returns <i>SQL_FILE_NOT_SUPPORTED</i> . The driver isn't a single-tier file.
SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1: Describes the forward-only cursor attributes supported by the driver.	Returns <i>SQL_CA1_NEXT</i> .
SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES2: Describes extended attributes for the forward-only cursor designated by <i>SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1</i> .	Returns <i>SQL_CA2_READ_ONLY_CONCURRENCY</i> , <i>SQL_CA2_CRC_EXACT</i> .
SQL_GETDATA_EXTENSIONS: Lists supported extensions to <i>SQLGetData</i> .	Returns <i>SQL_GD_ANY_COLUMN</i> , <i>SQL_GD_ANY_ORDER</i> , <i>SQL_GD_BLOCK</i> , <i>SQL_GD_BOUNDED</i> .
SQL_GROUP_BY: Indicates the relationship between a <i>GROUP BY</i> clause and columns in the <i>SELECT</i> list.	Returns <i>SQL_GB_GROUP_BY_EQUALS_SELECT</i> .
SQL_IDENTIFIER_CASE: Indicates case sensitivity and case storage of SQL identifiers.	Returns <i>SQL_IC_LOWER</i> .
SQL_INDEX_KEYWORDS: Indicates support for the <i>CREATE INDEX</i> statement.	Returns <i>SQL_IK_NONE</i> .
SQL_INFO_SCHEMA_VIEWS: Lists the views supported in the <i>INFORMATION_SCHEMA</i> .	Returns 0.
SQL_INTEGRITY Prev. SQL_ODBC_SQL_OPT_IEF: Indicates server support for referential integrity syntax checking.	Returns N.
SQL_INSERT_STATEMENT: Indicates level of support for the <i>INSERT</i> statement.	Returns <i>SQL_IS_INSERT_LITERALS</i> , <i>SQL_IS_INSERT_SEARCHED</i> , <i>SQL_IS_SELECT INTO</i> .
SQL_KEYSET_CURSOR_ATTRIBUTES1: Describes the first set of keyset cursor attributes supported by the driver.	Returns <i>SQL_CA1_NEXT</i> , <i>SQL_CA1_ABSOLUTE</i> , <i>SQL_CA1_RELATIVE</i> , <i>SQL_CA1_BOOKMARK</i> , <i>SQL_CA1_LOCK_NO_CHANGE</i> , <i>SQL_CA1_POS_POSITION</i> , <i>SQL_CA1_POS_UPDATE</i> , <i>SQL_CA1_POS_DELETE</i> , <i>SQL_CA1_POS_REFRESH</i> , <i>SQL_CA1_BULK_ADD</i> , <i>SQL_CA1_BULK_UPDATE_BY_BOOKMARK</i> , <i>SQL_CA1_BULK_DELETE_BY_BOOKMARK</i> , <i>SQL_CA1_BULK_FETCH_BY_BOOKMARK</i> .

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_KEYSET_CURSOR_ATTRIBUTES2: Describes the second set of keyset cursor attributes supported by the driver.	Returns <i>SQL_CA2_READ_ONLY_CONCURRENCY</i> , <i>SQL_CA2_OPT_ROWVER_CONCURRENCY</i> , <i>SQL_CA2_SENSITIVITY_ADDITIONS</i> , <i>SQL_CA2_SENSITIVITY_DELETIONS</i> , <i>SQL_CA2_SENSITIVITY_UPDATES</i> , <i>SQL_CA2_CRC_EXACT</i> .
SQL_KEYWORDS: Identifies the server-specific reserved keywords.	Returns “”. There are no server-specific reserved keywords.
SQL_LIKE_ESCAPE_CLAUSE: Indicates support for an escape character in <i>LIKE</i> predicates.	Returns <i>N</i> . EDB Postgres Advanced Server doesn't support escape characters in <i>LIKE</i> predicates.
SQL_LOCK_TYPES: Lists supported lock types (deprecated in ODBC 3.0).	Returns <i>SQL_LCK_NO_CHANGE</i> .
SQL_MAX_ASYNC_CONCURRENT_STATEMENTS: The number of active concurrent statements that the driver can support.	This info_type is currently unsupported.
SQL_MAX_BINARY_LITERAL_LEN: The maximum length of a binary literal.	Returns 0. The maximum length is unspecified.
SQL_MAX_CATALOG_NAME_LEN: The maximum length of a catalog name on the server.	Returns 0. The maximum length is unspecified.
SQL_MAX_QUALIFIER_NAME_LEN: The maximum length of a qualifier.	Returns 0. The maximum length is unspecified.
SQL_MAX_CHAR_LITERAL_LEN: The maximum number of characters in a character string.	Returns 0. The maximum length is unspecified.
SQL_MAX_COLUMN_NAME_LEN: The maximum length of a column name.	Returns 64. Column names can't exceed 64 characters in length.
SQL_MAX_COLUMNS_IN_GROUP_BY: The maximum number of columns allowed in a <i>GROUP BY</i> clause.	Returns 0. The maximum length is unspecified.
SQL_MAX_COLUMNS_IN_INDEX: The maximum number of columns allowed in an index.	Returns 0. The maximum length is unspecified.
SQL_MAX_COLUMNS_IN_ORDER_BY: The maximum number of columns allowed in an <i>ORDER BY</i> clause.	Returns 0. The maximum length is unspecified.
SQL_MAX_COLUMNS_IN_SELECT: The maximum number of columns allowed in a <i>SELECT</i> list.	Returns 0. The maximum length is unspecified.
SQL_MAX_COLUMNS_IN_TABLE: The maximum number of columns allowed in a table.	Returns 0. The maximum length is unspecified.
SQL_MAX_CONCURRENT_ACTIVITIES prev. SQL_MAX_ACTIVE_STATEMENTS: The maximum number of active SQL statements that the driver can support.	Returns 0. The maximum length is unspecified.
SQL_MAX_CURSOR_NAME_LEN: The maximum length of a cursor name.	Returns 32. A cursor name can't exceed 32 characters in length.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**		EDB_ODBC/EDB Postgres Advanced Server returns
SQL_MAX_DRIVER_CONNECTIONS	prev.	Returns 0. There is no specified limit to the number of connections supported.
SQL_ACTIVE_CONNECTIONS: The maximum number of active connections the driver can support.		
SQL_MAX_IDENTIFIER_LEN: The maximum identifier length allowed by the server.		Returns 64. Identifiers can't exceed 64 characters in length.
SQL_MAX_INDEX_SIZE: The maximum number of bytes allowed in the (combined) fields of an index.		Returns 0. The maximum size is unspecified.
SQL_MAX_OWNER_NAME_LEN	Now	Returns 64. The maximum length of an owner name is 64 characters.
SQL_MAX_SCHEMA_NAME_LEN: The maximum length of an owner name allowed by the server.		
SQL_MAX_PROCEDURE_NAME_LEN: The maximum length of a procedure name allowed by the server.		Returns 0. The maximum length is unspecified.
SQL_MAX_QUALIFIER_NAME_LEN	Now	Returns 0. The maximum length of a qualifier is unspecified.
SQL_MAX_CATALOG_NAME_LEN: The maximum length of a qualifier name allowed by the server.		
SQL_MAX_ROW_SIZE: The maximum length of a row.		Returns 0. The maximum row length is unspecified.
SQL_MAX_ROW_SIZE_INCLUDES_LONG: Indicates whether the <i>SQL_MAX_ROW_SIZE</i> includes the length of any <i>LONGVARCHAR</i> or <i>LONGVARBINARY</i> columns in the row.		Returns <i>Y</i> . <i>SQL_MAX_ROW_SIZE</i> includes the length of any <i>LONGVARCHAR</i> or <i>LONGVARBINARY</i> columns in the row.
SQL_MAX_SCHEMA_NAME_LEN: The maximum length of a schema name allowed by the server.		Returns 64. The maximum length of a schema name is 64 characters.
SQL_MAX_STATEMENT_LEN: The maximum length of a SQL statement.		Returns 0. Maximum statement length is limited by available memory.
SQL_MAX_TABLE_NAME_LEN: The maximum length of a table name allowed by the server.		Returns 64. The maximum length of a table name is 64 characters.
SQL_MAX_TABLES_IN_SELECT: The maximum number of tables allowed in the <i>FROM</i> clause of a <i>SELECT</i> statement.		Returns 0. The maximum number of tables allowed is unspecified.
SQL_MAX_USER_NAME_LEN: The maximum length of the user name allowed by the server.		Returns 0. The maximum length of a user name is unspecified.
SQL_MULT_RESULT_SETS: Indicates server support for multiple result sets.		Returns <i>Y</i> . EDB Postgres Advanced Server supports multiple result sets.
SQL_MULTIPLE_ACTIVE_TXN: Indicates if the server supports multiple active transactions.		Returns <i>Y</i> . EDB Postgres Advanced Server supports multiple active transactions.
SQL_NEED_LONG_DATA_LEN: Indicates if the server needs the length of a <i>LONG</i> data value before receiving the value.		Returns <i>N</i> . EDB Postgres Advanced Server doesn't need the length of a <i>LONG</i> data value before receiving the value.
SQL_NON_NULLABLE_COLUMNS: Indicates if the server supports NOT NULL values in columns.		Returns <i>SQL_NNC_NON_NULL</i> . EDB Postgres Advanced Server does support NOT NULL values in columns.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_NULL_COLLATION: Indicates where NULL values are located in a result set.	Returns <i>SQL_NC_HIGH</i> . The location of NULL values in a data set is determined by the <i>ASC</i> and <i>DESC</i> keywords. NULL values are sorted to the high end of the data set.
SQL_NUMERIC_FUNCTIONS: Lists the numeric functions supported by the driver and the server.	Returns <i>SQL_FN_NUM_ABS</i> , <i>SQL_FN_NUM_ATAN</i> , <i>SQL_FN_NUM_CEILING</i> , <i>SQL_FN_NUM_COS</i> , <i>SQL_FN_NUM_EXP</i> , <i>SQL_FN_NUM_FLOOR</i> , <i>SQL_FN_NUM_LOG</i> , <i>SQL_FN_NUM_MOD</i> , <i>SQL_FN_NUM_SIGN</i> , <i>SQL_FN_NUM_SIN</i> , <i>SQL_FN_NUM_SQRT</i> , <i>SQL_FN_NUM_TAN</i> , <i>SQL_FN_NUM_RAND</i> , <i>SQL_FN_NUM_POWER</i> , <i>SQL_FN_NUM_ROUND</i> .
SQL_ODBC_API_CONFORMANCE: Indicates the ODBC 3.0 compliance level.	Returns <i>SQL_OAC_LEVEL1</i> . The driver conforms to ODBC Level 1 interface.
SQL_ODBC_INTERFACE_CONFORMANCE: Indicates the ODBC interface that the driver adheres to.	Returns <i>SQL_OIC_CORE</i> .
SQL_ODBC_SAG_CLI_CONFORMANCE: Indicates the SQL Access Group compliance level that the driver adheres to.	Returns <i>SQL_OSCC_NOT_COMPLIANT</i> . The driver isn't SAG CLI compliant.
SQL_ODBC_SQL_CONFORMANCE: Indicates the SQL grammar level that the driver conforms to.	Returns <i>SQL_OSC_CORE</i> . The driver conforms to the core grammar level.
SQL_ODBC_SQL_OPT_IEF Now SQL_INTEGRITY: Indicates server support for referential integrity syntax checking.	Returns <i>N</i> . The server doesn't support referential integrity syntax checking.
SQL_ODBC_VER: The ODBC version supported by the driver manager	Returns <i>03.52.0000</i> .
SQL_OJ_CAPABILITIES: Identifies the outer joins that are supported by the server.	Returns <i>SQL_OJ_LEFT</i> , <i>SQL_OJ_RIGHT</i> , <i>SQL_OJ_FULL</i> , <i>SQL_OJ_NESTED</i> , <i>SQL_OJ_NOT_ORDERED</i> , <i>SQL_OJ_INNER</i> , <i>SQL_OJ_ALL_COMPARISON_OPS</i> .
SQL_OUTER_JOINS: Indicates support for outer joins and the outer join escape sequence.	Returns <i>Y</i> . Outer joins are supported.
SQL_OWNER_TERM prev. SQL_SCHEMA_TERM: The term used to describe a schema.	Returns schema.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_ORDER_BY_COLUMNS_IN_SELECT: Indicates if the columns in an <i>ORDER BY</i> clause must be included in the <i>SELECT</i> list.	Returns <i>N</i> . Columns in an <i>ORDER BY</i> clause don't have to be in the <i>SELECT</i> list.
SQL_OWNER_USAGE prev. SQL_SCHEMA_USAGE: Returns a string that indicates which statements support schema qualifiers.	Returns <i>SQL_OU_DML_STATEMENTS</i> , <i>SQL_OU_TABLE_DEFINITION</i> , <i>SQL_OU_INDEX_DEFINITION</i> , <i>SQL_OU_PRIVILEGE_DEFINITION</i> .
SQL_PARAM_ARRAY_ROW_COUNTS: Indicates if the server returns a single row count or separate row counts for each element in an array when executing a parameterized statement with at least one parameter bound to the array.	Returns <i>SQL_PARC_BATCH</i> if separate row counts are available for each element in an array. Returns <i>SQL_PARC_NO_BATCH</i> if a single, cumulative row count is available for the entire array.
SQL_PARAM_ARRAY_SELECTS: Indicates if the server returns one result set or a separate result set for each element in an array (or if the driver doesn't allow this feature) when executing a parameterized statement with at least one parameter bound to the array.	Returns <i>SQL_PAS_BATCH</i> . One data set is available for each element in an array.
SQL_POS_OPERATION: Lists the options supported by <i>SQLSetPos()</i> .	Returns <i>SQL_POS_POSITION</i> , <i>SQL_POS_REFRESH</i> , <i>SQL_POS_UPDATE</i> , <i>SQL_POS_DELETE</i> , <i>SQL_POS_ADD</i> .
SQL_POSITIONED_STATEMENTS: Lists the supported positioned SQL statements.	Returns <i>SQL_PS_POSITIONED_DELETE</i> , <i>SQL_PS_POSITIONED_UPDATE</i> , <i>SQL_PS_SELECT_FOR_UPDATE</i> .
SQL_PROCEDURE_TERM: The term used to describe a procedure.	Returns procedure.
SQL_PROCEDURES: Indicates if the server and the driver support SQL procedures and procedure invocation syntax.	Returns <i>Y</i> . The server and driver support procedures and procedure invocation syntax.
SQL_QUALIFIER_LOCATION prev. SQL_CATALOG_LOCATION: Indicates the position of the schema name in a qualified table name.	Returns <i>SQL_CL_START</i> . The catalog portion of a qualified table name is at the beginning of the name.
SQL_QUALIFIER_NAME prev. SQL_CATALOG_NAME: Indicates server support for catalog names.	Returns <i>Y</i> . The server supports catalog names.
SQL_QUALIFIER_NAME_SEPARATOR prev. SQL_CATALOG_NAME_SEPARATOR: Character separating the qualifier name from the adjacent name element.	Returns <i>'.'</i> . The server expects a <i>'.'</i> character between the qualifier and the table name.
SQL_QUALIFIER_TERM prev. SQL_CATALOG_TERM: The term used to describe a qualifier.	Returns catalog.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_QUALIFIER_USAGE prev. SQL_CATALOG_USAGE: Indicates the SQL statements that might refer to qualifiers.	Returns <i>SQL_CU_DML_STATEMENTS</i> . Catalog names can be used in <i>SELECT</i> , <i>INSERT</i> , <i>UPDATE</i> , <i>DELETE</i> , <i>SELECT FOR UPDATE</i> , and positioned <i>UPDATE</i> and <i>DELETE</i> statements.
SQL_QUALIFIER_USAGE Now SQL_CATALOG_USAGE: Identifies DML statements that support qualifier names.	Returns <i>SQL_CU_DML_STATEMENTS</i> . Qualifiers can be used in all DML statements (<i>SELECT</i> , <i>INSERT</i> , <i>UPDATE</i> , <i>DELETE</i> , <i>SELECT FOR UPDATE</i>).
SQL_QUOTED_IDENTIFIER_CASE: Indicates case sensitivity of quoted identifiers.	Returns <i>SQL_IC_SENSITIVE</i> . Quoted identifiers are case sensitive.
SQL_CATALOG_NAME_SEPARATOR prev. SQL_QUALIFIER_NAME_SEPARATOR: The character that separates the name qualifier from the name element.	Returns <i>'.'</i> . The <i>'.'</i> character is used as a separator in qualified names.
SQL_QUALIFIER_TERM: The term used to describe a qualifier.	Returns catalog
SQL_QUALIFIER_LOCATION: The position of the qualifier in a qualified table name.	Returns <i>SQL_CL_START</i> . The qualifier precedes the table name in a qualified table name.
SQL_ROW_UPDATES: Indicates if keyset-driven or mixed cursors maintain row versions or values.	Returns <i>Y</i> . Cursors maintain values for all fetched rows and can detect updates to the row values.
SQL_SCHEMA_TERM: The term used to describe a schema.	Returns schema.
SQL_SCHEMA_USAGE: Indicates the SQL statements that might refer to schemas.	Returns <i>SQL_OU_DML_STATEMENTS</i> , <i>SQL_OU_TABLE_DEFINITION</i> , <i>SQL_OU_INDEX_DEFINITION</i> , <i>SQL_OU_PRIVILEGE_DEFINITION</i> .
SQL_SCROLL_CONCURRENCY: Indicates the cursor concurrency control options supported by the server.	Returns <i>SQL_SCCO_READ_ONLY</i> , <i>SQL_SCCO_OPT_ROWVER</i> .
SQL_SCROLL_OPTIONS: Indicates the cursor scroll options supported by the server.	Returns <i>SQL_SO_FORWARD_ONLY</i> , <i>SQL_SO_KEYSET_DRIVEN</i> , <i>SQL_SO_STATIC</i> .
SQL_SEARCH_PATTERN_ESCAPE: The escape character that allows use of the wildcard characters % and _ in search patterns.	Returns <i>"</i> . The <i>"</i> character is used as an escape character for the <i>'%'</i> and <i>'_'</i> characters in search patterns.
SQL_SERVER_NAME: Indicates the name of the host.	The returned value is determined by connection properties.
SQL_SPECIAL_CHARACTERS: Indicates any special characters allowed in identifier names.	Returns <i>_</i> . The underscore character is allowed in identifier names.
SQL_SQL_CONFORMANCE: Indicates the level of SQL-92 compliance.	Returns <i>SQL_SC_SQL92_ENTRY</i> . The driver is SQL92 entry-level compliant.

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_SQL92_DATETIME_FUNCTIONS: Lists the datetime functions supported by the server.	Returns <i>SQL_SDF_CURRENT_DATE</i> , <i>SQL_SDF_CURRENT_TIME</i> , <i>SQL_SDF_CURRENT_TIMESTAMP</i> .
SQL_SQL92_FOREIGN_KEY_DELETE_RULE: Indicates the server-enforced rules for using a foreign key in a <i>DELETE</i> statement.	Returns <i>SQL_SFKD_CASCADE</i> , <i>SQL_SFKD_NO_ACTION</i> , <i>SQL_SFKD_SET_DEFAULT</i> , <i>SQL_SFKD_SET_NULL</i> .
SQL_SQL92_FOREIGN_KEY_UPDATE_RULE: Indicates the server-enforced rules for using a foreign key in an <i>UPDATE</i> statement.	Returns <i>SQL_SFKU_CASCADE</i> , <i>SQL_SFKU_NO_ACTION</i> , <i>SQL_SFKU_SET_DEFAULT</i> , <i>SQL_SFKU_SET_NULL</i> .
SQL_SQL92_GRANT: Indicates the supported <i>GRANT</i> statement clauses.	Returns <i>SQL_SG_DELETE_TABLE</i> , <i>SQL_SG_INSERT_TABLE</i> , <i>SQL_SG_REFERENCES_TABLE</i> , <i>SQL_SG_SELECT_TABLE</i> , <i>SQL_SG_UPDATE_TABLE</i> .
SQL_SQL92_NUMERIC_VALUE_FUNCTIONS: Lists the scalar numeric functions supported by the server and driver.	Returns <i>SQL_SNVF_BIT_LENGTH</i> , <i>SQL_SNVF_CHAR_LENGTH</i> , <i>SQL_SNVF_CHARACTER_LENGTH</i> , <i>SQL_SNVF_EXTRACT</i> , <i>SQL_SNVF_OCTET_LENGTH</i> , <i>SQL_SNVF_POSITION</i> .
SQL_SQL92_PREDICATES, Identifies the predicates of a <i>SELECT</i> statement supported by the server.	Returns <i>SQL_SP_EXISTS</i> , <i>SQL_SP_ISNOTNULL</i> , <i>SQL_SP_ISNULL</i> , <i>SQL_SP_OVERLAPS</i> , <i>SQL_SP_LIKE</i> , <i>SQL_SP_IN</i> , <i>SQL_SP_BETWEEN</i> , <i>SQL_SP_COMPARISON</i> , <i>SQL_SP_QUANTIFIED_COMPARISON</i> .
SQL_SQL92_RELATIONAL_JOIN_OPERATORS: Identifies the relational join operators supported by the server.	Returns <i>SQL_SRJO_CROSS_JOIN</i> , <i>SQL_SRJO_EXCEPT_JOIN</i> , <i>SQL_SRJO_FULL_OUTER_JOIN</i> , <i>SQL_SRJO_INNER_JOIN</i> , <i>SQL_SRJO_INTERSECT_JOIN</i> , <i>SQL_SRJO_LEFT_OUTER_JOIN</i> , <i>SQL_SRJO_NATURAL_JOIN</i> , <i>SQL_SRJO_RIGHT_OUTER_JOIN</i> , <i>SQL_SRJO_UNION_JOIN</i> .

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_SQL92_REVOKE: Identifies the clauses in a <i>REVOKE</i> statement that are supported by the server.	Returns <i>SQL_SR_DELETE_TABLE</i> , <i>SQL_SR_INSERT_TABLE</i> , <i>SQL_SR_REFERENCES_TABLE</i> , <i>SQL_SR_SELECT_TABLE</i> , <i>SQL_SR_UPDATE_TABLE</i> .
SQL_SQL92_ROW_VALUE_CONSTRUCTOR: Indicates the row value constructor expressions in a <i>SELECT</i> statement that are supported by the server.	Returns <i>SQL_SRVC_VALUE_EXPRESSION</i> , <i>SQL_SRVC_NULL</i> .
SQL_SQL92_STRING_FUNCTIONS: Lists the string scalar functions supported by the server and driver.	Returns <i>SQL_SSF_CONVERT</i> , <i>SQL_SSF_LOWER</i> , <i>SQL_SSF_UPPER</i> , <i>SQL_SSF_SUBSTRING</i> , <i>SQL_SSF_TRANSLATE</i> , <i>SQL_SSF_TRIM_BOTH</i> , <i>SQL_SSF_TRIM_LEADING</i> , <i>SQL_SSF_TRIM_TRAILING</i> .
SQL_SQL92_VALUE_EXPRESSIONS: Indicates the value expressions supported by the server.	Returns <i>SQL_SVE_CASE</i> , <i>SQL_SVE_CAST</i> , <i>SQL_SVE_COALESCE</i> , <i>SQL_SVE_NULLIF</i> .
SQL_STANDARD_CLI_CONFORMANCE: Indicates the CLI standard the driver conforms to.	This info_type is currently unsupported.
SQL_STATIC_CURSOR_ATTRIBUTES1: Describes the first set of static cursor attributes supported by the driver.	Returns <i>SQL_CA1_NEXT</i> , <i>SQL_CA1_ABSOLUTE</i> , <i>SQL_CA1_RELATIVE</i> , <i>SQL_CA1_BOOKMARK</i> , <i>SQL_CA1_LOCK_NO_CHANGE</i> , <i>SQL_CA1_POS_POSITION</i> , <i>SQL_CA1_POS_UPDATE</i> , <i>SQL_CA1_POS_DELETE</i> , <i>SQL_CA1_POS_REFRESH</i> , <i>SQL_CA1_BULK_ADD</i> , <i>SQL_CA1_BULK_UPDATE_BY_BOOKMARK</i> , <i>SQL_CA1_BULK_DELETE_BY_BOOKMARK</i> , <i>SQL_CA1_BULK_FETCH_BY_BOOKMARK</i> .
SQL_STATIC_CURSOR_ATTRIBUTES2: Describes the second set of static cursor attributes supported by the driver.	Returns <i>SQL_CA2_READ_ONLY_CONCURRENCY</i> , <i>SQL_CA2_OPT_ROWVER_CONCURRENCY</i> , <i>SQL_CA2_SENSITIVITY_ADDITIONS</i> , <i>SQL_CA2_SENSITIVITY_DELETIONS</i> , <i>SQL_CA2_SENSITIVITY_UPDATES</i> , <i>SQL_CA2_CRC_EXACT</i> .
SQL_STATIC_SENSITIVITY: Indicates whether changes made to a static cursor by <i>SQLSetPos()</i> or <i>UPDATE</i> or <i>DELETE</i> statements are detected by the application.	Returns <i>SQL_SS_ADDITIONS</i> , <i>SQL_SS_DELETIONS</i> , <i>SQL_SS_UPDATES</i> .

次のページに続く

表 1 – 前のページからの続き

SQL info_type argument and description**	EDB_ODBC/EDB Postgres Advanced Server returns
SQL_STRING_FUNCTIONS: Lists the scalar string functions supported by the server and driver.	Returns <i>SQL_FN_STR_CONCAT</i> , <i>SQL_FN_STR_LTRIM</i> , <i>SQL_FN_STR_LENGTH</i> , <i>SQL_FN_STR_LOCATE</i> , <i>SQL_FN_STR_LCASE</i> , <i>SQL_FN_STR_RTRIM</i> , <i>SQL_FN_STR_SUBSTRING</i> , <i>SQL_FN_STR_UCASE</i> .
SQL_SUBQUERIES: Identifies the subquery predicates to a <i>SELECT</i> statement supported by the server.	Returns <i>SQL_SQL_COMPARISON</i> , <i>SQL_SQL_EXISTS</i> , <i>SQL_SQL_IN</i> , <i>SQL_SQL_QUANTIFIED</i> .
SQL_SYSTEM_FUNCTIONS: Lists the scalar system functions supported by the server and driver.	Returns 0.
SQL_TABLE_TERM: The term used to describe a table.	Returns table.
SQL_TIMEDATE_ADD_INTERVALS: Indicates the timestamp intervals supported by the server for the <i>TIMESTAMPADD</i> scalar function.	Returns 0.
SQL_TIMEDATE_DIFF_INTERVALS: Indicates the timestamp intervals supported by the server for the <i>TIMESTAMPDIFF</i> scalar function.	Returns 0.
SQL_TIMEDATE_FUNCTIONS: Indicates the date and time functions supported by the server.	Returns <i>SQL_FN_TD_NOW</i> , <i>SQL_FN_TD_CURDATE</i> , <i>SQL_FN_TD_CURTIME</i> .
SQL_TXN_CAPABLE: Identifies the transaction support offered by the server and driver.	Returns <i>SQL_TC_ALL</i> . Transactions can contain both DML and DDL statements.
SQL_TXN_ISOLATION_OPTION: Indicates the transaction isolation level supported by the server.	Returns <i>SQL_TXN_READ_COMMITTED</i> , <i>SQL_TXN_SERIALIZABLE</i> .
SQL_UNION: Indicates server support for the <i>UNION</i> clause.	Returns <i>SQL_U_UNION</i> , <i>SQL_U_UNION_ALL</i> .
SQL_USER_NAME: Identifies the name of the user connected to a database. Can be different than the login name.	This value is determined by the connection properties.
SQL_XOPEN_CLI_YEAR: The publication year of the X/Open specification that the driver manager complies with.	This info_type is currently unsupported.

6.2 接続属性

ODBC `SQLGetConnectAttr()` および `SQLSetConnectAttr()` ファンクションを使用して、接続属性の値を取得または設定できます。

6.2.1 SQLGetConnectAttr()

`SQLGetConnectAttr()` ファンクションは、接続属性の現在の値を返します。署名は次のとおりです。

```
SQLRETURN SQLGetConnectAttr
(
    SQLHDBC conn_handle, //Input
    SQLINTEGER attribute, //Input
    SQLPOINTER value_pointer, //Output
    SQLINTEGER buffer_length, //Input
    SQLINTEGER * string_length_pointer //Output
);
```

- `conn_handle` — 接続ハンドル。
- `attribute` — 値を取得する属性を識別します。
- `value_pointer` — `attribute` 値を受け取るメモリ内の場所へのポインター。
- `buffer_length` — `attribute` が ODBC で定義され、`value_pointer` が文字列またはバイナリバッファを指している場合、`buffer_length` は `value_pointer` の長さです。`value_pointer` が固定サイズの値（整数など）を指している場合、`buffer_length` は無視されます。

EDB-ODBC が属性を定義する場合、`SQLGetConnectAttr()` は `buffer_length` パラメーターを設定します。`buffer_length` は次のとおりです。

Value type	Meaning
Character string	The length of the character string
Binary buffer	The result of <code>SQL_LEN_BINARY_ATTR(length)</code>
Fixed length data type	<code>SQL_IS_INTEGER</code> or <code>SQL_IS_INTEGER</code>
Any other type	<code>SQL_IS_POINTER</code>

- `string_length_pointer` — `value_pointer` で返すために使用可能なバイト数を受け取る `SQLINTEGER` へのポインター。`value_pointer` が `NULL` の場合、`string_length_pointer` は返されません。

この関数は、`SQL_SUCCESS`、`SQL_SUCCESS_WITH_INFO`、`SQL_NO_DATA`、`SQL_ERROR`、または `SQL_INVALID_HANDLE` を返します。

EDB-ODBC がサポートする接続属性を次の表に示します。

Attribute	Supported?	Notes
SQL_ATTR_ACCESS_MODE	No	SQL_MODE_READ_WRITE
SQL_ATTR_ASYNC_ENABLE	No	SQL_ASYNC_ENABLE_OFF
SQL_ATTR_AUTO_IPC	No	該当なし
SQL_ATTR_AUTOCOMMIT	Yes	SQL_AUTOCOMMIT、 SQL_AUTOCOMMIT_ON、 SQL_AUTOCOMMIT_OFF
SQL_ATTR_CONNECTION_POOLING	No	該当なし
SQL_ATTR_CURRENT_CATALOG	No	該当なし
SQL_ATTR_DISCONNECT_BEHAVIOR	No	該当なし
SQL_ATTR_ENLIST_IN池	Yes	win32 および条件付きコンパイルの場合
SQL_ATTR_ENLIST_IN池	No	該当なし
SQL_ATTR_LOGIN_TIMEOUT	No	SQL_LOGIN_TIMEOUT
SQL_ATTR_ODBC_CURSORS	No	該当なし
SQL_ATTR_PACKET_SIZE	No	該当なし
SQL_ATTR_QUIET_MODE	No	該当なし
SQL_ATTR_TRACE	No	該当なし
SQL_ATTR_TRACEFILE	No	該当なし
SQL_ATTR_TRANSLATE_CURSORS	No	該当なし
SQL_ATTR_TRANSLATE_CURSORS	No	該当なし
SQL_ATTR_TXN_ISOLATION	Yes	SQL_TXN_ISOLATION、 SQL_DEFAULT_TXN_ISOLATION

6.2.2 SQLSetConnectAttr()

ODBC SQLSetConnectAttr() ファンクションを使用して、接続属性の値を設定できます。関数のシグネチャは次のとおりです。

```
SQLRETURN SQLSetConnectAttr
(
    SQLHDBC conn_handle , // Input
    SQLINTEGER attribute , // Input
    SQLPOINTER value_pointer , // Input
    SQLINTEGER string_length , // Input
);
```

- `conn_handle` —接続ハンドル。
- `attribute` —値を設定する属性を識別します。
- `value_pointer` —属性が想定する値へのポインター。
- `string_length` — `attribute` が ODBC で定義され、`value_pointer` がバイナリバッファまたは文字列を指している場合、`string_length` は `value_pointer` の長さです。`value_pointer` が固定長値（整数など）を指す場合、`string_length` は無視されます。

EDB-ODBC が属性を定義する場合、アプリケーションは `string_length` パラメーターを設定します。可能な `string_length` 値を表に示します。

Value type	Meaning
Character string	The length of the character string or <i>SQL_NTS</i>
Binary buffer	The result of <i>SQL_LEN_BINARY_ATTR(length)</i>
Fixed length data type	<i>SQL_IS_INTEGER</i> or <i>SQL_IS_INTEGER</i>
Any other type	<i>SQL_IS_POINTER</i>

`SQLSetConnectAttr()` は、`SQL_SUCCESS`、`SQL_SUCCESS_WITH_INFO`、`SQL_ERROR`、`SQL_STILL_EXECUTING` または `SQL_INVALID_HANDLE` を返します。

`SQLSetConnectAttr()` は、接続ハンドルが割り当てられた後、`SQLFreeHandle()` への呼び出しで接続が閉じられるまでの間、いつでも呼び出すことができます。呼び出しによって設定されたすべての属性は、`SQLFreeHandle()` への呼び出しまで持続します。

接続属性には、設定できる特定の時間枠があります。一部の属性は接続を確立する前に設定する必要がありますが、他の属性は接続の確立後にのみ設定できます。

次の表に、接続属性とそれらを設定できる時間枠を示します。

Attribute	Set before or after establishing a connection?
<code>SQL_ATTR_ACCESS_MODE</code>	Before or after
<code>SQL_ATTR_ASYNC_ENABLE</code>	Before or after
<code>SQL_ATTR_AUTO_IPD</code>	Before or after
<code>SQL_ATTR_AUTOCOMMIT</code>	Before or after
<code>SQL_ATTR_CONNECTION_TIMEOUT</code>	Before or after
<code>SQL_ATTR_CURRENT_CATALOG</code>	Before or after
<code>SQL_ATTR_ENLIST_IN池</code>	After
<code>SQL_ATTR_ENLIST_IN池</code>	After
<code>SQL_ATTR_LOGIN_TIMEOUT</code>	Before
<code>SQL_ATTR_ODBC_CUR池</code>	Before
<code>SQL_ATTR_PACKET_SIZE</code>	Before
<code>SQL_ATTR_QUIET_MODE</code>	Before or after
<code>SQL_ATTR_TRACE</code>	Before or after
<code>SQL_ATTR_TRACEFILE</code>	Before or after
<code>SQL_ATTR_TRANSLATE_CATALOG</code>	After
<code>SQL_ATTR_TRANSLATE_CATALOG</code>	After
<code>SQL_ATTR_TXN_ISOLATION</code>	Before or after

6.3 環境属性

ODBC SQLGetEnvAttr() および SQLSetEnvAttr() ファンクションを使用して、環境属性の値を取得または設定できます。

6.3.1 SQLGetEnvAttr()

SQLGetEnvAttr() 関数を使用して、システムの環境属性の現在の値を見つけます。関数のシグネチャは次のとおりです。

```
SQLRETURN SQLGetConnectAttr
(
    SQLHDBC env_handle, // Input
    SQLINTEGER attribute, // Input
    SQLPOINTER value_ptr, // Output
    SQLINTEGER buffer_length, // Input
    SQLINTEGER * string_length_pointer // Output
);
```

- env_handle —環境ハンドル。
- attribute —値を取得する属性を識別します。
- value_pointer — attribute 値を受け取るメモリ内の場所へのポインター。
- buffer_length —属性が文字列の場合、buffer_length は value_ptr の長さです。属性の値が文字列でない場合、buffer_length は未使用です。
- string_length_pointer — value_pointer で返すために使用可能なバイト数を受け取る SQLINTEGER へのポインター。 value_pointer が NULL の場合、string_length_pointer は返されません。

この関数は、 SQL_SUCCESS 、 SQL_SUCCESS_WITH_INFO 、 SQL_NO_DATA 、 SQL_ERROR または SQL_INVALID_HANDLE を返します。

EDB-ODBC がサポートする環境属性を次の表に示します。

Attribute	Supported?	Restrictions?
SQL_ATTR_CONNEC	SQL_CP_ONE_PER_DRIVER or SQL_CP_OFF	Determined by connection properties
SQL_ATTR_ODBC_VE	(SQL_OV_ODBC3), (SQL_OV_ODBC2)	None
SQL_ATTR_OUTPUT_	SQL_SUCCESS	None

6.4 SQLSetEnvAttr()

SQLSetEnvAttr() ファンクションを使用して、環境属性の値を設定できます。関数のシグネチャは次のとおりです。

```
SQLRETURN SQLSetEnvAttr
(
    SQLHENV  env_handle , //Input
    SQLINTEGER attribute , //Input
    SQLPOINTER value_pointer , //Input
    SQLINTEGER string_length //Input
);
```

- **env_handle** —環境ハンドル。
- **attribute** —値を設定する属性を識別します。
- **value_pointer** —属性に割り当てられた値へのポインター。値は、指定された属性に応じて、NULL で終了する文字列または 32 ビット整数値です。
- **string_length** & mdash; **value_pointer** がバイナリバッファまたは文字列へのポインターの場合、**string_length** は **value_pointer** の長さです。属性に割り当てられている値が文字の場合、**string_length** はその文字列の長さです。**value_pointer** が NULL の場合、**string_length** は返されません。**value_pointer** が整数の場合、**string_length** は無視されます。

SQLSetEnvAttr() は、SQL_SUCCESS、SQL_INVALID_HANDLE、SQL_ERROR または SQL_SUCCESS_WITH_INFO を返します。アプリケーションは、接続ハンドルを割り当てる前に SQLSetEnvAttr() を呼び出す必要があります。環境属性に適用されたすべての値は、接続に対して SQLFreeHandle() が呼び出されるまで持続します。ODBC バージョン 3.x では、複数の環境ハンドルを同時に割り当てることができます。

次の表に、SQLSetAttr() で設定できる環境属性をリストします。

Attribute	Value_pointer type	Restrictions?
SQL_ATTR_ODBC_VI	32-bit integer	Set this attribute before the application calls any function that includes an <i>SQLHENV</i> argument.
SQL_ATTR_OUTPUT_	32-bit integer	Defaults to <i>SQL_TRUE</i> . Calls that set this attribute to <i>SQL_FALSE</i> return <i>SQL_ERROR/SQLSTATEHYC00</i> (feature not implemented).

6.5 ステートメント属性

ODBC SQLGetStmtAttr() および SQLSetStmtAttr() 関数を使用して、ステートメント属性の値を取得および設定できます。

6.5.1 SQLGetStmtAttr()

SQLGetStmtAttr() ファンクションは、ステートメント属性の現在の値を返します。署名は次のとおりです。

```
SQLRETURN SQLGetConnectAttr
(
    SQLHDBC stmt_handle, //Input
    SQLINTEGER attribute, //Input
    SQLPOINTER value_ptr, //Output
    SQLINTEGER buffer_length, //Input
    SQLINTEGER * string_length_pointer //Output
);
```

- `stmt_handle` —ステートメントハンドル。
- `attribute` —属性値。
- `value_pointer` — `attribute` 値を受け取るメモリ内の場所へのポインター。
- `buffer_length` —属性が ODBC で定義されている場合、`buffer_length` は `value_pointer` の長さです（`value_pointer` が文字列またはバイナリバッファを指している場合）。`value_pointer` が整数を指す場合、`buffer_length` は無視されます。

EDB-ODBC が属性を定義する場合、アプリケーションは `buffer_length` パラメーターを設定します。`buffer_length` は次のとおりです。

Value type	Meaning
Character string	The length of the character string
Binary buffer	The result of <i>SQL_LEN_BINARY_ATTR(length)</i>
Fixed length data type	<i>SQL_IS_INTEGER</i> or <i>SQL_IS_INTEGER</i>
Any other type	<i>SQL_IS_POINTER</i>

- `string_length_pointer` —要求された値を保持するために必要なバイト数を受け取る `SQLINTEGER` へのポインター。`value_pointer` が NULL の場合、`string_length_pointer` は返されません。

この関数は、SQL_SUCCESS、SQL_SUCCESS_WITH_INFO、SQL_ERROR または SQL_INVALID_HANDLE を返します。

Attribute	Supported?	Restrictions?
SQL_ATTR_APP_PAR	Yes	

次のページに続く

表 9 – 前のページからの続き

Attribute	Supported?	Restrictions?
SQL_ATTR_APP_ROW	Yes	
SQL_ATTR_ASYNC_E	No	
SQL_ATTR_CONCUR_I	Yes	SQL_CONCUR_READ_ONLY
SQL_ATTR_CURSOR_	Yes	
SQL_ATTR_CURSOR_	Yes	SQL_CURSOR_FORWARD_ONLY
SQL_ATTR_CURSOR_	Yes	SQL_INSENSITIVE
SQL_ATTR_ENABLE_	No	
SQL_ATTR_FETCH_B	Yes	
SQL_ATTR_IMP_PAR_	Yes	
SQL_ATTR_IMP_ROW	Yes	
SQL_ATTR_KEYSET_	No	
SQL_ATTR_MAX_LEN	No	
SQL_ATTR_MAX_ROW	No	
SQL_ATTR_METADATA	Yes	
SQL_ATTR_NOSCAN	No	
SQL_ATTR_PARAM_E	Yes	ODBC V2.0
SQL_ATTR_PARAM_E	Yes	
SQL_ATTR_PARAM_C	Yes	
SQL_ATTR_PARAM_S	Yes	
SQL_ATTR_PARAMS_	Yes	
SQL_ATTR_PARAMS_I	Yes	
SQL_ATTR_QUERY_T	No	
SQL_ATTR_RETRIEVE	No	
SQL_ATTR_ROW_BIN	Yes	
SQL_ATTR_ROW_BIN	No	
SQL_ATTR_ROW_NUM	Yes	
SQL_ATTR_ROW_OPE	Yes	
SQL_ATTR_ROW_STA	Yes	
SQL_ATTR_ROWS_FE	Yes	
SQL_ATTR_ROW_ARI	Yes	
SQL_ATTR_SIMULAT	No	
SQL_ATTR_USE_BOO	Yes	
SQL_ROWSET_SIZE	Yes	

6.5.2 SQLSetStmtAttr()

SQLSetStmtAttr() ファンクションを使用して、環境属性の値を設定できます。署名は次のとおりです。

```
SQLRETURN SQLSetStmtAttr
(
    SQLHENV stmt_handle , //Input
    SQLINTEGER attribute , //Input
```

(次のページに続く)

(前のページからの続き)

```
SQLPOINTER value_pointer , //Input
SQLINTEGER string_length //Input
);
```

- stmt_handle —環境ハンドル。
- attribute —値を設定するステートメント属性を識別します。
- value_pointer —属性に割り当てられる値を保持するメモリ内の場所へのポインター。value_pointer は、以下へのポインターにすることができます。
- ヌル終了文字列 - バイナリバッファ - ドライバーによって定義された値 - SQLLEN、SQLULEN または SQLUSMALLINT 型の値

value-pointer は、オプションで次の値のいずれかを保持できます。 - ODBC 記述子ハンドル - SQLINTEGER 値 - SQLULEN 値 - 符号付き INTEGER（属性がドライバー固有の値の場合）

- string_length & mdash; attribute が ODBC で定義され、value_pointer がバイナリバッファまたは文字列を指している場合、string_length は value_pointer の長さです。value_pointer が整数を指している場合、string_length は無視されます。EDB-ODBC が属性を定義している場合、アプリケーションは string_length パラメーターを設定します。可能な string_length 値は次のとおりです。

Value type	Meaning
Character string	The length of the character string or <i>SQL_NTS</i>
Binary buffer	The result of <i>SQL_LEN_BINARY_ATTR(length)</i>
Fixed length data type	<i>SQL_IS_INTEGER</i> or <i>SQL_IS_INTEGER</i>
Any other type	<i>SQL_IS_POINTER</i>

6.6 エラー処理

ODBC SQLGetDiagRec() 関数を使用して、前述の ODBC 関数の診断情報を取得できます。

6.6.1 SQLGetDiagRec()

SQLGetDiagRec() ファンクションは、属性値を取得または設定する ODBC ファンクションによって書き込まれた診断レコードから、ステータスおよびエラー情報を返します。署名は次のとおりです。

```
SQLRETURN SQLGetDiagRec
(
    SQLSMALLINT handle_type, // Input
    SQLHANDLE handle, // Input
    SQLSMALLINT record_number, // Input
    SQLCHAR *SQLState_pointer, // Output
```

(次のページに続く)

(前のページからの続き)

```
SQLINTEGER *native_error_pointer, // Output
SQLCHAR *error_text_pointer, // Output
SQLSMALLINT buffer_length, // Input
SQLSMALLINT *text_length_pointer // Output
);
```

- **handle_type** — **handle** 引数のハンドルタイプ。 **handle_type** は次のいずれかでなければなりません。
- **SQL_HANDLE_ENV** は、環境ハンドルを指定します。 - **SQL_HANDLE_STMT** は、ステートメントハンドルを指定します。 - **SQL_HANDLE_DBC** は、接続ハンドルを指定します。 - **SQL_HANDLE_DESC** は、記述子ハンドルを指定します。
- **handle** — 属性エラーメッセージに関連付けられたハンドル。
- **record_number** — アプリケーションが情報を求めているステータスレコード（1 以上でなければなりません）。
- **SQLState_pointer** — レコードから **SQLState** エラーコードを受け取るメモリバッファーへのポインター。
- **native_error_pointer** — データソース（ **SQL_DIAG_NATIVE** フィールドに含まれます）のネイティブエラーメッセージを受け取るバッファーへのポインター。
- **error_text_pointer** — エラーテキスト（ **SQL_DIAG_MESSAGE_TEXT** フィールドに含まれます）を受け取るメモリバッファーへのポインター。
- **buffer_length** — **error_text** バッファーの長さ。
- **text_length_pointer** — **error_text_pointer** フィールドのサイズ（文字単位）を受け取るバッファーへのポインター。 **error_text_pointer** パラメーターの文字数が使用可能な数（**buffer_length** 内）を超える場合、 **error_text_pointer** は切り捨てられます。

SQLGetDiagRec() は、 **SQL_SUCCESS**、 **SQL_ERROR**、 **SQL_INVALID_HANDLE**、 **SQL_SUCCESS_WITH_DATA** または **SQL_NO_DATA** を返します。

6.7 サポートされている ODBC API 関数

次の表に、ODBC API 関数を示します。API が EDB-ODBC ドライバーでサポートされている場合、右側の列には [はい] が指定されます。ODBC **SQLGetFunctions()** ファンクション（**SQL_API_ODBC3_ALL_FUNCTIONS** のファンクション ID を指定）を使用して、このリストの現在のバージョンを返します。

ODBC API function name	Supported by EDB-ODBC?
SQLAllocConnect()	Yes
SQLAllocEnv()	Yes
SQLAllocStmt()	Yes

次のページに続く

表 11 – 前のページからの続き

ODBC API function name	Supported by EDB-ODBC?
SQLBindCol()	Yes
SQLCancel()	Yes
SQLColAttributes()	Yes
SQLConnect()	Yes
SQLDescribeCol()	Yes
SQLDisconnect()	Yes
SQLError()	Yes
SQLExecDirect()	Yes
SQLExecute()	Yes
SQLFetch()	Yes
SQLFreeConnect()	Yes
SQLFreeEnv()	Yes
SQLFreeStmt()	Yes
SQLGetCursorName()	Yes
SQLNumResultCols()	Yes
SQLPrepare()	Yes
SQLRowCount()	Yes
SQLSetCursorName()	Yes
SQLSetParam()	Yes
SQLTransact()	Yes
SQLColumns()	Yes
SQLDriverConnect()	Yes
SQLGetConnectOption()	Yes
SQLGetData()	Yes
SQLGetFunctions()	Yes
SQLGetInfo()	Yes
SQLGetStmtOption()	Yes
SQLGetTypeInfo()	Yes
SQLParamData()	Yes
SQLPutData()	Yes
SQLSetConnectOption()	Yes
SQLSetStmtOption()	Yes
SQLSpecialColumns()	Yes
SQLStatistics()	Yes
SQLTables()	Yes
SQLBrowseConnect()	No
SQLColumnPrivileges()	No
SQLDataSources()	Yes
SQLDescribeParam()	No
SQLExtendedFetch()	Yes
SQLForeignKeys()	Yes

次のページに続く

表 11 – 前のページからの続き

ODBC API function name	Supported by EDB-ODBC?
SQLMoreResults()	Yes
SQLNativeSQL()	Yes
SQLNumParams()	Yes
SQLParamOptions()	Yes
SQLPrimaryKeys()	Yes
SQLProce- dureColumns()	Yes
SQLProcedures()	Yes
SQLSetPos()	Yes
SQLSetScrollOptions()	No
SQLTablePrivileges()	Yes
SQLDrivers()	Yes
SQLBindParameter()	Yes
SQLAllocHandle()	Yes
SQLBindParam()	Yes
SQLCloseCursor()	Yes
SQLColAttribute()	Yes
SQLCopyDesc()	Yes
SQLEndTran()	Yes
SQLFetchScroll()	Yes
SQLFreeHandle()	Yes
SQLGetConnectAttr()	Yes
SQLGetDescField()	Yes
SQLGetDescRec()	Yes
SQLGetDiagField()	Yes
SQLGetDiagRec()	Yes
SQLGetEnvAttr()	Yes
SQLGetStmtAttr()	Yes
SQLSetConnectAttr()	Yes
SQLSetDescField()	Yes
SQLSetDescRec()	No
SQLSetEnvAttr()	Yes
SQLSetStmtAttr()	Yes
SQLBulkOperations()	Yes

6.8 サポートされているデータ型

EDB-ODBC は、以下の ODBC データ型をサポートしています。

ODBC data type	Corresponding EDB Postgres Advanced Server data type
SQL_BIGINT	PG_TYPE_INT8
SQL_BINARY	PG_TYPE_BYTEA
SQL_BIT	PG_TYPE_BOOL or PG_TYPE_CHAR
SQL_CHAR	PG_TYPE_BPCHAR
SQL_TYPE_DATE	PG_TYPE_DATE
SQL_DECIMAL	PG_TYPE_NUMERIC
SQL_DOUBLE	PG_TYPE_FLOAT8
SQL_FLOAT	PG_TYPE_FLOAT8
SQL_INTEGER	PG_TYPE_INT4
SQL_LONGVARBINARY	PG_TYPE_BYTEA
SQL_LONGVARCHAR	PG_TYPE_VARCHAR or PG_TYPE_TEXT
SQL_NUMERIC	PG_TYPE_NUMERIC
SQL_NUMERIC	PG_TYPE_NUMERIC
SQL_REAL	PG_TYPE_FLOAT4
SQL_SMALLINT	PG_TYPE_INT2
SQL_TYPE_TIME	PG_TYPE_TIME
SQL_TYPE_TIMESTAMP	PG_TYPE_DATETIME
SQL_TINYINT	PG_TYPE_INT2
SQL_VARBINARY	PG_TYPE_BYTEA
SQL_VARCHAR	PG_TYPE_VARCHAR

6.9 ADO ユーザーの前提条件

Command.Execute を実行する前に Command.Prepared = True を実行する必要があります。

6.10 スレッドセーフ

EDB-ODBC はスレッドセーフです。

第7章 Scram compatibility

EDB ODBC コネクタは、EDB Postgres Advanced Server バージョン 10 以降の SCRAM-SHA-256 サポートを提供します。このサポートは、EDB ODBC 10.01.0000.01 リリース以降で利用できます。