



ODBC Connector

リリース *12.00.0000.01*

EDB Postgres™ Advanced Server ODBC Connector Guide

2020 年 06 月 30 日

目次

第 1 章	新機能	3
第 2 章	要件の概要	5
2.1	サポートされているバージョン	5
2.2	サポートされているプラットフォーム	5
第 3 章	EDB-ODBC の概要	7
3.1	EDB-ODBC のインストール	7
3.1.1	RPM パッケージを使用したコネクタのインストール	8
3.1.2	SLES12 ホストへのコネクタのインストール	11
3.1.3	Debian または Ubuntu ホストへの DEB パッケージのインストール	13
3.1.4	グラフィカルインストーラーを使用したコネクタのインストール	14
第 4 章	データソースの作成	19
第 5 章	EDB-ODBC 接続プロパティ	21
5.1	Windows でのデータソース定義の追加	29
5.2	Linux でのデータソース定義の追加	39
第 6 章	EDB-ODBC ドライバーの機能	47
6.1	SQLGetInfo ()	47
6.2	接続属性	64
6.2.1	SQLGetConnectAttr ()	64
6.2.2	SQLSetConnectAttr ()	66
6.3	環境属性	68
6.3.1	SQLGetEnvAttr ()	68
6.3.2	SQLSetEnvAttr ()	70
6.4	ステートメント属性	72
6.4.1	SQLGetStmtAttr ()	72
6.4.2	SQLSetStmtAttr ()	74
6.5	エラー処理	76
6.5.1	SQLGetDiagRec ()	76
6.6	サポートされる ODBC API 関数	78
6.7	サポートされているデータ型	81

6.8	スレッドセーフティ	82
第 7 章	セキュリティと暗号化	83
7.1	スクラムの互換性	83
第 8 章	結論	85
索引		87

ODBC (OpenDatabaseConnectivity) 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任意の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 は、EDBPostgresAdvancedServer (AdvancedServer) と ODBC 準拠のアプリケーション間の接続を提供します。

このガイドには、EDB-ODBC のインストール情報と、EDB-ODBC のデータソース定義の作成に関する情報が含まれています。このガイドには、EDB-ODBC でサポートされる ODBC 機能の詳細を説明する参照情報も含まれています。

第 1 章

新機能

AdvancedServerODBC コネクタ 12.00.0000.01 を作成するために、以下の機能が追加されます。

- EDBODBC コネクタは、EDBPostgresAdvancedServer12 をサポートするようになりました。
- EDBODBC コネクタは、WindowsServer2019 プラットフォームでもサポートされるようになりました。
- プロジェクトは、上流のコミュニティドライババージョン 12.00.0000.00 とマージされます。

第 2 章

要件の概要

2.1 サポートされているバージョン

AdvancedServerODBC コネクタは、AdvancedServer バージョン 9.4 以降で認定されています。

2.2 サポートされているプラットフォーム

AdvancedServerODBCConnector ネイティブパッケージは、次のプラットフォームでサポートされています。

64 ビット Linux:

- RedHatEnterpriseLinux (x86_64) 6.x および 7.x
- CentOS (x86_64) 6.x および 7.x
- OELLinux6.x および 7.x
- RHEL または CentOS7.x を実行する PPC-LE8
- SLES12.x
- Debian9.x
- Ubuntu18.04

AdvancedServerODBCConnector グラフィカルインストーラーは、次の Windows プラットフォームでサポートされています。

64 ビット Windows:

- Windows サーバー 2019
- Windows サーバー 2016
- Windows サーバー 2012R2

- ウィンドウズ 10
- ウィンドウズ 8
- Windows7

32 ビット Windows:

- ウィンドウズ 10
- ウィンドウズ 8
- Windows7

第 3 章

EDB-ODBC の概要

EDB-ODBC は、ODBC 準拠のクライアントアプリケーションが AdvancedServer 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 コネクタを使用すると、他のデータベースと連携して動作するように設計されたアプリケーションを AdvancedServer で実行できます。EDB-ODBC 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、AdvancedServer から結果を取得する方法を提供します。

EDB-ODBC は一定レベルのアプリケーションの移植性を提供しますが、移植性には制限があることに注意してください。EDB-ODBC は接続を提供しますが、コマンドの互換性を保証するものではありません。別のデータベースで受け入れ可能なコマンドは、AdvancedServer 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。

- クライアントアプリケーション-ODBC のバインディングを持つ言語で記述されている
- ODBC アドミニストレーター-Windows または Linux の名前付き接続を処理します
- データベース固有の ODBC ドライバー-EDB-ODBC
- ODBC 準拠サーバー-EDBPostgresAdvancedServer

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

3.1 EDB-ODBC のインストール

EDB-ODBC コネクタは、EDBPostgresAdvancedServer のグラフィカルインストーラーまたは RPM インストーラーと共に配布およびインストールされます。

3.1.1 RPM パッケージを使用したコネクタのインストール

ODBC コネクタをインストールする前に、以下を行う必要があります。

epel-release パッケージをインストールします:

- RHEL または CentOS7 の場合:

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

- RHEL または CentOS8 の場合:

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

注釈: (/etc/yum.repos.d にある) CentOS-Base.repo ファイルの [extras] リポジトリ定義を有効にする必要があるかもしれません。

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-odbc をインストールします。

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。

ファイルを変更し、ユーザー名とパスワードを提供します

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
```

ODBC コネクタのインストール

変更を構成ファイルに保存した後、`yum install` コマンドを使用して RHEL または CentOS7 に ODBC コネクタをインストールできます。たとえば、次のコマンドは ODBC コネクタをインストールします。

- RHEL または CentOS7 の場合:

```
yum install edb-odbc
yum install edb-odbc-devel
```

- RHEL または CentOS8 の場合:

```
dnf install edb-odbc
dnf install edb-odbc-devel
```

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。

RPM インストールの更新

既存の 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、次のように入力します:

- RHEL または CentOS7 の場合:

```
yum upgrade edb-repo
```

- RHEL または CentOS8 の場合:

```
dnf upgrade edb-repo
```

yum or dnf will update the `edb.repo` file to enable access to the current EDB repository, configured to connect with the credentials specified in your `edb.repo` file. Then, you can use yum or dnf to upgrade any installed packages:

- RHEL または CentOS7 の場合:

```
yum upgrade edb-odbc
```

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

- RHEL または CentOS8 の場合:

```
dnf upgrade edb-odbc
```

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

3.1.2 SLES12 ホストへのコネクタのインストール

zypper パッケージマネージャーを使用して、SLES12 ホストにコネクタをインストールできます。Zypper は、パッケージをインストールするときにパッケージの依存関係を満たそうとしますが、EnterpriseDB でホストされていない特定のリポジトリへのアクセスが必要です。

コネクタをインストールする前に、次のコマンドを使用して EnterpriseDB リポジトリ構成ファイルを SLES ホストに追加します。

```
zypper addrepo https://zypp.enterprisedb.com/suse/epas12-sles.repo
zypper addrepo https://zypp.enterprisedb.com/suse/epas-sles-tools.repo
zypper addrepo https://zypp.enterprisedb.com/suse/epas-sles-dependencies.repo
```

各コマンドはリポジトリ設定ファイルを /etc/zypp/repos.d directory に作成します。ファイルの名前は次のとおりです。

- Edbas12suse.repo
- edbasdependencies.repo
- edbastools.repo

リポジトリ設定ファイルを作成したら、zypper refresh コマンドを使用して SLES ホストのメタデータを更新し、EnterpriseDB リポジトリを含めます。

```
/etc/zypp/repos.d # zypper refresh
Repository 'SLES12-12-0' is up to date.
Repository 'SLES12-Pool' is up to date.
Repository 'SLES12-Updates' is up to date.
Retrieving repository 'EDB Postgres Advanced Server 12 12 - x86_64'
metadata -----[\\]

Authentication required for
'https://zypp.enterprisedb.com/12/suse/suse-12-x86_64'
User Name:
Password:

Retrieving repository 'EDB Postgres Advanced Server 12 12 - x86_64'
metadata.....[done]
Building repository 'EDB Postgres Advanced Server 12 12 - x86_64'
cache.....[done]
All repositories have been refreshed.
...
```

User Name と Password の入力を求められたら、EnterpriseDB リポジトリの接続認証情報を入力します。資格情報が必要な場合は、`EnterpriseDB` にお問い合わせください<<https://www.enterprisedb.com/repository-access-request/>>`。

EDBPostgresAdvancedServer またはサポートコンポーネントをインストールする前に、SUSEConnect と SUSEPackageHub 拡張を SLES ホストに追加し、ホストを SUSE に登録して、SUSE リポジトリへのアクセスを許可する必要があります。次のコマンドを使用します。

```
zypper install SUSEConnect
SUSEConnect -p PackageHub/12/x86_64
SUSEConnect -p sle-sdk/12/x86_64
```

SUSE ホストの登録の詳細については、ここをクリックしてください<<https://www.suse.com/support/kb/doc/?id=7016626>>`。

次に、zypper ユーティリティを使用してコネクタをインストールします。

```
zypper install edb-odbc
zypper install edb-odbc-devel
```

3.1.3 Debian または Ubuntu ホストへの DEB パッケージのインストール

Debian または Ubuntu ホストに DEB パッケージをインストールするには、EnterpriseDB リポジトリへのアクセスを許可する資格情報が必要です。リポジトリの認証情報をリクエストするには、[ここをクリックしてください](https://www.enterprisedb.com/repository-access-request/) [<https://www.enterprisedb.com/repository-access-request/>](https://www.enterprisedb.com/repository-access-request/)。

次の手順では、EnterpriseDBapt リポジトリを使用して DEB パッケージをインストールする方法について説明します。コマンドを使用する場合、username と password を EnterpriseDB によって提供される資格情報に置き換えます。

1. スーパーユーザー権限を想定します。

```
sudo su -
```

2. EnterpriseDB リポジトリを設定します。

```
sh -c 'echo "deb
https://username:password@apt.enterprisedb.com/$(lsb_release
-cs)-edb/ $(lsb_release -cs) main" >
/etc/apt/sources.list.d/edb-$(lsb_release -cs).list'
```

3. 安全な APT リポジトリのサポートをシステムに追加します。

```
apt-get install apt-transport-https
```

4. EDB 署名鍵を追加します。

```
wget -q -O - https://username:password
@apt.enterprisedb.com/edb-deb.gpg.key \& apt-key add -
```

5. リポジトリのメタデータを更新します。

```
apt-get update
```

6. DEB パッケージをインストールします。

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

3.1.4 グラフィカルインストーラーを使用したコネクタのインストール

EnterpriseDB コネクタインストールウィザードを使用して、システムに ODBC コネクタを追加できます。ウィザードは`ここにありますが<<https://www.enterprisedb.com/advanced-downloads/>>`。

インストーラーをダウンロードし、インストーラーアイコンを右クリックして、コンテキストメニューから [0] を選択します。

Language Selection ポップアップが開いたら、インストール言語を選択し、OK をクリックして Setup ウィンドウ（下図に示す）に進みます。

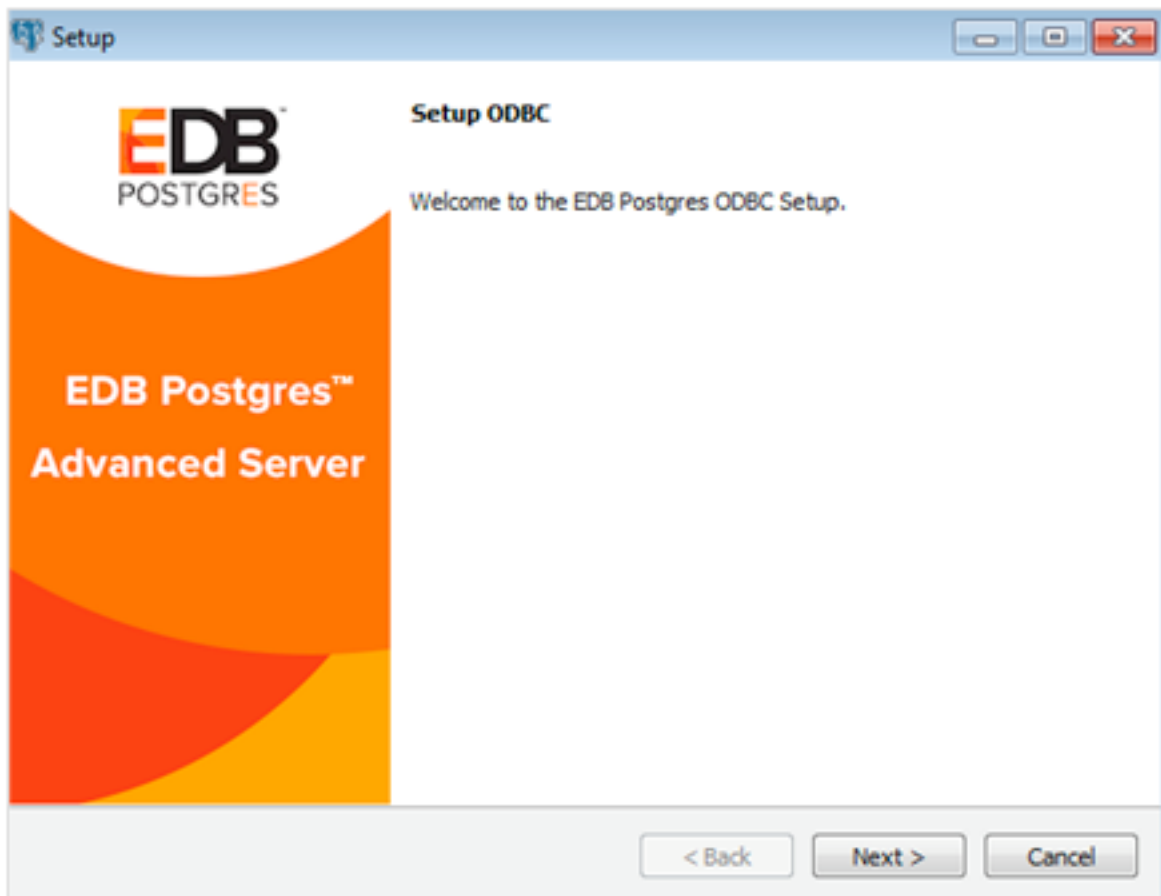


図 1 ODBC コネクタインストールウィザード。

Next をクリックして続行します。

Installation Directory ダイアログを使用してコネクタをインストールするディレクトリを指定し、Next をクリックして続行します。

Ready to Install ダイアログの Next をクリックしてインストールを開始します。ポップアップダイアログは、インストールウィザードの進行状況を確認します。

セットアップが完了したことがウィザードから通知されたら、Finish ボタンをクリックしてダイアログを終了

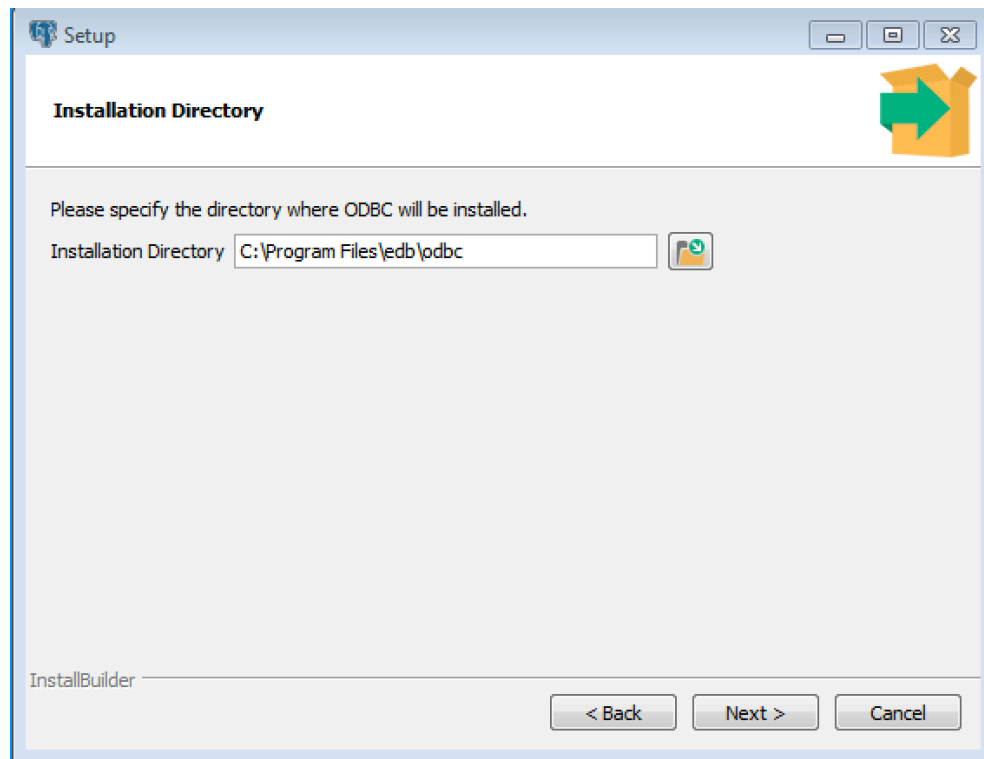


図 2 インストールダイアログ。

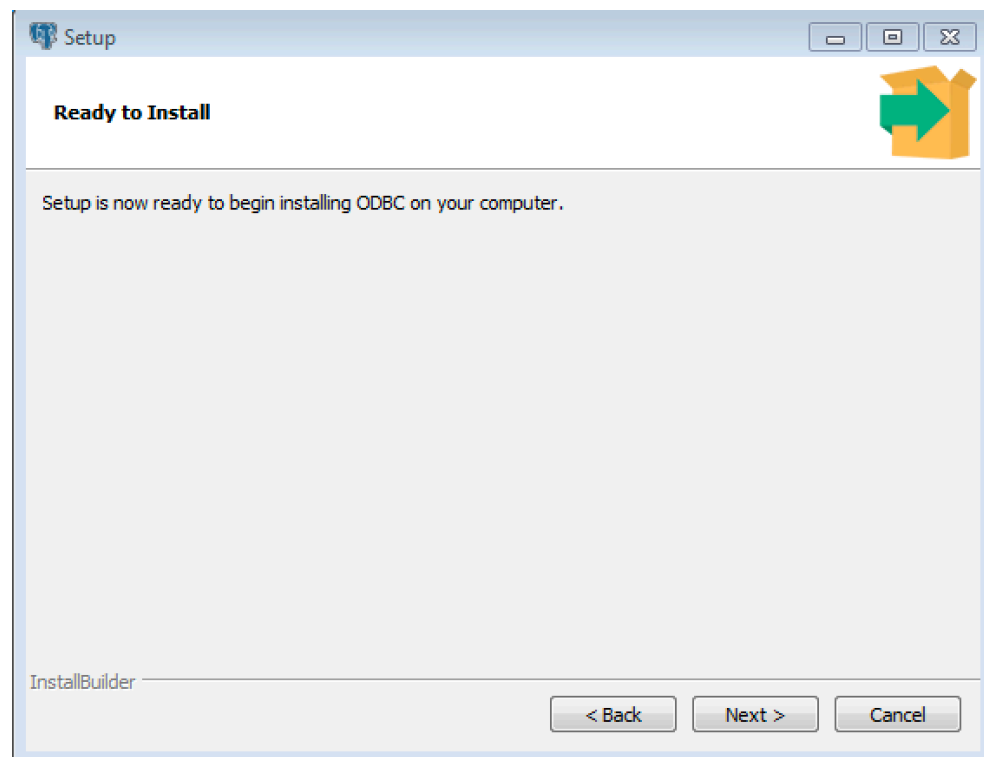


図 3 [インストールの準備完了] ダイアログ。

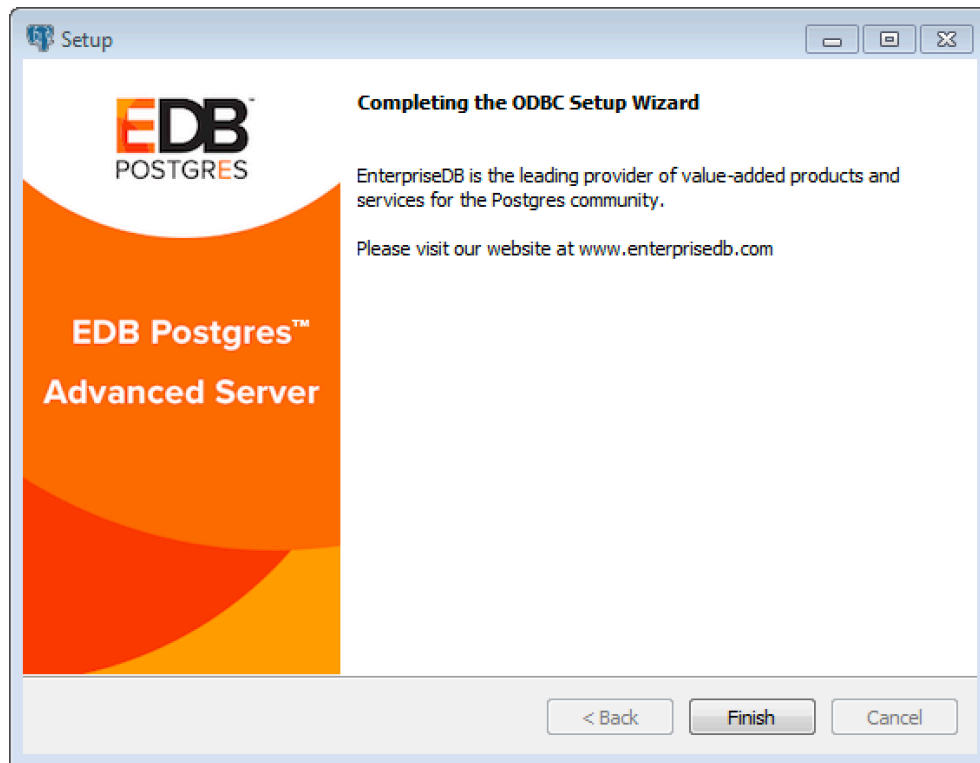


図 4 インストールが完了しました。

します。

StackBuilderPlus を使用して、既存の AdvancedServer インストールでコネクタを追加または更新することもできます。StackBuilderPlus を開くには、Windows アプリメニューまたは Linux の Start メニューから StackBuilderPlus を選択します。

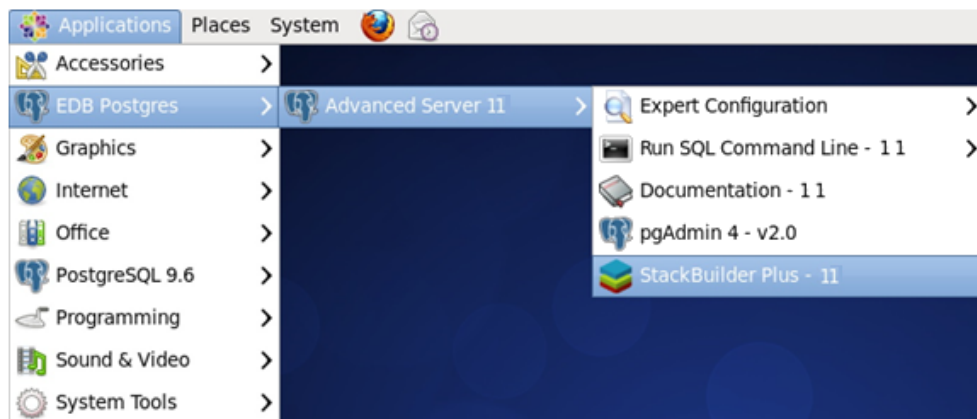


図 5 StackBuilderPlus の起動

StackBuilderPlus が開いたら、画面の指示に従います。ツリーコントロールの Database Drivers ノードから EnterpriseDB ODBC Connector オプションを選択します。

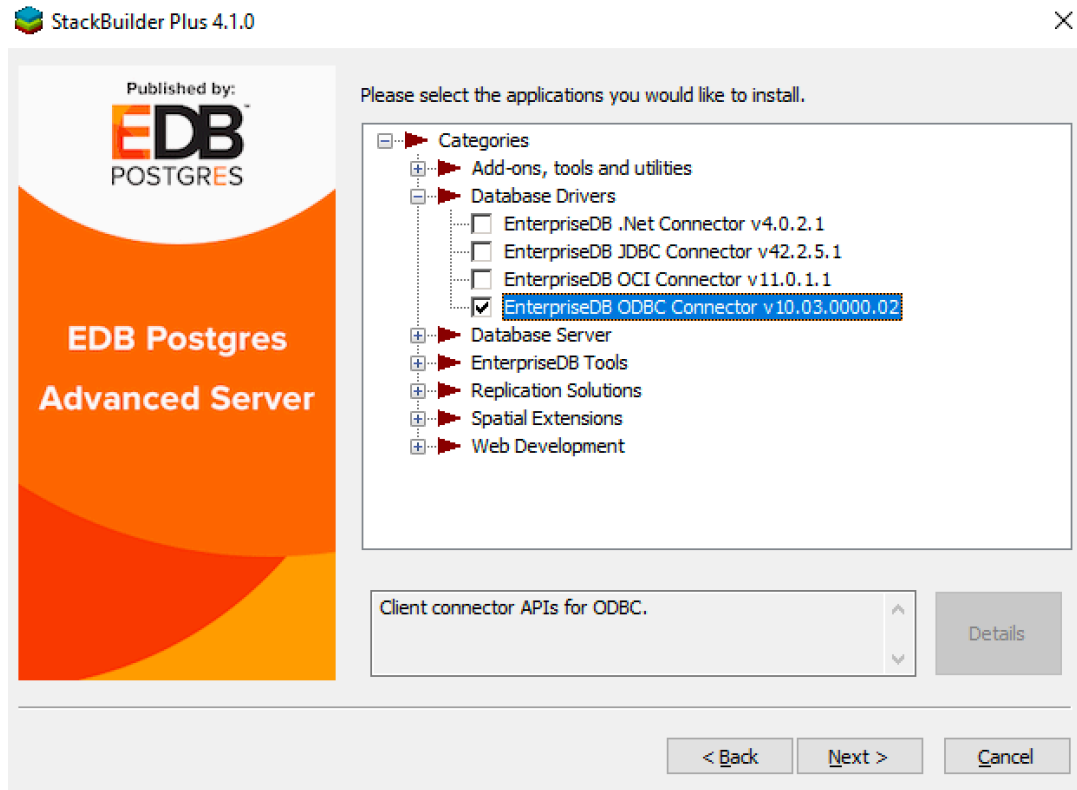


図 6 コネクターインストーラーの選択

画面上のウィザードの指示に従って、EnterpriseDB コネクタのインストールを追加または更新します。

第 4 章

データソースの作成

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、通常、データソース名（`DSN`とも呼ばれます）が提供されます。ドライバーマネージャーは、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、データソースはファイルで定義されます。このファイルの名前は通常/etc/odbc.ini ですが、名前（および場所）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確認します。

```
$ 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。

データソース定義（`ファイル DSN`と呼ばれる）を任意の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 ドライバー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' のペアが続きます。

ODBC 管理ユーティリティは、ODBC 接続用の名前付きデータソースを作成します。ほとんどの場合、ODBC 管理者ユーティリティはオペレーティングシステムと共に配布されます（Windows または unixODBC を使用し

ている場合、ツールは ODBC Data Source Administrator と呼ばれます)。オペレーティングシステムに ODBC 管理者が含まれていない場合は、サードパーティのオプションをオンラインで利用できます。

セクション Adding a Data Source Definition in Windows と Adding a Data Source Definition in Linux では、各オペレーティングシステムで利用できるグラフィカルツールを使用して、Windows と Linux にデータソースを追加する方法を説明しています。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ように求められます。セクション EDB-ODBC Connection Properties には optional 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で接続プロパティを指定するか、テキストエディターで `odbc.ini` ファイルを編集できます。

第 5 章

EDB-ODBC 接続プロパティ

The following table describes the connection properties that you can specify through the dialogs in the graphical connection manager tools, or in the `odbc.ini` file that defines a named data source. The columns identify the connection property (as it appears in the ODBC Administrator dialogs), the corresponding keyword (as it appears in the `odbc.ini` file), the default value of the property, and a description of the connection property.

Property	キーワード名	デフォルト値	説明
データベース	データベース	なし	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。
サーバ	サーバー名	ローカルホスト	接続す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。
説明	説明		データソースの説明的な名前。
ユーザー名	ユーザー名		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。
パスワード	パスワード		この名前付きデータソースに関連付けられているユーザーのパスワード。
CPOutput	CPOutput	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	ポストマスターがリッスンしている TCP ポート。
プロトコル	プロトコル	7.4	指定した場合、ドライバーに指定されたプロトコルバージョンを使用するように強制します。

次のページに続く

表 1 – 前のページからの続き

Property	キーワード名	デフォルト値	説明
エラー時のロールバックのレベル	[プロトコル] オプションを使用して、ロールバック動作を指定します。	トランザクションレベル	ドライバー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します 2-ステートメントをロールバックする
使用回数	UsageCount	1	このドライバを使用しているインストールの数。
読み取り専用	読み取り専用	番号	接続が READONLY であることを指定します。
システムテーブルを表示	ShowSystemTables	番号	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 オプション:列の表示	ShowOidColumn	番号	有効な場合、SQLColumns () 関数は OID 列を報告します。
OID オプション:偽のインデックス	FakeOidIndex	番号	有効な場合、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。
キーセットクエリの最適化	KSQO	オン	有効にすると、キーセットクエリ (MSJet 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適用されます。
一意のインデックスを認識する	UniqueIndex	オン	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。有効でない場合、SQLStatistics () 関数は、インデックスが重複する値を許可することを報告します。
宣言/フェッチを使用する	UseDeclareFetch	オフ	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、値 1 を指定します。UseDeclareFetch を無効にするには、値 0 を指定します。
CommLog	CommLog	オフ	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がログファイルに記録されます。

次のページに続く

表 1 – 前のページからの続き

Property	キーワード名	デフォルト値	説明
文の解析	解析	オフ	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。
FreeStmt としてキャンセル	CancelAsFreeStmt	オフ	有効な場合、SQLCancel () 関数はユーザーに代わって SQLFreeStmt (SQL_Close) を呼び出します。
MyLog	デバッグ	オフ	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_ です。<process-id>;Linux では、ファイル名は/tmp/mylog_ です。<username><process-id>。ログ。
不明なサイズ	不明なサイズ	最大	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。タイプに許可されている最大サイズをドライバーに報告させるには、0 を指定します。ドライバーに不明な長さを報告させるには 1 を指定し、最長の値を見つけるためにドライバーに結果セットを強制的に探索させるには 2 を指定します。UseDeclareFetch を有効にしている場合は、2 を指定しないでください。
LongVarchar としてのテキスト	長いテキスト varchar	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。
LongVarchar として不明	LONGVARCHAR	誤り	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。チェックされていない場合、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、不明なタイプの値は Y 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	キーワード名	デフォルト値	説明
Char としてのブール	BoolsAsChar	オン	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。
マックス・バーチャ	MaxVarcharSize	255	有効にすると、ドライバーは MaxVarcharSize より長い VARCHAR および BPCHAR 値を SQL_LONGVARCHAR 値として扱います
最大ロング Varchar サイズ	MaxLongVarcharSize	8190	TextAsLongVarChar がオンの場合、ドライバーは TEXT 値が MaxLongVarcharSize バイト長であることを報告します。 UnknownAsLongVarChar がオンの場合、不明なタイプの列の長さは MaxLongVarcharSize バイトです。それ以外の場合、それらは MaxVarcharSize バイトの長さであると報告されます。
キャッシュサイズ	フェッチ	100	UseDeclareFetch が有効な場合に、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 プレフィックス	ExtraSysTablePrefixes	dd_;	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_ が含まれています。
挿入の累積行数	MapSqlParcNoBatch	オフ/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、個々の行数がパラメーター設定ごとに返されます。デフォルトでは、このオプションは無効になっています。

次のページに続く

表 1 – 前のページからの続き

Property	キーワード名	デフォルト値	説明
LF<->CR/LF 変換	LFConversion	システム依存	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改行文字を改行/改行ペアに変換し、改行/改行ペアを改行に戻すようにドライバーに指示します文字値をサーバーに送信するときの文字。デフォルトでは、このオプションは有効になっています。
更新可能なカーソル	更新可能なカーソル	オフ	SQLSetPos () または SQLBulkOperations () 関数を使用して、位置付けされた UPDATE および DELETE 操作を許可します。
ByteaasLongVarBinary	ByteaAsLongVarBinary	オフ	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。
ByteaasLO	ByteaAsLO	誤り	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。
行のバージョン管理	RowVersioning	オフ	[行のバージョン管理] オプションは、テーブル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 であるカーソルを作成する場合は、行のバージョン管理を使用する必要があります。
Disallow Premature	DisallowPremature	いいえ/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。はいの場合、ドライバ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。いいえの場合、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。

次のページに続く

表 1 – 前のページからの続き

Property	キーワード名	デフォルト値	説明
True is -1	TrueIsMinus1	オフ/0	TrueIsMinus1 は、TRUE の BOOL 値を-1 として返すようにドライバーに指示します。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 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。
サーバー側の準備	UseServerSidePrepare	いいえ/0	有効にすると、ドライバーは PREPARE コマンドと EXECUTE コマンドを使用して、準備/実行モデルを実装します。
GSSAPI を GSS リクエストに使用する	GssAuthUseGSS	誤り/0	True/1 に設定すると、ドライバーは GSS-API 認証要求をサーバーに送信します。Windows のみ。
Int8As	とともに	0	BI の値は、ドライバーが BIGINT 値を処理する方法を決定します。 SQL_BIGINT として-5 の場合、 SQL_NUMERIC として 2 の場合、 SQL_DOUBLE として 8 の場合、 SQL_INTEGER として 4 の場合、 SQL_VARCHAR として 12 の場合、 0 (MSJet クライアント上) の場合、 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	キーワード名	デフォルト値	説明
追加オプション	から	0x0	0x1-短い形式のフォーマットされた接続文字列を強制的に出力します。MFCC-Database 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-MSAccess 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。 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。このオプションは、Unicode データの処理が難しいアプリケーションに指定します。 0x8-設定されている場合、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、ネットワーク遅延を削減します。
接続設定	ConnSettings		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リストが含まれています。
/	ソケット	4096	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ーサイズを指定します。
/	横たわる	オフ	有効にすると、ドライバーはサポートされていない ODBC 機能をサポートすると主張します。
小文字の識別子	LowerCaseIdentifier	オフ	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	キーワード名	デフォルト値	説明
遺伝的オプティマイザを無効にする	オプティマイザ	はい/1	遺伝的クエリオプティマイザを無効にします。
キーセットを許可	更新可能なカーソル	はい/1	キーセットドリブンカーソルを許可する
SSL モード	SSLMode	無効	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。
省略された接続文字列を強制	CX	いいえ/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。
偽の MSS	FakeOidIndex	いいえ/0	MSSQLServer になりすまして、MSAccess 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 環境	BDE	いいえ/0	このオプションを有効にすると、EDB-ODBC が BorlandDatabaseEngine 準拠の出力 (Unicode に関連) に対応するように調整されます。
XA_Opt	INI_XAOPT	はい/1	有効になっている場合、SQL_TABLES の呼び出しには、ユーザーがアクセスできるテーブルのみが含まれます。

5.1 Windows でのデータソース定義の追加

WindowsODBC Data Source Administrator 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。Control Panel に移動して Administrative Tools メニューを開き、適切な ODBC Data Sources アイコン (32- or 64- bit) をダブルクリックすると、ODBC Data Source Administrator を開くことができます。

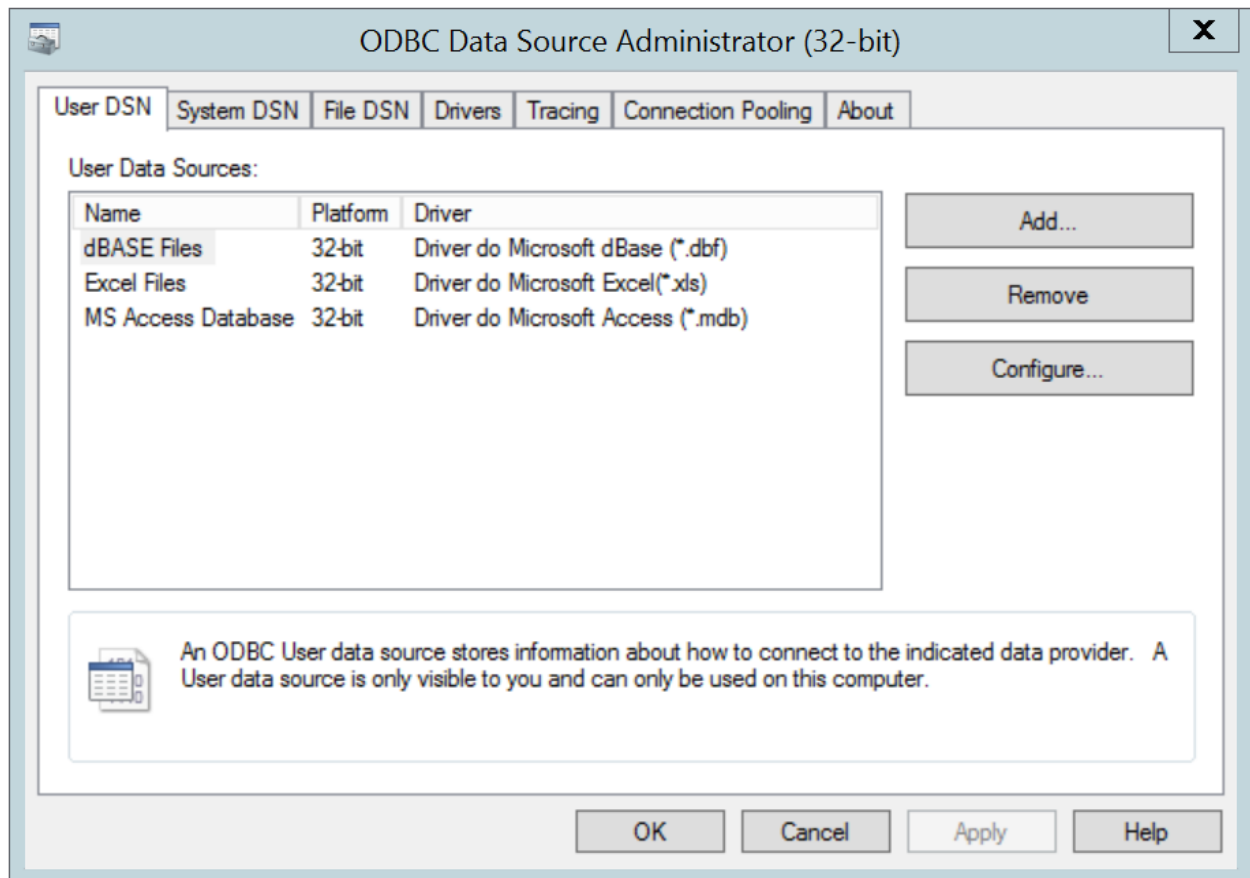


図1 Windows データソースアドミニストレーター

Add ボタンをクリックして Create New Data Source ダイアログを開きます。ドライバのリストから EnterpriseDB (ANSI) または EnterpriseDB (UNICODE) を選択し、Finish をクリックします。

EnterpriseDBODBCDriver ダイアログが開きます。

ダイアログ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。

- Database フィールドにデータベース名を入力します。
- AdvancedServer のホスト名または IP アドレスを Server フィールドに入力します。
- ユーザーの名前を User Name フィールドに入力します。
- 名前付きのデータソースのわかりやすい名前を Description フィールドに入力します。

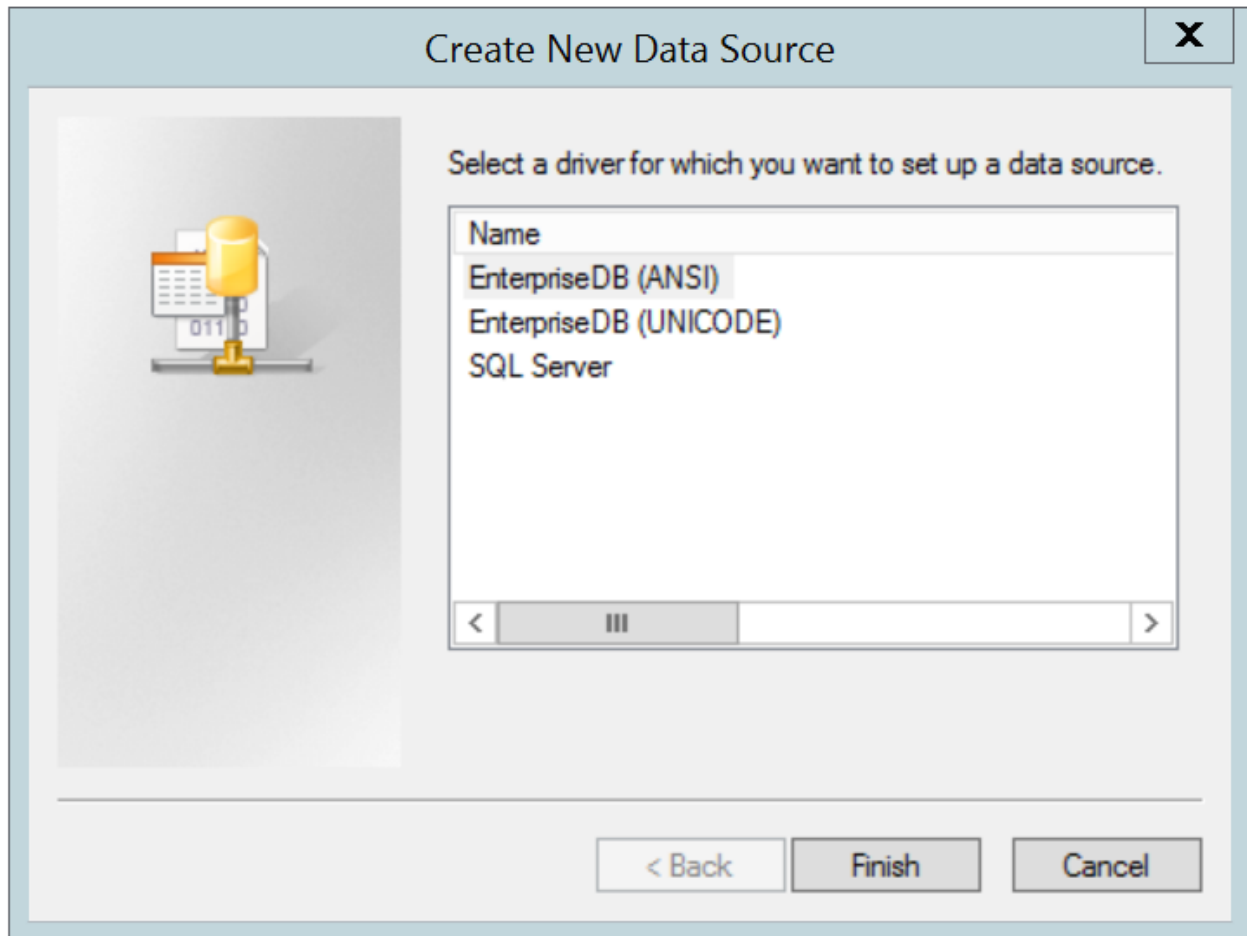


図2 CreateNewDataSource ダイアログ

- 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、SSL Mode ラベルの横にあるドロップダウンリストボックスがアクティブになり、SSL およびその他の AdvancedServer ユーティリティを使用できるようになります。
- デフォルトのポート番号 (5444) を受け入れるか、Port 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。

Options ボックスにある Datasource ボタンを使用して Advanced Options ダイアログを開き、接続プロパティを指定します。

Global ボタンをクリックすると、EDB-ODBC ドライバー（データソースではなくドライバー自体）のログオプションを指定できるダイアログが開きます。

- 遺伝的クエリオプティマイザーを無効にするには、Disable Genetic Optimizer の横のボックスをオンにします。デフォルトでは、クエリオプティマイザーは on です。
- キーセットクエリのサーバー側サポートを有効にするには、KSQO (Keyset Query Optimization)

図3 データソースを定義します

の横のボックスをオンにします。デフォルトでは、Keyset Query Optimization は on です。

- Recognize Unique Indexes の横のボックスをチェックして、SQLStatistics() 関数に一意のインデックスを報告させる; オプションがチェックされていない場合、SQLStatistics() 関数はすべてのインデックスが重複した値を許可することを報告します。デフォルトでは、Recognize Unique Indexes は on です。
- 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するには、Use Declare/Fetch の横のボックスをチェックします。デフォルトでは、Use Declare/Fetch は off です。
- CommLog (C:/psqlodbc_XXXX.log) の横にあるボックスをチェックして、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、ロギングは off です。
- Check the box next to Parse Statements to specify that the driver (rather than the server) should attempt to parse simple SELECT statements when you call the SQLNumResultCols(), SQLDescribeCol(), or SQLColAttributes() function. By default, this option is off.
- Cancel as FreeStmt (Exp) の隣にあるボックスをチェックして、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 off です。
- MyLog (C:/mylog_XXXX.log) の横のボックスをチェックして、ログファイルにドライバーアクティビティの詳細な記録を記録します。ログファイルの名前は c:/mylog/_/ *process-id*.log です。

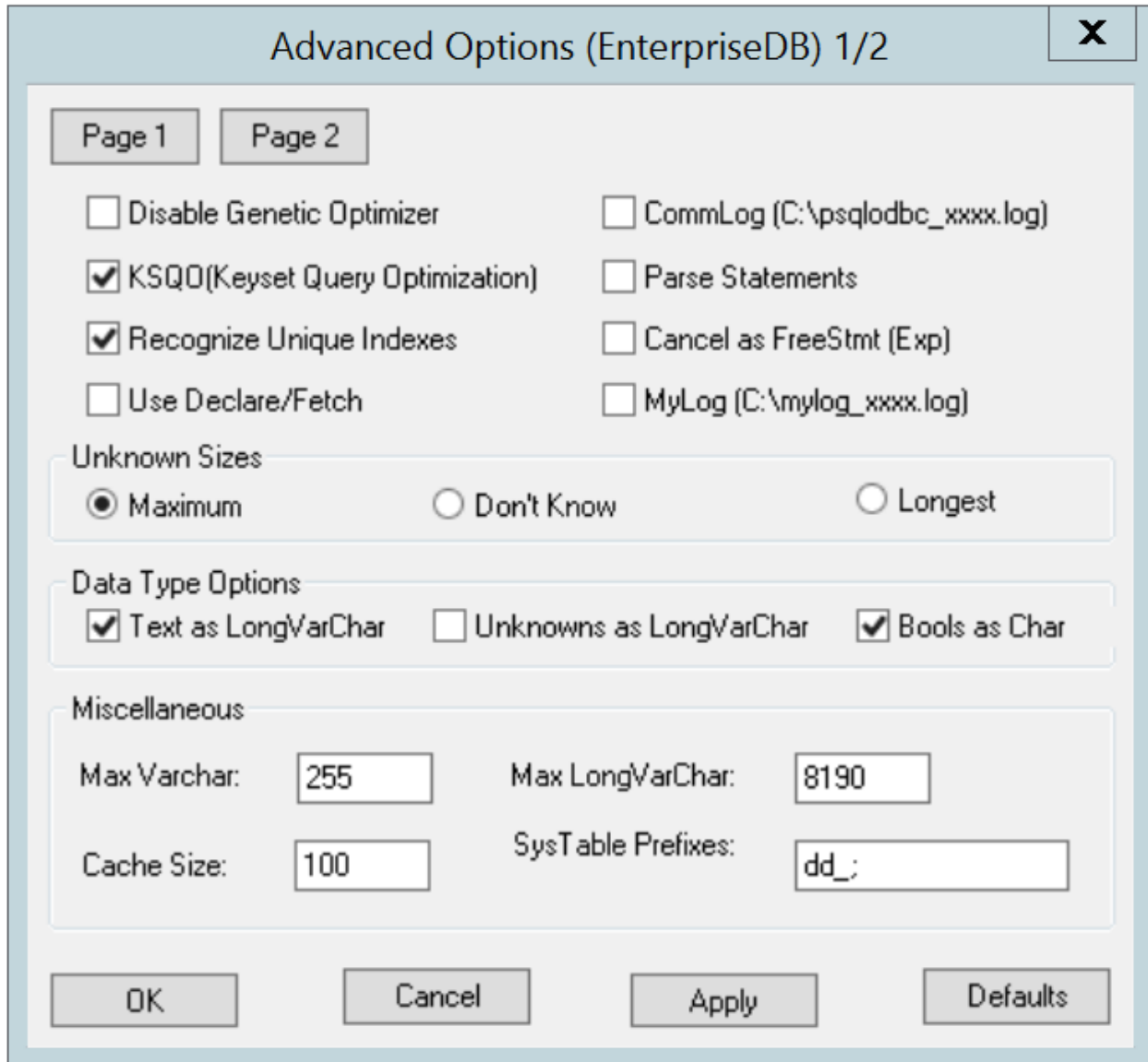


図 4 詳細オプションダイアログのページ 1

デフォルトでは、ロギングは off です。

不明なサイズボックスのラジオボタンは、`SQLDescribeCol()` および `SQLColAttributes()` 関数が未知のタイプの列のサイズを計算する方法を指定します（既知のデータタイプのリストについては、セクション `Supported Data Types` を参照してください）。

- Choose the button next to Maximum to specify that the driver report the maximum size allowed for a VARCHAR or LONGVARCHAR (dependent on the Unknowns as LongVarChar setting). If Unknowns as LongVarChar is enabled, the driver returns the maximum size of a LONGVARCHAR (specified in the Max LongVarChar field in the Miscellaneous box). If Unknowns as LongVarChar is not enabled, the driver returns the size specified in the Max Varchar field (in the Miscellaneous box).

- Choose the button next to Don't know to specify that the driver report a length of "unknown".
- ドライバーが結果セットを検索し、見つかった最も長い値を報告することを指定するには、Longest の横にあるボタンを選択します。(注: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。)

Data Type Options ボックスのプロパティは、ドライバーが特定のタイプの列を処理する方法を決定します:

- Check the box next to Text as LongVarChar to treat TEXT values as if they are of type SQL_LONGVARCHAR. If the box is not checked, the driver will treat TEXT values as SQL_VARCHAR values. By default, TEXT values are treated as SQL_LONGVARCHAR values.
- Check the box next to Unknowns as LongVarChar to specify that the driver treat values of unknown type as SQL_LONGVARCHAR values. If unchecked, the driver will treat values of unknown type as SQL_VARCHAR values. By default, values of unknown type are treated as SQL_VARCHAR values.
- Check the box next to Booleans as Char to specify that the driver treat BOOL values as SQL_CHAR values. If unchecked, BOOL values are treated as SQL_BIT values. By default, BOOL values are treated as SQL_CHAR values.

Miscellaneous ボックスのフィールドで、名前付きデータソースに関連付けられている一部のプロパティの値を指定できます。

- Max VarChar 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。
- Max LongVarChar 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。
- (UseDeclareFetch が有効な場合) ドライバーがフェッチする行数を Cache Size フィールドで指定します。デフォルト値は 100 です。
- 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するには、SysTablePrefixes フィールドを使用します。デフォルトでは、リストには dd_ ; が含まれています。

Defaults ボタンを選択すると、このダイアログの値をデフォルト設定にリセットできます。

データソースのプロパティに変更を適用するには Apply ボタンをクリックし、変更を適用せずにダイアログを終了するには Cancel ボタンをクリックします。OK ボタンを選択してダイアログに変更を適用し、終了します。

Advanced Options ダイアログの左上隅にある Page 2 ボタンを選択して、2 つ目の詳細オプションセットにアクセスします。

- ドライバーが次のコマンドを実行しないようにするには、Read Only の横のチェックボックスをオンにします:INSERT、UPDATE、DELETE、CREATE、ALTER、DROP、GRANT、REVOKE または LOCK。Read Only 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 off です。

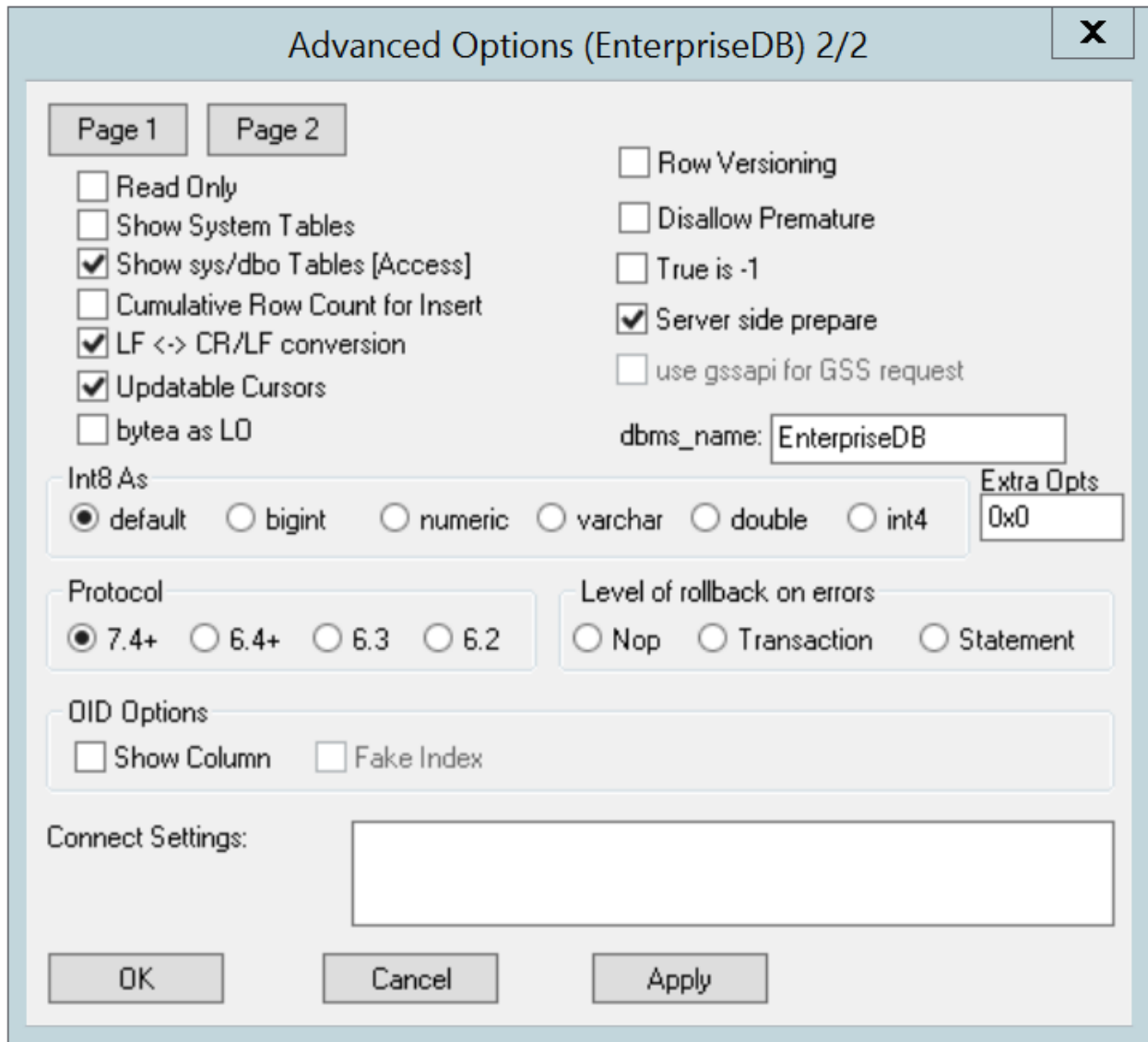


図 5 詳細オプションダイアログのページ 2

- Show System Tables の横にあるボックスをチェックして、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、ドライバーは、名前が pg/_ で始まるテーブル、または Advanced Options ダイアログの Page 1 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を含めます。デフォルトでは、このオプションは off です。
- Show sys/dbo Tables [Access] の横のボックスにチェックを入れて、ODBC データソースを介して sys スキーマと 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、このオプションは有効になっています (チェックされています)。
- Check the box next to Cumulative Row Count for Insert to cause a single, cumulative row count to be returned for the entire array of parameter settings for an INSERT statement when a call to the SQLRowCount() method is performed. If this option is not enabled (the box is not checked), then an in-

dividual row count is available for each parameter setting in the array, and thus, a call to `SQLRowCount()` returns the count for the last inserted row.

- LF<->CR/LF 変換の横のボックスをチェックして、サーバーから文字値をフェッチするときに改行文字を改行/改行ペアに変換し、改行/改行ペアを改行に変換するようドライバーに指示します-文字値をサーバーに送信するときに文字をフィードします。デフォルトでは、このオプションは有効になっています。
- Updatable Cursors の横にあるボックスをチェックして、ドライバーが `SQLSetPos()` または `SQLBulkOperations()` 機能で UPDATE および DELETE の位置決め操作を許可する必要があることを指定します。デフォルトでは、このオプションは有効になっています。
- Check the box next to `bytea as LO` to specify that the driver should treat BYTEA values as if they are `SQL_LONGVARBINARY` values. If the box is not checked, EDB-ODBC will treat BYTEA values as if they are `SQL_VARBINARY` values. By default, BYTEA values are treated as `SQL_VARBINARY` values.
- テーブルの列を報告するときに xmin 列を含めるには、Row Versioning の横のボックスをオンにします。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` のカーソルを作成する場合は、行の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、Row Versioning は off です。
- Check the box next to `Disallow Premature` to specify that the driver should retrieve meta-data about a query (i.e., the number of columns in a result set, or the column types) without actually executing the query. If this option is not specified, the driver executes the query when you request meta-data about the query. By default, `Disallow Premature` is off.
- 0 の横のボックスをチェックして、True の 1 の値を -1 として返すようにドライバーに指示します。このオプションが有効でない場合、ドライバは True の 1 値を 1 として返します。ドライバは常に False の 1 値を 0 として返します。
- Server side prepare の横にあるボックスをチェックして、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ドライバーに指示します。デフォルトでは、このボックスはチェックされています。
- use gssapi for GSS request の横のボックスをチェックして、サーバーに GSSAPI 接続要求を送信するようドライバーに指示します。
- Enter the database system (either EnterpriseDB or PostgreSQL) in the `dbms_name` field. The value entered here is returned in the `SQL_DBMS_NAME` argument when the `SQLGetInfo()` function is called. The default is EnterpriseDB.

Int8 As ボックスのラジオボタンを使用して、ドライバーがクライアントに BIGINT 値を返す方法を指定します。default の横にあるラジオボタンを選択して、クライアントが MSJet の場合は NUMERIC、クライアントが他の ODBC クライアントの場合は BIGINT のデフォルトタイプを指定します。オプションで、ドライバーが BIGINT の値を 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。

The default value of the `Extra Opts` field is 0x0. `Extra Opts` may be:

オプション	指定する
0x1	短い形式の接続文字列を強制的に出力します。MFCCDatabase 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	MSAccess 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。
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 データの処理が困難なアプリケーションには、このオプションを選択してください。
0x8	設定されている場合、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 コマンドはできるだけ少ないラウンドトリップにグループ化され、ネットワーク遅延を削減します。

Protocol 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ラジオボタンが含まれています。デフォルトでは、選択された Protocol 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。

Level of Rollback on errors 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ラジオボタンが含まれています:

オプション	指定する
トランザクション	ドライバでエラーが発生すると、現在のトランザクションがロールバックされます。
ステートメント	ドライバでエラーが発生すると、現在のステートメントがロールバックされます。
いいえ	ドライバでエラーが発生した場合は、アプリケーションを続行する前に、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。

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

- Check the box next to `Show Column` to include the OID column in the result set of the `SQLColumns()` function. If this box is not checked, the OID column is hidden from `SQLColumns()`.
- Fake Columns の隣のボックスをチェックして、`SQLStatistics()` 関数が各 OID 列に一意のインデックスが存在することを報告することを指定します。

Connect Settings フィールドを使用して、この接続を開くときにドライバーが使用する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設定パラメータは、セミコロンで区切られたリスト

に含めることができます。例えば:

```
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ら、Apply ボタンをクリックしてオプションを適用します。オプションで Cancel を選択すると、オプションを保存せずに終了できます。オプションを保存して終了するには、OK ボタンを選択します。

(EnterpriseDB ODBC Driver ダイアログの) Global ボタンを選択して Global Settings ダイアログを開きます。このダイアログのオプションは、EDB-ODBC ドライバーのロギングオプションを制御します。このダイアログを使用して、名前付きデータソースなしでドライバを使用する場合、または接続文字列が解析される前に発生するドライバ操作をログに記録する場合に、ログを適用します。

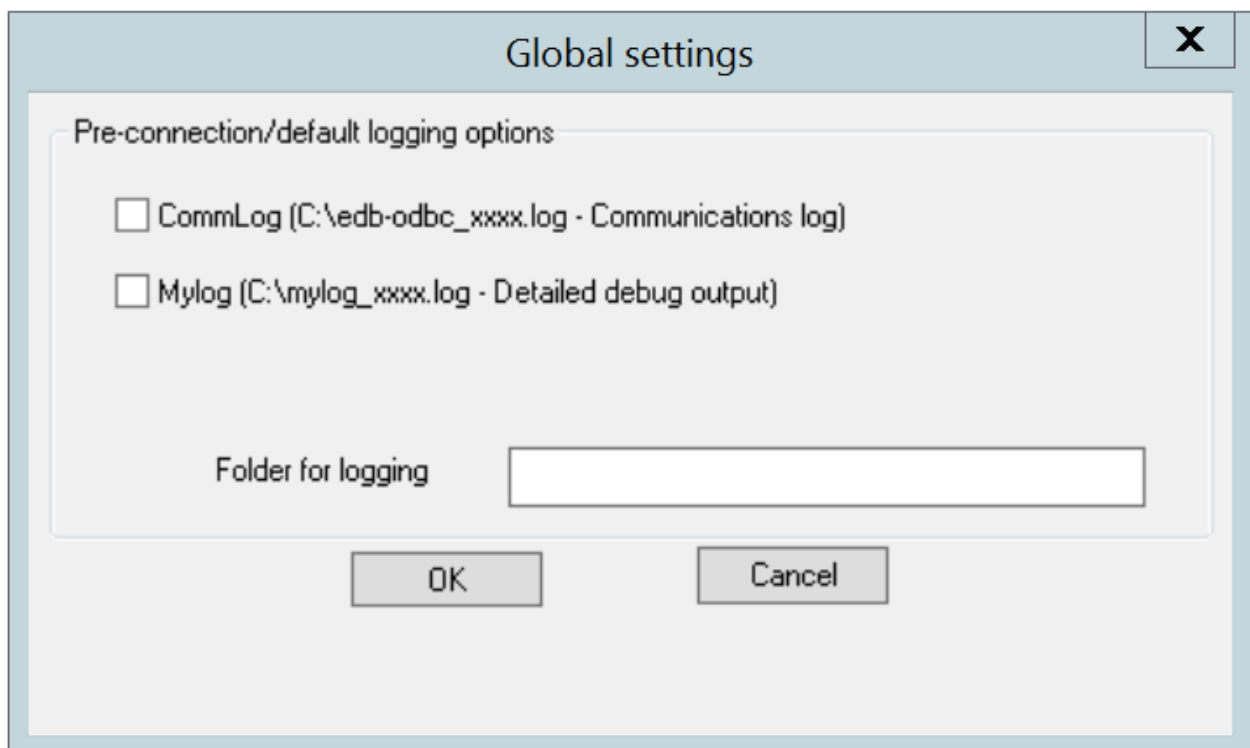


図 6 グローバル設定ダイアログ

- CommLog フィールドの横にあるチェックボックスをオンにして、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、process-id は使用中のプロセスの名前です。
- ドライバーのアクティビティのログファイルを保持するには、Mylog フィールドの横にあるボックスをチェックします。ログファイルの名前は c:/mylog_process-id で、process-id は使用中のプロセスの名前です。
- Folder for logging フィールドで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ら、Test ボタンをクリックして、ドライバマネージャが定義済

みのデータソースに接続できることを確認します。

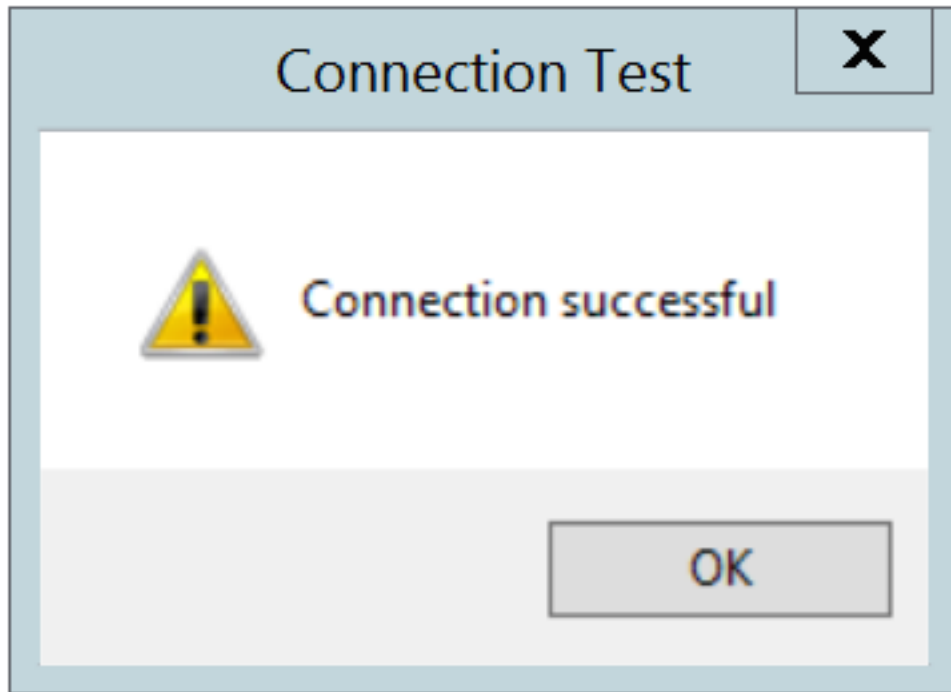


図 7 接続に成功しました

OK ボタンをクリックして Connection Test ダイアログを終了します。接続が成功したら、Save ボタンをクリックして名前付き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、もう一度テストしてください。

5.2 Linux でのデータソース定義の追加

The Linux ODBC Administrator is a graphical tool that is distributed with unixODBC; you can use the ODBC Administrator to manage ODBC drivers and named resources. To add the ODBC Administrator to your system, open a terminal window, assume superuser privileges, and enter:

```
yum install unixODBC
```

に続く:

```
yum install unixODBC-kde
```

ODBC Administrator を呼び出すには、ターミナルウィンドウを開き、ODBCConfig と入力します。



図 8 unixODBC データソースアドミニストレーター

AdvancedServer Connectors コンポーネントをインストールすると、ODBC アドミニストレーターのドライバーのリストに EDB-ODBC ドライバーが追加されます。Advanced をクリックし、Drivers タブを選択して、enterprisedb ドライバがリストに表示されることを確認します。

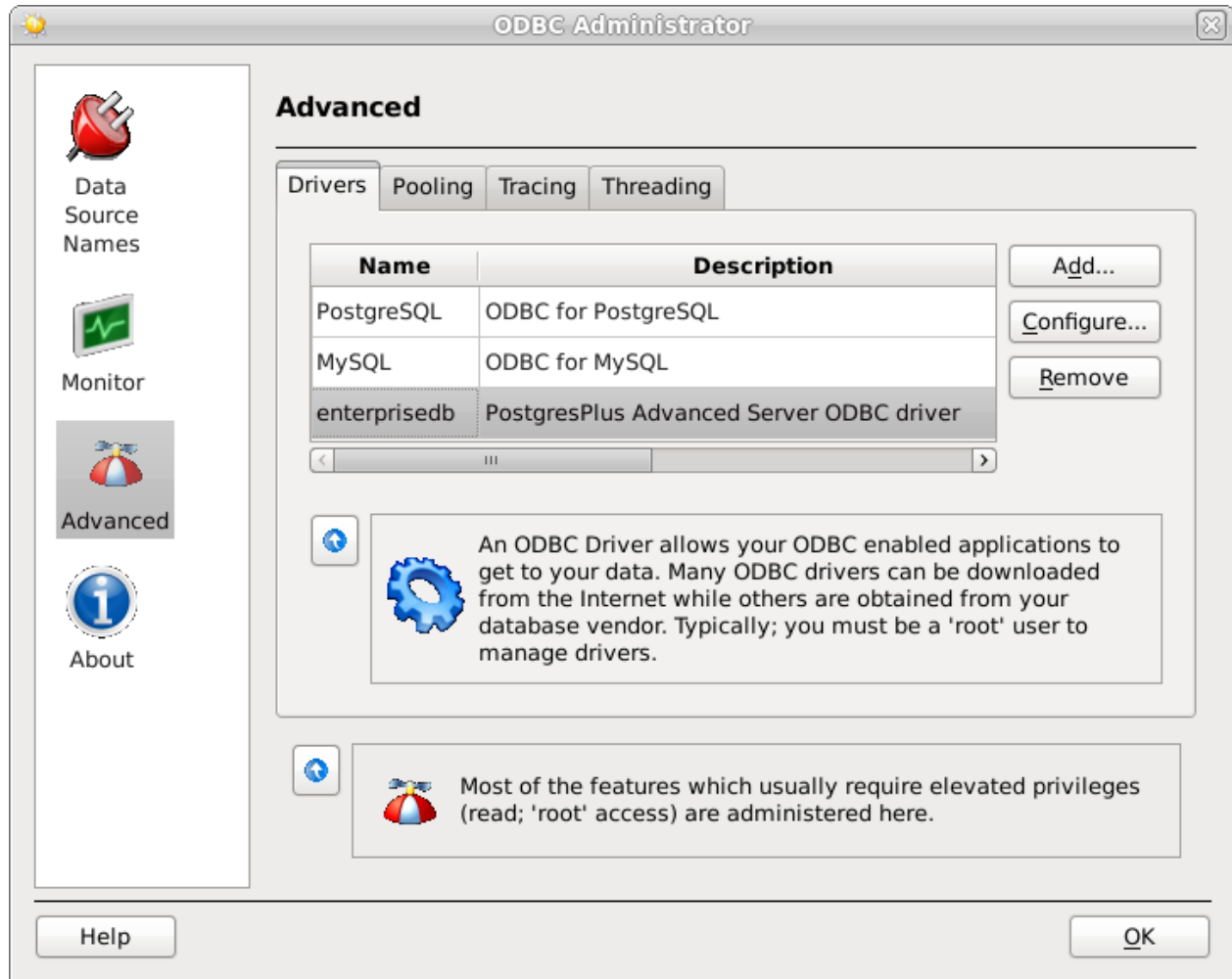


図9 [ドライバー] タブには、インストールされている EDB-ODBC ドライバーが表示されます

EDB-ODBC ドライバーがドライバーのリストに表示されない場合は、ODBC Administrator を使用して追加できます。ドライバー定義を追加するには、Drivers タブを選択し、Add をクリックします。以下に示すように、Driver Properties (new) ウィンドウが開きます:

Driver Properties ウィンドウに入力して、EDB-ODBC ドライバーをドライバーマネージャーに登録します:

- ドライバーの一意の名前を Name フィールドに追加します。
- ドライバの説明を Description フィールドに追加します。
- EDB-ODBC ドライバーの場所へのパスを Driver フィールドに追加します。デフォルトでは、ドライバーへの完全なパスは次のとおりです。

/usr/edb/odbc/lib/edb-odbc.so

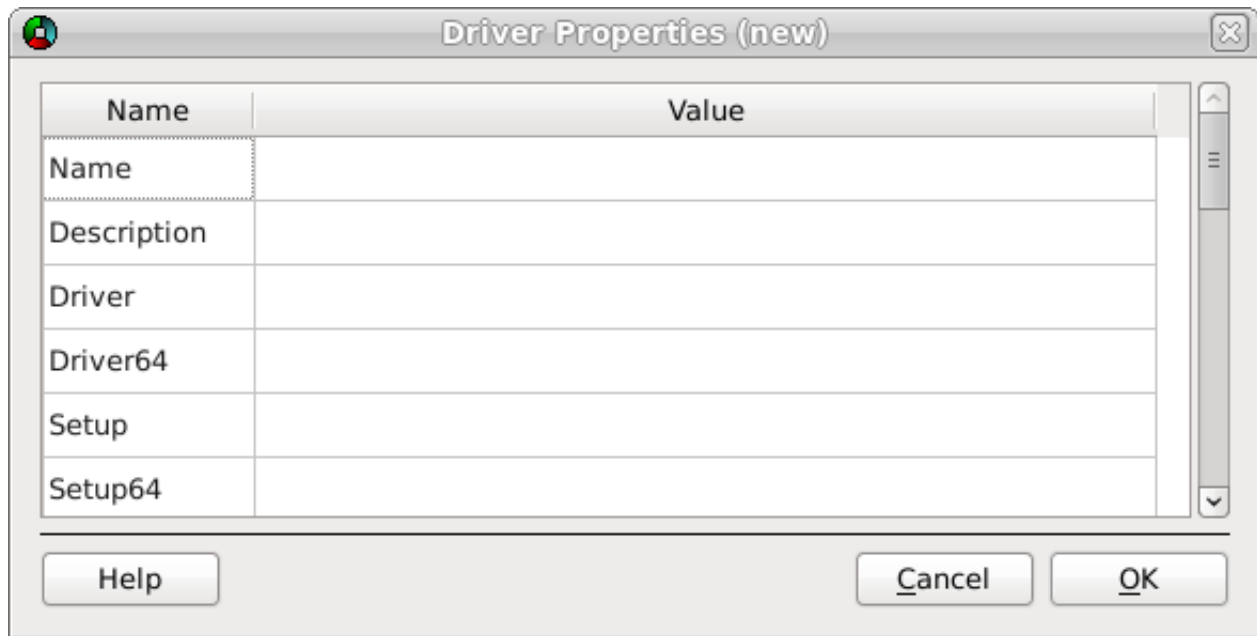


図 10 [ドライバーのプロパティ] ウィンドウ

- EDB-ODBC ドライバセットアップファイルの場所へのパスを Setup フィールドに追加します。デフォルトでは、ドライバセットアップファイルへの完全なパスは次のとおりです。

```
/usr/edb/odbc/lib/libodbcedbS.so
```

EDB-ODBC ドライバのドライバープロパティを記述したら、OK をクリックします。[ODBC データソースアドミニストレータ] ウィンドウには、使用可能な ODBC ドライバのリストに EDB-ODBC ドライバが含まれています。

ドライバーマネージャーで使用する EDB-ODBC ドライバを使用して、データソースを追加できます。左側のパネルで Data Source の名前オプションをクリックし、追加するデータソース名のタイプに応じて適切な DSN タブを選択します。

- User タブを選択して、現在のユーザーのみが利用できる名前付きデータソースを追加します（データソースは /user/.odbc.ini に格納されます）。
- System タブを選択して、すべてのユーザーが利用できる名前付きデータソースを追加します。すべてのシステムデータソースは 1 つのファイル（通常は /etc/odbc.ini）に保存されます。
- File タブを選択して、すべてのユーザーが利用できる名前付きデータソースを追加しますが、これは選択したファイルに保存されます。

適切なタブを選択し、Add をクリックします。以下に示すように、Create a New Data Source... ウィンドウが開きます：

リストから EDB-ODBC ドライバを選択し、OK をクリックして Data Source Properties ウィンドウを開きます。

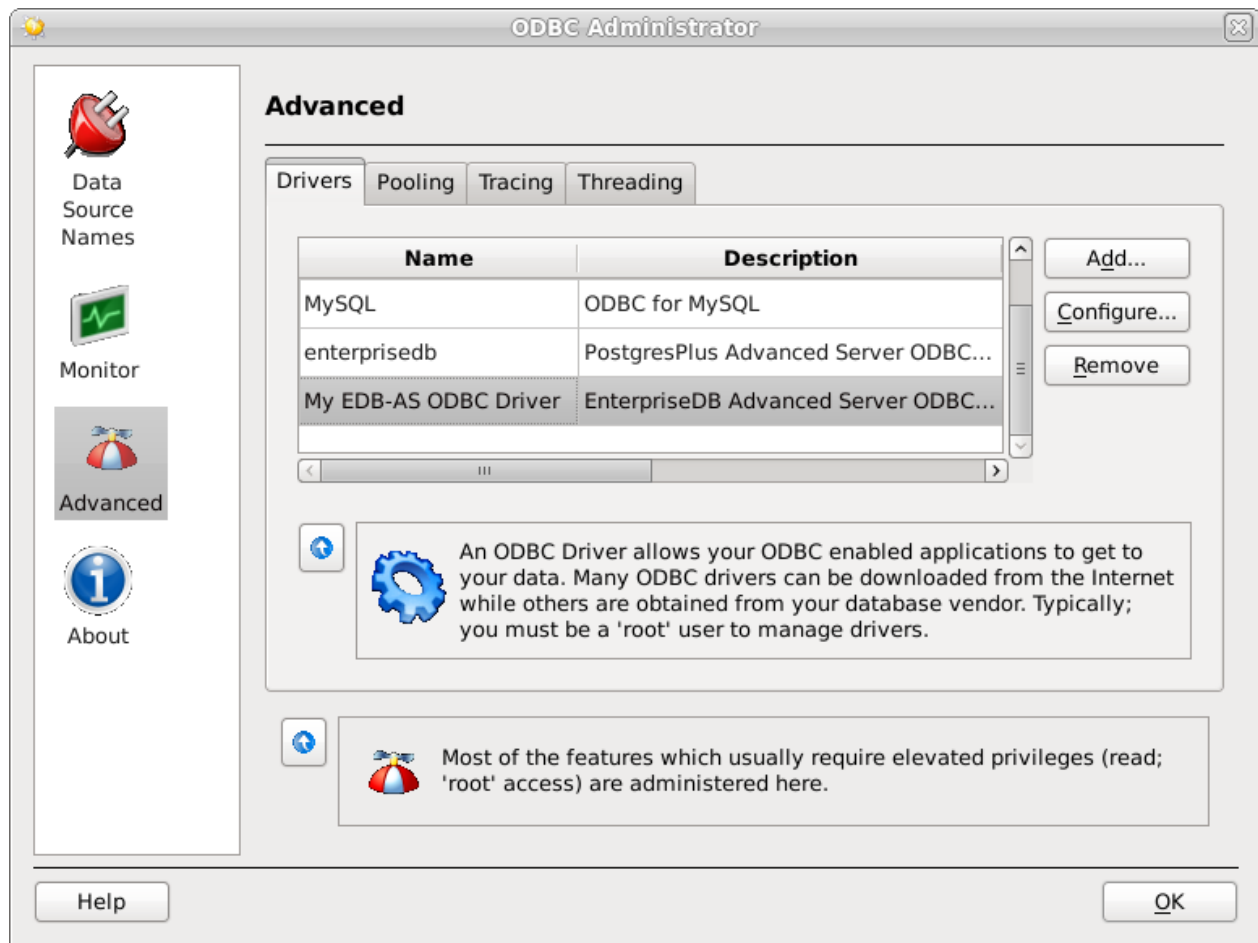


図 11 [ドライバー] タブには新しいドライバーの定義が表示されます

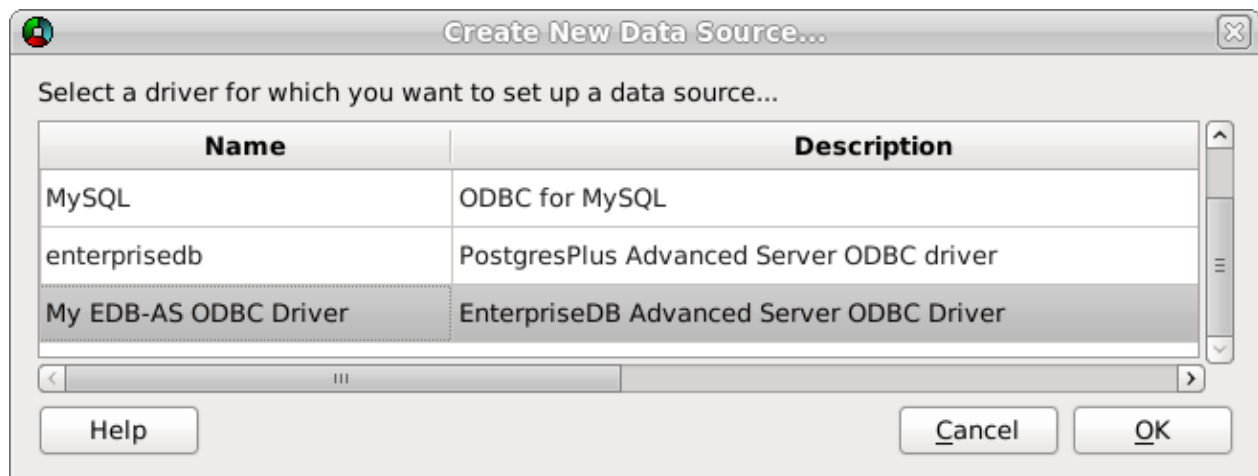
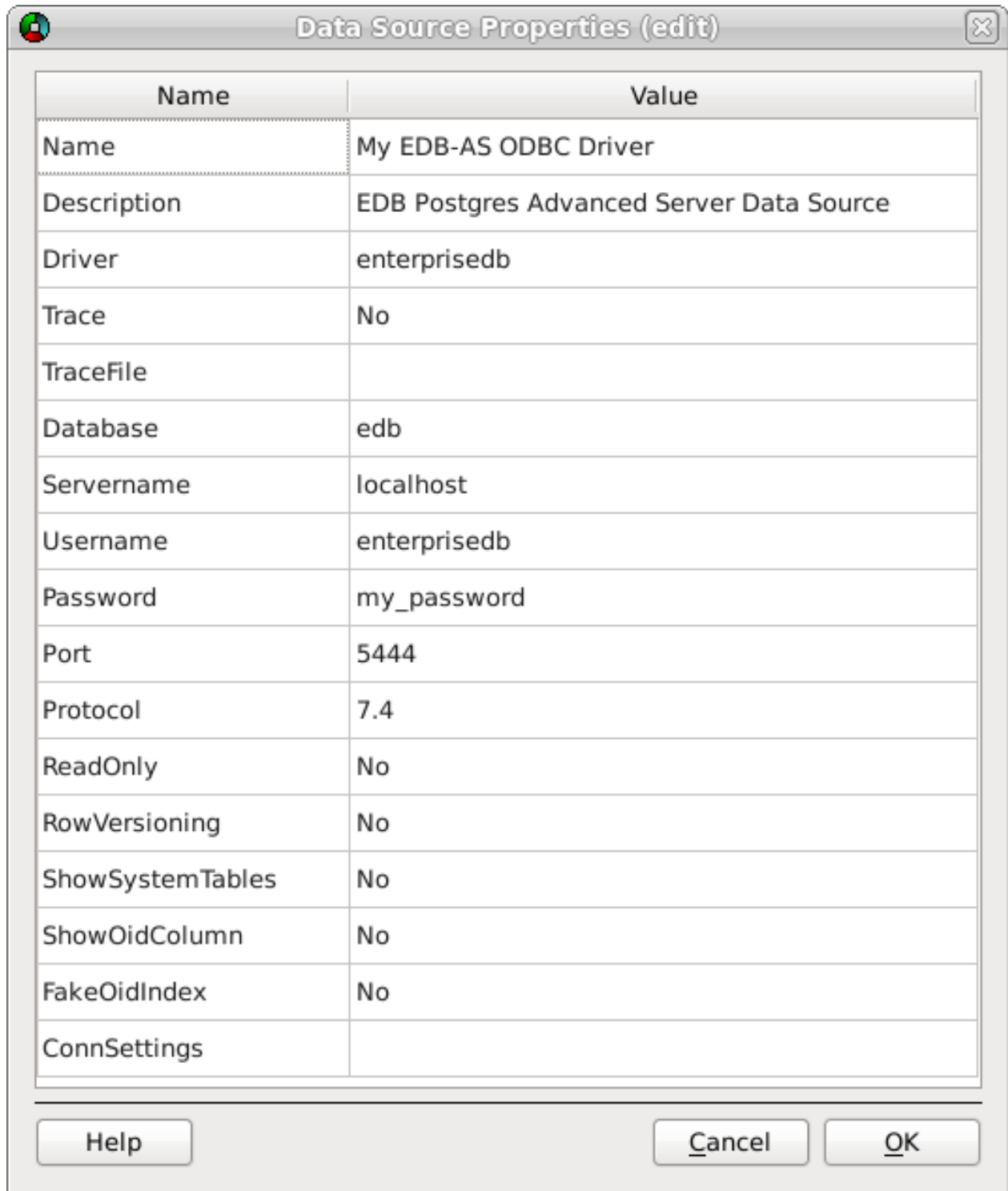


図 12 名前付きデータソースのドライバーを選択してください

EDB-ODBC ドライバーの接続プロパティを指定して、Data Source Properties (new) ウィンドウに入力します。

- Name フィールドにデータソース名を入力します。
- 名前の付いたデータソースの説明を Description フィールドに入力します。
- unixODBC ドライバーには、ODBC アプリケーションが行った一連の呼び出しをログファイルに記録するトレースユーティリティが含まれています。Trace フィールドに Yes を指定して、トレースユーティリティをオンにします。トレース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可能性があることに注意してください。
- TraceFile フィールドを使用して、Trace ユーティリティから返された情報を受け取るファイルを指定します。
- AdvancedServer データベースの名前を Database フィールドに入力します。
- AdvancedServer のホスト名または IP アドレスを Servername フィールドに入力します。
- ユーザーの名前を Username フィールドに入力します。
- Password フィールドにユーザーのパスワードを入力します。
- Port フィールドにポート番号を入力します（またはデフォルト値の 5444 を受け入れます）。
- Protocol フィールドを使用して、フロントエンド/バックエンドのプロトコルバージョンを指定します。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 から選択できます。
- ドライバーが次のコマンドを実行しないようにするには、ReadOnly フィールドを使用して Yes を指定します: INSERT、UPDATE、DELETE、CREATE、ALTER、DROP、GRANT、REVOKE または LOCK。Read Only オプションを有効にすると、ODBC 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、ReadOnly は No に設定されています。
- テーブルの列を報告するときにドライバーが xmin 列を含める必要がある場合は、RowVersioning フィールドを使用して Yes を指定します。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 のカーソルを作成する場合は、行の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、Row Versioning は No に設定されています。
- ドライバーが SQLTables() 関数の結果セットにシステムテーブルを含める必要がある場合は、ShowSystemTables フィールドを使用して Yes を指定します。デフォルトでは、このフィールドは No に設定されています。
- Use the ShowOidColumn field to specify Yes if the driver should include the OID column in the result set of the SQLColumns() function. If ShowOidColumn is set to No, the OID column is hidden from SQLColumns(). By default, this option is set to No.



The image shows a Windows-style dialog box titled "Data Source Properties (edit)". It contains a table with two columns: "Name" and "Value". The table lists various properties for an ODBC data source. At the bottom of the dialog, there are three buttons: "Help", "Cancel", and "OK".

Name	Value
Name	My EDB-AS ODBC Driver
Description	EDB Postgres Advanced Server Data Source
Driver	enterprisedb
Trace	No
TraceFile	
Database	edb
Servename	localhost
Username	enterprisedb
Password	my_password
Port	5444
Protocol	7.4
ReadOnly	No
RowVersioning	No
ShowSystemTables	No
ShowOidColumn	No
FakeOidIndex	No
ConnSettings	

Help Cancel OK

図 13 データソースのプロパティウィンドウ

- `SQLStatistics()` 関数が各 OID 列に一意のインデックスが存在することを報告する必要がある場合は、`FakeOidIndex` フィールドを使用して **Yes** を指定します。これは、アプリケーションに一意の識別子が必要であり、テーブルに一意の識別子が含まれていない場合に役立ちます。デフォルト値は **No** です。
- `ConnSettings` フィールドを使用して、この接続を開くときにドライバーが使用するパラメーター割り当てのリストを指定します。

接続プロパティを定義したら、OK をクリックします。

新しいデータソースがデータソース名のリストに追加されます。

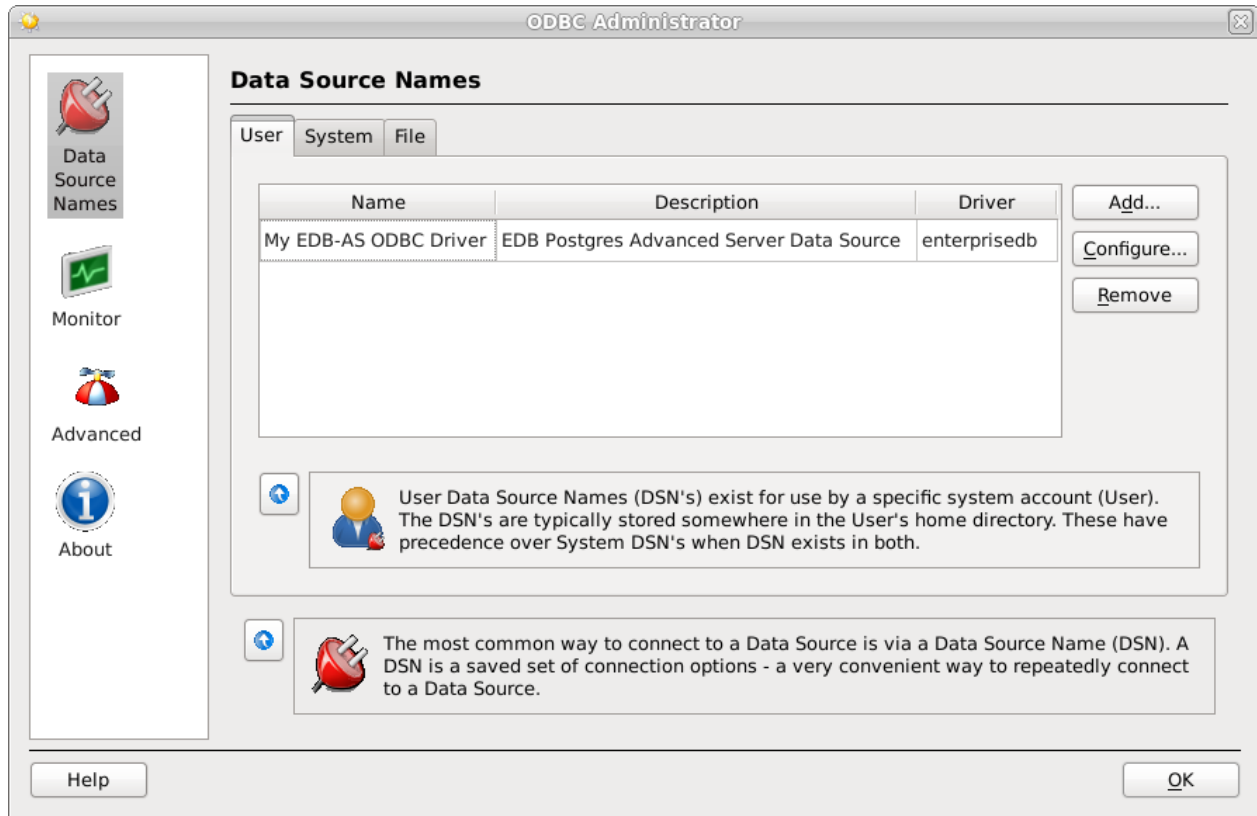


図 14 新しいデータソースがデータソース名リストに含まれています

第 6 章

EDB-ODBC ドライバーの機能

ODBC 関数を使用して、ODBC に 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を照会できます。

- `SQLGetInfo()` は EDB-ODBC ドライバーと `AdvancedServer` 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 とのインターフェースに使用するオブジェクトのさまざまな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属性を設定します。

6.1 SQLGetInfo ()

The ODBC `SQLGetInfo()` function returns information about the EDB-ODBC driver and Advanced Server. You must have an open connection to call `SQLGetInfo()`, unless you specify `SQL_ODBC_VER` as the `info_type`. The signature for `SQLGetInfo()` is:

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

```

(次のページに続く)

(前のページからの続き)

```
SQLSMALLINT \*\ *string_length_pointer* // Output  
);
```

conn_handle

接続ハンドル。

info_type

SQLGetInfo () が取得する情報のタイプ。

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。

引数 info_type が SQL_DRIVER_HDESC または SQL_DRIVER_HSTMT の場合、info_pointer 引数は Input と Output の両方になります。

buffer_len

buffer_len is the length of the allocated memory buffer pointed to by info_pointer. If info_pointer is NULL, buffer_len is ignored. If the returned value is a fixed size, buffer_len is ignored. buffer_len is only used if the requested value is returned in the form of a character string.

string_length_pointer

string_length_pointer は SQLSMALLINT 値へのポインタです。SQLGetInfo () は要求された値のサイズをこの整数に書き込みます。

A typical usage is to call SQLGetInfo () with a NULL info_pointer to obtain the length of the requested value, allocate the required number of bytes, and then call SQLGetInfo () again (providing the address of the newly allocated buffer) to obtain the actual value. The first call retrieves the number of bytes required to hold the value; the second call retrieves the value.

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 は無視されます（要求された値のサイズは SQLGetInfo () では提供されません）。

SQLGetInfo () は次のいずれかの形式で情報を書き込みます：

- SQLUIINTEGER ビットマスク
- SQLUIINTEGER フラグ
- SQLUIINTEGER バイナリ値
- SQLUSMALLINT の値
- NULL で終了する文字列

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、AdvancedServer 接続に関して EDB-ODBC から返される情報を示しています。

表 1: EDB_ODBC

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_ACCESSIBLE_PROCEDURES:SQLProcedures () から返されたプロシージャをアプリケーションで 実行できるかどうかを示します。	N を返します。SQLProcedures () 関数によって実行される いくつかのプロシージャは、アプリケーションによって 実行される場合があります。
SQL_ACCESSIBLE_TABLES:SQLTables () によって 返されたすべてのテーブル名に対する SELECT 権限 がユーザーにあるかどうかを示します。	N を返します。ユーザーは、SQLTables () 関数によって戻 された 1 つ以上の表に対する選択特権を持っていない可能 性があります。
SQL_ACTIVE_CONNECTIONSprev. SQL_MAX_DRIVER_CONNECTIONS:EDB-ODBC がサポートできる接続の最大数を示します。	0 を返します。許可される接続数に制限はありません。
SQL_ACTIVE_ENVIRONMENTS:EDB-ODBC がサ ポートできるアクティブな環境の数。	0 を返します。許可される環境の数に制限はありません。
SQL_ACTIVE_STATEMENTS 前 へ SQL_MAX_CONCURRENT_ACTIVITIES:EDB- ODBC がサポートできるアクティブなステートメン トの最大数を示します。	0 を返します。許可されるアクティブなステートメントの 数に制限はありません。
SQL_AGGREGATE_FUNCTION:サーバーおよびド ライバーでサポートされる集約関数を識別します。	SQL_AF_ALL を返します
SQL_ALTER_DOMAIN:サーバーでサポートされる ALTERDOMAIN 句を識別します。	0 を返します。ALTERDOMAIN 句はサポートされていま せん。
SQL_ALTER_TABLE:サーバーがサポートする AL TERTABLE 句を識別します。	SQL_AT_ADD_COLUMN、SQL_AT_DROP_TABLE_CONSTRAINT_C SQL_AT_DROP_TABLE_CONSTRAINT、 SQL_AT_CONSTRAINT_INITIALLY_DEFERRED、 SQL_AT_CONSTRAINT_INITIALLY_IMMEDIATE、 SQL_AT_CONSTRAINT_DEFERRABLE を返します
SQL_ASYNC_MODE:EDB-ODBC でサポートされる 非同期モードのレベル。	SQL_AM_NONE を返します。非同期モードはサポートさ れていません。
SQL_BATCH_ROW_COUNT:ドライバーが行カウ ントを返す方法を示します。	SQL_BRC_EXPLICIT を返します。行カウントは、SQLEx ecute または SQLExecDirect を呼び出して実行するとき に使用できます。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_BATCH_SUPPORT:バッチステートメント実行のサポートを示します。	戻り値:SQL_BS_SELECT_EXPLICIT、SQL_BS_ROW_COUNT_EXPLICIT。ドライバは、結果セットと行数を生成するステートメントを含む明示的なバッチをサポートします。
SQL_BOOKMARK_PERSISTENCE:ブックマークのサポートのレベルを示します。	戻り値:SQL_BP_DELETE、SQL_BP_TRANSACTION、SQL_BP_UPDATE、SQL_BP_SCROLL。
SQL_CATALOG_LOCATIONS:修飾されたテーブル名でのカタログの位置を示します。	SQL_CL_START を返します。修飾テーブル名のカタログ部分は、名前の最初にあります。
SQL_CATALOG_NAME が SQL_QUALIFIER_NAME になりました:カタログ名のサポートを示します。	Y を返します。サーバーはカタログ名をサポートしています。
SQL_CATALOG_NAME_SEPARATOR:隣接する名前要素からカタログ名を区切る文字。	戻り値:SQL_CS_SEPARATOR。サーバーは ``.`` を期待します修飾子とテーブル名の間の文字。/“;”
SQL_CATALOG_TERM が SQL_QUALIFIER_TERM になりました:カタログの説明に使用される用語。	カタログを返します。
SQL_CATALOG_USAGES:カタログを参照する可能性のある SQL ステートメントを示します。	SQL_CU_DML_STATEMENTS を返します。カタログ名は、SELECT、INSERT、UPDATE、DELETE、SELECT-FORUPDATE、および位置付け UPDATE および DELETE ステートメントで使用できます。
SQL_COLLATION_SEQ:照合順序の名前を返します。	空の文字列を返します。デフォルトの照合の名前は不明です。
SQL_COLUMN_ALIAS:サーバーが列の別名をサポートしていることを示します。	Y を返します。サーバーは列のエイリアスをサポートしています。
SQL_CONCAT_NULL_BEHAVIOR:サーバーが NULL 値の連結を処理する方法を示します。	SQL_CB_NON_NULL を返します。NULL 値と非 NULL 値を連結すると、NULL 値になります。
SQL_CONVERT_BIGINT:CONVERT 関数を使用した BIGINT タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_BINARY:CONVERT 関数を使用した BINARY タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_BIT:CONVERT 関数を使用した BIT タイプからの変換サポートを示します。	戻り値:SQL_CVT_INTEGER、SQL_CVT_BIT。
SQL_CONVERT_CHAR:CONVERT 関数を使用した CHAR タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_DATE:CONVERT 関数を使用した DATE タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_CONVERT_DECIMAL: CONVERT 関数を使用した DECIMAL タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_DOUBLE: CONVERT 関数を使用した DOUBLE タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_FLOAT: CONVERT 関数を使用した FLOAT タイプからの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_FUNCTIONS: CONVERT 関数を使用してサーバーとドライバーがサポートするスカラー変換関数をリストします。	戻り値: SQL_FN_CVT_CONVERT。
SQL_CONVERT_INTEGER: CONVERT 関数を使用した INTEGER タイプからの変換サポートをリストします。	戻り値: SQL_CVT_INTEGER、SQL_CVT_BIT。
SQL_CONVERT_INTERVAL_DAY_TIME: CONVERT 関数を使用した INTERVAL_DAY_TIME タイプからの変換サポートを示します。	この info_type は現在サポートされていません。
SQL_CONVERT_INTERVAL_YEAR_MONTH: CONVERT 関数を使用した INTERVAL_YEAR_MONTH タイプからの変換サポートを示します。	この info_type は現在サポートされていません。
SQL_CONVERT_LONGVARBINARY: CONVERT 関数を使用した LONG_VARBINARY タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_LONGVARCHAR: CONVERT 関数を使用した LONGVARCHAR タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_NUMERIC: CONVERT 関数を使用した NUMERIC タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_REAL: CONVERT 関数を使用した REAL タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_SMALLINT: CONVERT 関数を使用した SMALLINT タイプの変換サポートを示します。	戻り値: SQL_CVT_INTEGER、SQL_CVT_BIT。
SQL_CONVERT_TIME: CONVERT 関数を使用した TIME タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CVT_TIMESTAMP: CONVERT 関数を使用した TIMESTAMP タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_TINYINT: CONVERT 関数を使用した TINYINT タイプの変換サポートを示します。	戻り値: SQL_CVT_INTEGER、SQL_CVT_BIT。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_CONVERT_VARBINARY: CONVERT 関数を使用した VARBINARY タイプの変換サポートを示します。	0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_VARCHAR: CONVERT 関数を使用した VARCHAR タイプの変換サポートを示します。	戻り値: SQL_CVT_INTEGER、SQL_CVT_BIT。
SQL_CONVERT_WCHAR: CONVERT 関数を使用した WCHAR タイプの変換サポートを示します。	この info_type は、Unicode ドライバーを使用する場合にのみ有効です。0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_WLONGVARCHAR: CONVERT 関数を使用した WLONGVARCHAR タイプの変換サポートを示します。	この info_type は、Unicode ドライバーを使用する場合にのみ有効です。0 を返します。サーバーは変換をサポートしていません。
SQL_CONVERT_WVARCHAR: CONVERT 関数を使用した WVARCHAR タイプの変換サポートを示します。	この info_type は、Unicode ドライバーを使用する場合にのみ有効です。0 を返します。サーバーは変換をサポートしていません。
SQL_CORRELATION_NAME: サーバーが相関名をサポートしていることを示します。	SQL_CN_ANY を返します。相関名がサポートされており、任意の有効な名前にすることができます。
SQL_CREATE_ASSERTION: CREATEASSERTION ステートメントのサポートを示します。	0 を返します。CREATEASSERTION ステートメントはサポートされていません。
SQL_CREATE_CHARACTER_SET: CREATECHARACTER SET ステートメントのサポートを示します。	0 を返します。CREATECHARACTER ステートメントはサポートされていません。
SQL_CREATE_COLLATION: CREATECOLLATION のサポートを示します。	0 を返します。CREATECOLLATION ステートメントはサポートされていません。
SQL_CREATE_DOMAIN: CREATEDOMAIN ステートメントのサポートを示します。	0 を返します。CREATEDOMAIN ステートメントはサポートされていません。
SQL_CREATE_SCHEMA: CREATESHEMA ステートメントのサポートを示します。	戻り値: SQL_CS_CREATE_SCHEMA、SQL_CS_AUTHORIZATION。
SQL_CREATE_TABLE: CREATETABLE ステートメントのサポートを示します。	戻り値: SQL_CT_CREATE_TABLE、SQL_CT_GLOBAL_TEMPORARY、SQL_CT_CONSTRAINT_INITIALLY_DEFERRED、SQL_CT_CONSTRAINT_INITIALLY_IMMEDIATE、SQL_CT_CONSTRAINT_DEFERRABLE、SQL_CT_COLUMN_CONSTRAINT、SQL_CT_COLUMN_DEFTRAT、SQL_CT_COLUMN_DEFTRAT、SQL_CT_CONTRA_STRAINT
SQL_CREATE_TRANSLATION: CREATETRANSLATION ステートメントのサポートを示します。	0 を返します。CREATETRANSLATION ステートメントはサポートされていません。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_CREATE_VIEW:CREATEVIEW ステートメントのサポートを示します。	SQL_CV_CREATE_VIEW を返します。
SQL_CURSOR_COMMIT_BEHAVIOR:COMMIT 操作がカーソルに与える影響を示します。	SQL_CB_PRESERVE を返します。カーソルは変更されず、引き続きデータをフェッチできます。
SQL_CURSOR_ROLLBACK_BEHAVIOR:ROLLBACK 操作後のサーバーの動作を示します。	SQL_CB_PRESERVE を返します。カーソルは変更されず、引き続きデータをフェッチできます。
SQL_CURSOR_SENSITIVITY:サーバーが変更の結果セットに同期する方法を示します。	この info_type は現在サポートされていません。
SQL_DATA_SOURCE_NAME:接続中に使用されたサーバー名を返します。	返される値は、接続プロパティによって決まります。
SQL_DATA_SOURCE_READ_ONLY:接続が読み取り専用モードであるかどうかを示します。	返される値は、接続プロパティによって決まります。
SQL_DATABASE_NAME:データベースの名前を返します。	返される値は、接続プロパティによって決まります。
SQL_DATETIME_LITERALS:サーバーがサポートする DATETIMELITERALS を示します。	この info_type はサポートされていません。
SQL_DBMS_NAME:DBMS システムの名前を返します。	Linux の odbc.ini ファイルの dbms_name パラメータで指定された値、または Windows でデータソースを定義するときに [詳細オプション] ダイアログボックスの 2 ページの dbms_name フィールドを返します。デフォルトは EnterpriseDB です。
SQL_DBMS_VER:サーバーのバージョンを返します。	サーバーによって決定されます。
SQL_DDL_INDEX:インデックスの作成と削除のサポートを示します。	戻り値:SQL_DI_CREATE_INDEX、SQL_DI_DROP_INDEX。
SQL_DEFAULT_TXN_ISOLATION:サーバーによるトランザクション分離のサポートを示します。	TXN_READ_COMMITTED を返します。繰り返し不可またはファントム読み取りが可能です。ダーティリードはそうではありません。
SQL_DESCRIBE_PARAMETER:DESCRIBEINPUT ステートメントのサポートを示します。	N を返します。DESCRIBEINPUT ステートメントはサポートされていません。
SQL_DM_VER:ドライバーマネージャーのバージョン。	ドライバーマネージャーによって決定されます。
SQL_DRIVER_HDBC:ドライバの接続ハンドル。	ドライバの接続ハンドルを含む SQLULEN 値を返します。
SQL_DRIVER_HDESC:ドライバ記述子ハンドル。	ドライバの記述子ハンドルを含む SQLULEN 値を返します。
SQL_DRIVER_HENV:ドライバの環境ハンドル。	ドライバの環境ハンドルを含む SQLULEN 値を返します。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_DRIVER_HLIB: ドライバーハンドル。	ライブラリハンドルを含む SQLULEN 値を返します (マネージャーがドライバーをロードしたときに ODBC ドライバーマネージャーに返されます)。
SQL_DRIVER_HSTMT: ドライバーのステートメントハンドル。	ドライバのステートメントハンドルを含む SQLULEN 値を返します。
SQL_DRIVER_NAME: ドライバーの名前。	EDB-ODBC.DLL を返します
SQL_DRIVER_ODBC_VER: ドライバーがサポートする ODBC バージョンを識別します。	03.50 を返します
SQL_DRIVER_VER: ドライバーのバージョンを識別します。	9.0.0.6 を返します
SQL_DROP_ASSERTION: サーバーでサポートされている DROPASSERTION 句を一覧表示します。	0 を返します
SQL_DROP_CHARACTER_SET: サーバーがサポートする DROPCHARACTER 句をリストします。	0 を返します
SQL_DROP_COLLATION: サーバーがサポートする DROPCOLLATION 句を一覧表示します。	0 を返します
SQL_DROP_DOMAIN: サーバーがサポートする DROPDOMAIN 句を一覧表示します。	0 を返します
SQL_DROP_SCHEMA: サーバーでサポートされている DROPSchema 句を一覧表示します。	戻り値: SQL_DS_DROP_SCHEMA、SQL_DS_RESTRICT、SQL_DS_CASCADE。
SQL_DROP_TABLE: サーバーがサポートする DROPTABLE 句をリストします。	戻り値: SQL_DT_DROP_TABLE、SQL_DS_RESTRICT、SQL_DS_CASCADE。
SQL_DROP_TRANSLATION: サーバーでサポートされている DROPTRANSLATION 句を一覧表示します。	0 を返します。
SQL_DROP_VIEW: サーバーがサポートする DROPVIEW 句をリストします。	戻り値: SQL_DV_DROP_VIEW、SQL_DS_RESTRICT、SQL_DS_CASCADE。
SQL_DYNAMIC_CURSOR_ATTRIBUTES1: ドライバーがサポートする動的カーソル属性の最初のセットについて説明します。	0 を返します
SQL_DYNAMIC_CURSOR_ATTRIBUTES2: ドライバーがサポートする動的カーソル属性の 2 番目のセットについて説明します。	0 を返します
SQL_EXPRESSIONS_IN_ORDERBY: サーバーによる ORDERBY のサポートを示します。	Y を返します。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_FETCH_DIRECTION:FETCH 順序オプションを示します (ODBC3.0 では非推奨)。	戻り値:SQL_FD_FETCH_NEXT、SQL_FD_FETCH_FIRS、SQL_FD_FETCH_LAST、SQL_FD_FETCH_PRIOR、SQL_FD_FETCH_ABSOLUTE、SQL_FD_FETCH_RELATIVE、SQL_FD_FETCH_BOOKMARK。
SQL_FILE_USAGE:単一層ドライバーがサーバー上のファイル処理する方法を示します。	SQL_FILE_NOT_SUPPORTED を返します。ドライバーは単一層ファイルではありません。
SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1:ドライバーがサポートする前方スクロールカーソル属性について説明します。	SQL_CA1_NEXT を返します。
SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES2:SQL_FORWARD_ONLY_CURSOR_ATTRIBUTES1によって指定された前方スクロールカーソルの拡張属性について説明します。	SQL_CA1_NEXT、SQL_CA1_ABSOLUTE、SQL_CA1_RELATIVE、SQL_CA1_BOOKMARK、SQL_CA1_LOCK_NO_CHANGE、SQL_CA1_POS_POSITION、SQL_CA1_POS_UPDATE、SQL_CA1_POS_DELETE、SQL_CA1_POS_CABO_BYLK_BOLK_BOLK_BOLK_BOLK_BOLK、SQL_CA2_CRC_EXACT。
SQL_GETDATA_EXTENSIONS:SQLGetData のサポートされる拡張機能をリストします。	戻り値:SQL_GD_ANY_COLUMN、SQL_GD_ANY_ORDER、SQL_GD_BLOCK、SQL_GD_BOUND。
SQL_GROUP_BY:GROUPBY 句と SELECT リストの列の間の関係を示します。	SQL_GB_GROUP_BY_EQUALS_SELECT を返します。
SQL_IDENTIFIER_CASE:SQL 識別子の大文字と小文字の区別と格納を示します。	SQL_IC_LOWER を返します。
SQL_INDEX_KEYWORDS:CREATEINDEX ステートメントのサポートを示します。	SQL_IK_NONE を返します。
SQL_INFO_SCHEMA_VIEWS:INFORMATION_SCHEMA でサポートされているビューをリストします。	N を返します。
SQL_INTEGRITY:SQL_ODBC_SQL_OPT_IEF:参照整合性構文チェックのサーバーサポートを示します。	N を返します。
SQL_INSERT_STATEMENT:INSERT ステートメントのサポートのレベルを示します。	戻り値:SQL_IS_INSERT_LITERALS、SQL_IS_INSERT_SEARCHED、SQL_IS_SELECT_INT。
SQL_KEYSET_CURSOR_ATTRIBUTES1:ドライバーがサポートするキーセットカーソル属性の最初のセットについて説明します。	戻り値:SQL_CA1_NEXT、SQL_CA1_ABSOLUTE、SQL_CA1_RELATIVE、SQL_CA1_BOOKMARK、SQL_CA1_LOCK_NO_CHANGE、SQL_CA1_POS_POSITION、SQL_CA1_POS_UPDATE、SQL_CA1_POS_DELETE、SQL_CA1_POS_CABO_BYLK_BOLK_BOLK_BOLK_BOLK_BOLK、SQL_CA2_CRC_EXACT。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_KEYSET_CURSOR_ATTRIBUTES2: ドライバーがサポートするキーセットカーソル属性の 2 番目のセットについて説明します。	戻り値: SQL_CA2_READ_ONLY_CONCURRENCY、SQL_CA2_OPT_ROWVER_CONCURRENCY、SQL_CA2_SENSITIVITY_ADDITIONS、SQL_CA2_SENSITIVITY_DELETIONS、SQL_CA2_SENSITIVITY_UPDATES、SQL_CA2_CRC_EXACT。
SQL_KEYWORDS: サーバー固有の予約済みキーワードを識別します。	戻り値 " "。サーバー固有の予約済みキーワードはありません。
SQL_LIKE_ESCAPE_CLAUSE: LIKE 述語でのエスケープ文字のサポートを示します。	AdvancedServer は LIKE 述語のエスケープ文字をサポートしていません。
SQL_LOCK_TYPES: サポートされているロックタイプをリストします (ODBC3.0 では非推奨)。	SQL_LCK_NO_CHANGE を返します。
SQL_MAX_ASYNC_CONCURRENT_STATEMENTS: ドライバーがサポートできるアクティブな並行ステートメントの数。	この info_type は現在サポートされていません。
SQL_MAX_BINARY_LITERAL_LEN: バイナリリテラルの最大長。	0 を返します。最大長は指定されていません。
SQL_MAX_CATALOG_NAME_LEN: サーバー上のカタログ名の最大長。	0 を返します。最大長は指定されていません。
SQL_MAX_QUALIFIER_NAME_LEN: 修飾子の最大長。	0 を返します。最大長は指定されていません。
SQL_MAX_CHAR_LITERAL_LEN: 文字ストリングの最大文字数。	0 を返します。最大長は指定されていません。
SQL_MAX_COLUMN_NAME_LEN: 列名の最大長。	64 を返します。列名の長さは 64 文字を超えることはできません。
SQL_MAX_COLUMNS_IN_GROUP_BY: GROUPBY 句で許可される列の最大数。	0 を返します。最大長は指定されていません。
SQL_MAX_COLUMNS_IN_INDEX: インデックスで許可される列の最大数。	0 を返します。最大長は指定されていません。
SQL_MAX_COLUMNS_IN_ORDER_BY: ORDERBY 句で許可される列の最大数。	0 を返します。最大長は指定されていません。
SQL_MAX_COLUMNS_IN_SELECT: SELECT リストで許可される列の最大数。	0 を返します。最大長は指定されていません。
SQL_MAX_COLUMNS_IN_TABLE: テーブルで許可される列の最大数。	0 を返します。最大長は指定されていません。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_MAX_CONCURRENT_ACTIVITIES 前へ SQL_MAX_ACTIVE_STATEMENTS: ドライバーがサポートできるアクティブな SQL ステートメントの最大数。	0 を返します。最大長は指定されていません。
SQL_MAX_CURSOR_NAME_LEN: カーソル名の最大長。	32 を返します。カーソル名の長さは 32 文字を超えることはできません。
SQL_MAX_DRIVER_CONNECTIONS 前へ SQL_ACTIVE_CONNECTIONS: ドライバーがサポートできるアクティブな接続の最大数。	0 を返します。サポートされる接続の数に制限はありません。
SQL_MAX_IDENTIFIER_LEN: サーバーで許可される ID の最大長。	64 を返します。識別子の長さは 64 文字を超えることはできません。
SQL_MAX_INDEX_SIZE: インデックスの（結合された）フィールドで許可される最大バイト数。	0 を返します。最大サイズは指定されていません。
SQL_MAX_OWNER_NAME_LEN: サーバーで許可されている所有者名の最大長。	64 を返します。所有者名の最大長は 64 文字です。
SQL_MAX_PROCEDURE_NAME_LEN: サーバーが許可するプロシージャ名の最大長。	0 を返します。最大長は指定されていません。
SQL_MAX_QUALIFIER_NAME_LEN: サーバーで許可されている修飾子名の最大長。	0 を返します。修飾子の最大長は指定されていません。
SQL_MAX_ROW_SIZE: 行の最大長。	0 を返します。行の最大長は指定されていません。
SQL_MAX_ROW_SIZE_INCLUDES_LONG: SQL_MAX_ROW_SIZE には、行の LONGVARCHAR または LONGVARBINARY 列の長さが含まれているかどうかを示します。	SQL_MAX_ROW_SIZE には、行の LONGVARCHAR または LONGVARBINARY 列の長さが含まれます。
SQL_MAX_SCHEMA_NAME_LEN: サーバーで許可されるスキーマ名の最大長。	64 を返します。スキーマ名の最大長は 64 文字です。
SQL_MAX_STATEMENT_LEN: SQL ステートメントの最大長。	0 を返します。ステートメントの最大長は、使用可能なメモリによって制限されます。
SQL_MAX_TABLE_NAME_LEN: サーバーが許可するテーブル名の最大長。	64 を返します。テーブル名の最大長は 64 文字です。
SQL_MAX_TABLES_IN_SELECT: SELECT ステートメントの FROM 句で許可されるテーブルの最大数。	0 を返します。許可されるテーブルの最大数は指定されていません。
SQL_MAX_USER_NAME_LEN: サーバーで許可されているユーザー名の最大長。	0 を返します。ユーザー名の最大長は指定されていません。
SQL_MULT_RESULT_SETS: サーバーが複数の結果セットをサポートすることを示します。	Y を返します。AdvancedServer は複数の結果セットをサポートしています。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_MULTIPLE_ACTIVE_TXN:サーバーが複数のアクティブなトランザクションをサポートするかどうかを示します。	Y を返します。AdvancedServer は複数のアクティブなトランザクションをサポートしています。
SQL_NEED_LONG_DATA_LEN:サーバーが値を受け取る前に LONG データ値の長さが必要かどうかを示します。	N を返します。AdvancedServer は、値を受け取る前に LONG データ値の長さを必要としません。
SQL_NON_NULLABLE_COLUMNS:サーバーが列で NOTNULL 値をサポートするかどうかを示します。	SQL_NNC_NON_NULL を返します。AdvancedServer は、列の NOTNULL 値をサポートします。
SQL_NULL_COLLATION:NULL 値が結果セットのどこにあるかを示します。	SQL_NC_HIGH を返します。データセット内の NULL 値の場所は、ASC および DESC キーワードによって決定されます。NULL 値は、データセットの上限にソートされます。
SQL_NUMERIC_FUNCTIONS:ドライバーとサーバーでサポートされている数値関数をリストします。	戻り値:SQL_FN_NUM_ABS、SQL_FN_NUM_ATAN、SQL_FN_NUM_CEILING、SQL_FN_NUM_COS、SQL_FN_NUM_EXP、SQL_FN_NUM_FLOOR、SQL_FN_NUM_LOG、SQL_FN_NUM_MOD、SQL_FN_NUM_SIGN、SQL_FN_NUM_SIN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN、SQL_FN_NUM_SUN
SQL_ODBC_API_CONFORMANCE:ODBC3.0 準拠レベルを示します	SQL_OAC_LEVEL1 を返します。ドライバーは、ODBC レベル 1 インターフェースに準拠しています。
SQL_ODBC_INTERFACE_CONFORMANCE:ドライバーが準拠する ODBC インターフェースを示します。	SQL_OIC_CORE を返します。
SQL_ODBC_SAG_CLI_CONFORMANCE:ドライバーが準拠している SQL アクセスグループのコンプライアンスレベルを示します。	SQL_OSCC_NOT_COMPLIANT を返します。ドライバーは SAGCLI に準拠していません。
SQL_ODBC_SQL_CONFORMANCE:ドライバーが準拠する SQL 文法レベルを示します。	SQL_OSC_CORE を返します。ドライバーはコア文法レベルに準拠しています。
SQL_ODBC_SQL_OPT_IEF になりました SQL_INTEGRITY:参照整合性構文チェックのサーバーサポートを示します。	N を返します。サーバーは参照整合性構文チェックをサポートしていません。
SQL_ODBC_VER:ドライバーマネージャーでサポートされている ODBC バージョン	03.52.0000 を返します。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_OJ_CAPABILITIES:サーバーでサポートされている外部結合を識別します。	戻り値:SQL_OJ_LEFT、SQL_OJ_RIGHT、SQL_OJ_FULL、SQL_OJ_NESTED、SQL_OJ_NOT_ORDERED、SQL_OJ_INNER、SQL_OJ_ALL_COMPARISON_OPS。
SQL_OUTER_JOINS:外部結合と外部結合エスケープシーケンスのサポートを示します。	Y を返します。外部結合がサポートされています。
SQL_OWNER_TERM 前へ SQL_SCHEMA_TERM:スキーマを説明するために使用される用語。	スキーマを返します。
SQL_ORDER_BY_COLUMNS_IN_SELECT:ORDERBY 句の列を SELECT リストに含める必要があるかどうかを示します。	Y を返します。ORDERBY 句の列は、SELECT リストにある必要はありません。
SQL_OWNER_USAGESQL_SCHEMA_USAGE:どのステートメントがスキーマ修飾子をサポートするかを示す文字列を返します。	戻り値:SQL_OU_DML_STATEMENTS、SQL_OU_TABLE_DEFINITION、SQL_OU_INDEX_DEFINITION、SQL_OU_PRIVILEGE_DEFINITION。
SQL_PARAM_ARRAY_ROW_COUNTS:少なくとも 1 つのパラメーターが配列にバインドされているパラメーター化されたステートメントを実行したときに、サーバーが配列内の各要素に対して単一の行数または個別の行数を返すかどうかを示します。	配列内の各要素に対して個別の行数が利用可能な場合、SQL_PARC_BATCH を返します。SQL_PARC_NO_BATCH (配列全体で単一の累積行カウントが使用可能な場合)。
SQL_PARAM_ARRAY_SELECTS:少なくとも 1 つのパラメーターが配列にバインドされているパラメーター化されたステートメントを実行したときに、サーバーが配列内の各要素に対して 1 つの結果セットまたは個別の結果セットを返すか (またはドライバーがこの機能を許可しない場合) を示します。	SQL_PAS_BATCH を返します。配列の要素ごとに 1 つのデータセットを使用できます。
SQL_POS_OPERATION:SQLSetPos () がサポートするオプションをリストします。	戻り値:SQL_POS_POSITION、SQL_POS_REFRESH、SQL_POS_UPDATE、SQL_POS_DELETE、SQL_POS_ADD。
SQL_POSITIONED_STATEMENTS:サポートされている位置付け SQL ステートメントをリストします。	戻り値:SQL_PS_POSITIONED_DELETE、SQL_PS_POSITIONED_UPDATE、SQL_PS_SELECT_FOR_UPDATE。
SQL_PROCEDURE_TERM:プロシージャの説明に使用される用語。	返品手順。
SQL_PROCEDURES:サーバーとドライバーが SQL プロシージャとプロシージャ呼び出し構文をサポートするかどうかを示します。	Y を返します。サーバーとドライバーのサポートプロシージャおよびプロシージャ呼び出し構文。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_QUALIFIER_LOCATION 前。 SQL_CATALOG_LOCATION:修飾テーブル名でのスキーマ名の位置を示します。	SQL_CL_START を返します。修飾テーブル名のカタログ部分は、名前の最初にあります。
SQL_QUALIFIER_NAME 前へ SQL_CATALOG_NAME:サーバーがカタログ名をサポートしていることを示します。	Y を返します。サーバーはカタログ名をサポートしています。
SQL_QUALIFIER_NAME_SEPARATOR 前へ SQL_CATALOG_NAME_SEPARATOR:修飾子名と隣接する名前要素を区切る文字。	戻り値';。';。サーバーは`。`を期待します修飾子とテーブル名の間の文字。
SQL_QUALIFIER_TERM 前へ SQL_CATALOG_TERM:修飾子を説明するために使用される用語。	カタログを返します。
SQL_QUALIFIER_USAGE 前へ SQL_CATALOG_USAGE:修飾子を参照する可能性のある SQL ステートメントを示します。	SQL_CU_DML_STATEMENTS を返します。カタログ名は、SELECT、INSERT、UPDATE、DELETE、SELECT-FORUPDATE、および位置付け UPDATE および DELETE ステートメントで使用できます。
SQL_QUALIFIER_USAGESQL_CATALOG_USAGE:修飾子名をサポートする DML ステートメントを識別します。	SQL_CU_DML_STATEMENTS を返します。修飾子は、すべての DML ステートメント (SELECT、INSERT、UPDATE、DELETE、SELECTFORUPDATE) で使用できます。
SQL_QUOTED_IDENTIFIER_CASE:引用された識別子の小文字と大文字の区別を示します。	SQL_IC_SENSITIVE を返します。引用識別子は小文字と大文字を区別します。
SQL_QUALIFIER_NAME_SEPARATORSQL_CATALOG_NAME_SEPARATOR:修飾子の区切り文字として使用されます。	SQL_QUALIFIER_NAME_SEPARATOR:修飾子の区切り文字として使用されます。
SQL_QUALIFIER_TERM:修飾子を説明するために使用される用語。	返品カタログ
SQL_QUALIFIER_LOCATION:修飾されたテーブル名での修飾子の位置。	SQL_CL_START を返します。修飾子は、修飾テーブル名でテーブル名の前にあります。
SQL_ROW_UPDATES:キーセットドリブンカーソルまたは混合カーソルが行のバージョンまたは値を維持するかどうかを示します。	Y を返します。カーソルは、フェッチされたすべての行の値を維持し、行の値の更新を検出できます。
SQL_SCHEMA_TERM:スキーマを説明するために使用される用語。	スキーマを返します
SQL_SCHEMA_USAGE:スキーマを参照する可能性のある SQL ステートメントを示します。	戻り値:SQL_OU_DML_STATEMENTS、SQL_OU_TABLE_DEFINITION、SQL_OU_INDEX_DEFINITION、SQL_OU_PRIVILEGE_DEFINITION。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_SCROLL_CONCURRENCY: サーバーでサポートされているカーソル同時実行制御オプションを示します。	戻り値: SQL_SCCO_READ_ONLY、SQL_SCCO_OPT_ROWVER。
SQL_SCROLL_OPTIONS: サーバーがサポートするカーソルスクロールオプションを示します。	戻り値: SQL_SO_FORWARD_ONLY、SQL_SO_KEYSET_DRIVEN、SQL_SO_STATIC。
SQL_SEARCH_PATTERN_ESCAPE: ワイルドカード文字 % および _ を検索パターンで使えるようにするエスケープ文字。	戻り値 /。 `` 文字は、検索パターンの % `` および `` `` 文字のエスケープ文字として使用されます。
SQL_SERVER_NAME: ホストの名前を示します。	戻り値は、接続プロパティによって決定されます。
SQL_SPECIAL_CHARACTERS: 識別子名に使用できる特殊文字を示します。	戻り値。識別子名にはアンダースコア文字を使用できません。
SQL_SQL_CONFORMANCE: SQL-92 準拠のレベルを示します。	SQL_SC_SQL92_ENTRY を返します。ドライバは SQL92 エントリレベルに準拠しています。
SQL_SQL92_DATETIME_FUNCTIONS: サーバーがサポートする日時関数をリストします。	戻り値: SQL_SDF_CURRENT_DATE、SQL_SDF_CURRENT_TIME、SQL_SDF_CURRENT_TIMESTAMP。
SQL_SQL92_FOREIGN_KEY_DELETE_RULE: DELETE ステートメントで外部キーを使用するためのサーバー強制ルールを示します。	戻り値: SQL_SFKD_CASCADE、SQL_SFKD_NO_ACTION、SQL_SFKD_SET_DEFAULT、SQL_SFKD_SET_NULL。
SQL_SQL92_FOREIGN_KEY_UPDATE_RULE: UPDATE ステートメントで外部キーを使用するためのサーバー強制ルールを示します。	戻り値: SQL_SFKU_CASCADE、SQL_SFKU_NO_ACTION、SQL_SFKU_SET_DEFAULT、SQL_SFKU_SET_NULL。
SQL_SQL92_GRANT: サポートされている GRANT ステートメント句を示します。	戻り値: SQL_SG_DELETE_TABLE、SQL_SG_INSERT_TABLE、SQL_SG_REFERENCES_TABLE、SQL_SG_SELECT_TABLE、SQL_SG_UPDATE_TABLE。
SQL_SQL92_NUMERIC_VALUE_FUNCTIONS: サーバーとドライバでサポートされているスカラー数値関数をリストします。	戻り値: SQL_SNVF_BIT_LENGTH、SQL_SNVF_CHAR_LENGTH、SQL_SNVF_CHARACTER_LENGTH、SQL_SNVF_EXTRACT、SQL_SNVF_OCTET_LENGTH、SQL_SNVF_POSITION。
SQL_SQL92_PREDICATES: サーバーでサポートされている SELECT ステートメントの述語を識別します。	戻り値: SQL_SP_EXISTS、SQL_SP_ISNOTNULL、SQL_SP_ISNULL、SQL_SP_OVERLAPS、SQL_SP_LIKE、SQL_SP_IN、SQL_SP_BETWEEN、SQL_SP_COMPARISON、SQL_SP_QUANTIFIED_COMPARISON。

次のページに続く

表 1 – 前のページからの続き

SQLInfo_type の引数と説明	EDB_ODBC/AdvancedServer の戻り値:
SQL_STATIC_SENSITIVITY:SQLSetPos () または UPDATE または DELETE ステートメントによって静的カーソルに加えられた変更がアプリケーションによって検出されたかどうかを示します。	戻り値:SQL_SS_ADDITIONS、SQL_SS_DELETIONS、SQL_SS_UPDATES。
SQL_STRING_FUNCTIONS:サーバーとドライバーでサポートされているスカラー文字列関数をリストします。	戻り値:SQL_FN_STR_CONCAT、SQL_FN_STR_LTRIM、SQL_FN_STR_LENGTH、SQL_FN_STR_LOCATE、SQL_FN_STR_LCASE、SQL_FN_STR_RTRIM、SQL_FN_STR_SUBSTRING、SQL_FN_STR_UCASE。
SQL_SUBQUERIES:サーバーでサポートされている SELECT ステートメントのサブクエリ述語を識別します。	戻り値:SQL_SQ_COMPARISON、SQL_SQ_EXISTS、SQL_SQ_IN、SQL_SQ_QUANTIFIED。
SQL_SYSTEM_FUNCTIONS:サーバーとドライバーがサポートするスカラーシステム関数を一覧表示します。	0 を返します。
SQL_TABLE_TERM:テーブルの説明に使用される用語。	テーブルを返します。
SQL_TIMEDATE_ADD_INTERVALS:TIMESTAMPADD() スカラー関数についてサーバーがサポートするタイムスタンプ間隔を示します。	0 を返します。
SQL_TIMEDATE_DIFF_INTERVALS:TIMESTAMPDIFF() スカラー関数についてサーバーがサポートするタイムスタンプ間隔を示します。	0 を返します。
SQL_TIMEDATE_FUNCTIONS:サーバーがサポートする日付と時刻の関数を示します。	戻り値:SQL_FN_TD_NOW、SQL_FN_TD_CURDATE、SQL_FN_TD_CURTIME。
SQL_TXN_CAPABLE:サーバーとドライバーによって提供されるトランザクションサポートを識別します。	SQL_TC_ALL を返します。トランザクションには、DML ステートメントと DDL ステートメントの両方を含めることができます。
SQL_TXN_ISOLATION_OPTION:サーバーがサポートするトランザクション分離レベルを示します。	戻り値:SQL_TXN_READ_COMMITTED、SQL_TXN_SERIALIZABLE。
SQL_UNION:サーバーが UNION 句をサポートしていることを示します。	戻り値:SQL_U_UNION、SQL_U_UNION_ALL。
SQL_USER_NAME:データベースに接続しているユーザーの名前を識別します。ログイン名とは異なる場合があります。	この値は、接続プロパティによって決定されます。
SQL_XOPEN_CLI_YEAR:ドライバーマネージャーが準拠する X/Open 仕様の発行年。	この info_type は現在サポートされていません。

6.2 接続属性

ODBC の `SQLGetConnectAttr()` および `SQLSetConnectAttr()` 関数を使用して、接続属性の値を取得または設定できます。

6.2.1 `SQLGetConnectAttr()`

`SQLGetConnectAttr()` 関数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`

`attribute` は値を取得したい属性を識別します。

`value_pointer`

`attribute` の値を受け取るメモリ内の場所へのポインタ。

`buffer_length`

If `attribute` is defined by ODBC and `value_pointer` points to a character string or binary buffer, `buffer_length` is the length of `value_pointer`. If `value_pointer` points to a fixed-size value (such as an integer), `buffer_length` is ignored.

If EDB-ODBC defines the `attribute`, `SQLGetConnectAttr()` sets the `buffer_length` parameter. `buffer_length` can be:

値のタイプ	意味
文字列	文字列の長さ
バイナリバッファ	<code>SQL_LEN_BINARY_ATTR</code> (長さ) の結果
固定長データ型	<code>SQL_IS_INTEGER</code> または <code>SQL_IS_INTEGER</code>
その他のタイプ	<code>SQL_IS_POINTER</code>

string_length_pointer

value_pointer で返すために利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。

次の表に、EDB-ODBC でサポートされている接続属性を示します。

属性	サポートされていますか？	ノート
SQL_ATTR_ACCESS_MODE	番号	SQL_MODE_READ_WRITE
SQL_ATTR_ASYNC_ENABLE	番号	SQL_ASYNC_ENABLE_OFF
SQL_ATTR_AUTO_IPD	番号	
SQL_ATTR_AUTOCOMMIT		SQL_AUTOCOMMIT、 SQL_AUTOCOMMIT_ON、 SQL_AUTOCOMMIT_OFF
SQL_ATTR_CONNECTION_TIMEOUT	番号	
SQL_ATTR_CURRENT_CATALOG	番号	
SQL_ATTR_DISCONNECT_BEHAVIOR	番号	
SQL_ATTR_ENLIST_IN_DTC	はい	win32 および条件付きコンパイルの場合
SQL_ATTR_ENLIST_IN_XA	番号	
SQL_ATTR_LOGIN_TIMEOUT	番号	SQL_LOGIN_TIMEOUT
SQL_ATTR_ODBC_CURSORS	番号	
SQL_ATTR_PACKET_SIZE	番号	
SQL_ATTR_QUIET_MODE	番号	
SQL_ATTR_TRACE	番号	
SQL_ATTR_TRACEFILE	番号	
SQL_ATTR_TRANSLATE_LIB	番号	
SQL_ATTR_TRANSLATE_OPTION	番号	
SQL_ATTR_TXN_ISOLATION	はい	SQL_TXN_ISOLATION、 SQL_DEFAULT_TXN_ISOLATION

6.2.2 SQLSetConnectAttr ()

ODBC SQLSetConnectAttr() 関数を使用して、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

attribute は、値を設定する属性を識別します

value_pointer

attribute が想定する値へのポインタ。

string_length

If attribute is defined by ODBC and value_pointer points to a binary buffer or character string, string_length is the length of value_pointer. If value_pointer points to a fixed-length value (such as an integer), string_length is ignored.

If EDB-ODBC defines the attribute, the application sets the string_length parameter. Possible string_length values are:

値のタイプ	意味
文字列	文字ストリングまたは SQL_NTS の長さ
バイナリバッファ	SQL_LEN_BINARY_ATTR (長さ) の結果
固定長データ型	SQL_IS_INTEGER または SQL_IS_INTEGER
その他のタイプ	SQL_IS_POINTER

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

You can call SQLSetConnectAttr() any time after the connection handle is allocated, until the time that the connection is closed with a call to SQLFreeHandle(). All attributes set by the call persist until the call to SQLFreeHandle().

接続属性には、設定可能な特定の時間枠があります。一部の属性は接続が確立される前に設定する必要がありますが、他の属性は接続が確立された後にのみ設定できます。

次の表は、接続属性とそれらを設定できる期間を示しています。

属性	接続を確立する前または後に設定しますか？
SQL_ATTR_ACCESS_MODE	前または後
SQL_ATTR_ASYNC_ENABLE	前または後
SQL_ATTR_AUTO_IPD	前または後
SQL_ATTR_AUTOCOMMIT	前または後
SQL_ATTR_CONNECTION_TIMEOUT	前または後
SQL_ATTR_CURRENT_CATALOG	前または後
SQL_ATTR_ENLIST_IN_DTC	後
SQL_ATTR_ENLIST_IN_XA	後
SQL_ATTR_LOGIN_TIMEOUT	前
SQL_ATTR_ODBC_CURSORS	前
SQL_ATTR_PACKET_SIZE	前
SQL_ATTR_QUIET_MODE	前または後
SQL_ATTR_TRACE	前または後
SQL_ATTR_TRACEFILE	前または後
SQL_ATTR_TRANSLATE_LIB	後
SQL_ATTR_TRANSLATE_OPTION	後
SQL_ATTR_TXN_ISOLATION	前または後

6.3 環境属性

ODBC の `SQLGetEnvAttr()` および `SQLSetEnvAttr()` 関数を使用して、環境属性の値を取得または設定できます。

6.3.1 `SQLGetEnvAttr()`

システムの環境属性の現在の値を見つけるには、`SQLGetEnvAttr()` 関数を使用します。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`

`attribute` は値を取得したい属性を識別します。

`value_pointer`

`attribute` の値を受け取るメモリ内の場所へのポインタ。

`buffer_length`

If the attribute is a character string, `buffer_length` is the length of `value_ptr`. If the value of the attribute is not a character string, `buffer_length` is unused.

`string_length_pointer`

`value_pointer` で返すために利用可能なバイト数を受け取る `SQLINTEGER` へのポインタ。 ```value_pointer*` is NULL, `` `string_length_pointer``` が返されない場合。

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。

次の表に、EDB-ODBC でサポートされている環境属性を示します。

属性	サポートされていますか？	制限？
SQL_ATTR_CONNECTION_POOLING	SQL_CP_ONE_PER_DRIVER または SQL_CP_OFF	接続プロパティによって決定
SQL_ATTR_ODBC_VERSION	(SQL_OV_ODBC3) (SQL_OV_ODBC2)	なし
SQL_ATTR_OUTPUT_NTS	SQL_SUCCESS	なし

6.3.2 SQLSetEnvAttr ()

SQLSetEnvAttr () 関数を使用して、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

attribute は値を設定したい属性を識別します。

value_pointer

A pointer to the value assigned to the attribute. The value will be a NULL terminated character string or a 32 bit integer value depending on the specified attribute.

string_length

If value_pointer is a pointer to a binary buffer or character string, string_length is the length of value_pointer. If the value being assigned to the attribute is a character, string_length is the length of that character string. If value_pointer is NULL, string_length is not returned. If value_pointer is an integer, string_length is ignored.

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 () を呼び出さなければなりません。環境属性に適用されるすべての値は、接続のために 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。

次の表は、SQLSetAttr () で設定できる環境属性の一覧です。

属性	Value_pointer タイプ	制限？
SQL_ATTR_ODBC_VERSION	32 ビット整数	アプリケーションが SQLHENV 引数を含む関数を呼び出す前に、この属性を設定してください。
SQL_ATTR_OUTPUT_NTS	32 ビット整数	デフォルトは SQL_TRUE です。この属性を SQL_FALSE に設定する呼び出しは、SQL_ERROR/SQLSTATEHYC00（機能は実装されていません）を返します。

6.4 ステートメント属性

ODBC の `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。

6.4.1 `SQLGetStmtAttr()`

`SQLGetStmtAttr()` 関数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`

`attribute` は属性値です

`value_pointer`

`attribute` の値を受け取るメモリ内の場所へのポインタ。

`buffer_length`

If the attribute is defined by ODBC, `buffer_length` is the length of `value_pointer` (if `value_pointer` points to a character string or binary buffer). If `value_pointer` points to an integer, `buffer_length` is ignored.

If EDB-ODBC defines the attribute, the application sets the `buffer_length` parameter. `buffer_length` can be:

値のタイプ	意味
文字列	文字列の長さ
バイナリバッファ	<code>SQL_LEN_BINARY_ATTR</code> (長さ) の結果
固定長データ型	<code>SQL_IS_INTEGER</code> または <code>SQL_IS_INTEGER</code>
その他のタイプ	<code>SQL_IS_POINTER</code>

`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。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。

次の表に、EDB-ODBC でサポートされているステートメント属性を示します。

属性	サポートされていますか？	制限？
SQL_ATTR_APP_PARAM_DESC	はい	
SQL_ATTR_APP_ROW_DESC	はい	
SQL_ATTR_ASYNC_ENABLE	番号	
SQL_ATTR_CONCURRENCY	はい	SQL_CONCUR_READ_ONLY
SQL_ATTR_CURSOR_SCROLLABLE	はい	
SQL_ATTR_CURSOR_TYPE	はい	SQL_CURSOR_FORWARD_ONLY
SQL_ATTR_CURSOR_SENSITIVITY	はい	SQL_INSENSITIVE
SQL_ATTR_ENABLE_AUTO_IPD	番号	
SQL_ATTR_FETCH_BOOKMARK_PTR	はい	
SQL_ATTR_IMP_PARAM_DESC	はい	
SQL_ATTR_IMP_ROW_DESC	はい	
SQL_ATTR_KEYSET_SIZE	番号	
SQL_ATTR_MAX_LENGTH	番号	
SQL_ATTR_MAX_ROWS	番号	
SQL_ATTR_METADATA_ID	はい	
SQL_ATTR_NOSCAN	番号	
SQL_ATTR_PARAM_BIND_OFFSET_PTR	はい	ODBCV2.0
SQL_ATTR_PARAM_BIND_TYPE	はい	
SQL_ATTR_PARAM_OPERATION_PTR	はい	
SQL_ATTR_PARAM_STATUS_PTR	はい	
SQL_ATTR_PARAMS_PROCESSED_PTR	はい	
SQL_ATTR_PARAMSET_SIZE	はい	
SQL_ATTR_QUERY_TIMEOUT	番号	
SQL_ATTR_RETRIEVE_DATA	番号	
SQL_ATTR_ROW_BIND_OFFSET_PTR	はい	
SQL_ATTR_ROW_BIND_TYPE	番号	
SQL_ATTR_ROW_NUMBER	はい	
SQL_ATTR_ROW_OPERATION_PTR	はい	
SQL_ATTR_ROW_STATUS_PTR	はい	
SQL_ATTR_ROWS_FETCHED_PTR	はい	

次のページに続く

表 2 – 前のページからの続き

属性	サポートされていますか？	制限？
SQL_ATTR_ROW_ARRAY_SIZE	はい	
SQL_ATTR_SIMULATE_CURSOR	番号	
SQL_ATTR_USE_BOOKMARKS	はい	
SQL_ROWSET_SIZE	はい	

。生::ラテックス

/新しいページ

6.4.2 SQLSetStmtAttr ()

SQLSetStmtAttr () 関数を使用して、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
    SQLPOINTER *value_pointer*, //Input
    SQLINTEGER *string_length* //Input
);
```

stmt_handle

stmt_handleは環境ハンドルです。

attribute

attribute は値を設定したいステートメント属性を識別します。

value_pointer

value_pointer is a pointer to the location in memory that holds the value that will be assigned to the attribute.
value_pointer can be a pointer to:

- null で終了する文字列
- バイナリバッファ
- ドライバーによって定義された値

- タイプ `SQLLEN`、`SQLULEN`、または `SQLUSMALLINT` の値

値ポインターは、オプションで以下の値のいずれかを保持することもできます。

- ODBC 記述子ハンドル
- `SQLUIINTEGER` の値
- `SQLULEN` の値
- 符号付き整数（属性がドライバー固有の値の場合）

`string_length`

If attribute is defined by ODBC and `value_pointer` points to a binary buffer or character string, `string_length` is the length of `value_pointer`. If `value_pointer` points to an integer, `string_length` is ignored.

If EDB-ODBC defines the attribute, the application sets the `string_length` parameter. Possible `string_length` values are:

値のタイプ	意味
文字列	文字ストリングまたは <code>SQL_NTS</code> の長さ
バイナリバッファ	<code>SQL_LEN_BINARY_ATTR</code> （長さ）の結果
固定長データ型	<code>SQL_IS_INTEGER</code> または <code>SQL_IS_INTEGER</code>
その他のタイプ	<code>SQL_IS_POINTER</code>

6.5 エラー処理

このガイドで言及されている ODBC 関数の診断情報は、ODBC `SQLGetDiagRec()` 関数を介して取得できます。

6.5.1 `SQLGetDiagRec()`

`SQLGetDiagRec()` 関数は、属性値を取得または設定する ODBC 関数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。

。コードブロック::テキスト

```
SQLRETURN SQLGetDiagRec (
    SQLSMALLINT*handle_type*, //入力
    SQLHANDLE*handle*, //入力
    SQLSMALLINT*record_number*, //入力
    SQLCHAR/**SQLState_pointer, //出力
    SQLINTEGER/**native_error_pointer, //出力
    SQLCHAR/**error_text_pointer, //出力
    SQLSMALLINT*buffer_length*, //入力
    SQLSMALLINT/*text_length_pointer//出力
);
```

`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探しているステータスレコード (1 以上である必要があります)。

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へのポインタ。

native_error_pointer

データソースのネイティブエラーメッセージを受け取るバッファへのポインタ（SQL_DIAG_NATIVE フィールドに含まれています）。

error_text_pointer

エラーテキストを受け取るメモリバッファへのポインタ（SQL_DIAG_MESSAGE_TEXT フィールドに含まれています）

buffer_length

error_text バッファの長さ。

text_length_pointer

Pointer to the buffer that receives the size (in characters) of the error_text_pointer field. If the number of characters in the error_text_pointer parameter exceeds the number available (in buffer_length), error_text_pointer will be truncated.

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

6.6 サポートされる 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、右側の列は Yes を示します。このリストの現在のバージョンを返すには、ODBC SQLGetFunctions() 関数 (SQL_API_ODBC3_ALL_FUNCTIONS の関数 ID を指定) を使用します。

ODBC API 関数名	EDB-ODBC でサポートされていますか？
SQLAllocConnect ()	はい
SQLAllocEnv ()	はい
SQLAllocStmt ()	はい
SQLBindCol ()	はい
SQLCancel ()	はい
SQLColAttributes ()	はい
SQLConnect ()	はい
SQLDescribeCol ()	はい
SQLDisconnect ()	はい
SQLError ()	はい
SQLExecDirect ()	はい
SQLExecute ()	はい
SQLFetch ()	はい
SQLFreeConnect ()	はい
SQLFreeEnv ()	はい
SQLFreeStmt ()	はい
SQLGetCursorName ()	はい
SQLNumResultCols ()	はい
SQLPrepare ()	はい
SQLRowCount ()	はい
SQLSetCursorName ()	はい
SQLSetParam ()	はい
SQLTransact ()	はい
SQLColumns ()	はい
SQLDriverConnect ()	はい
SQLGetConnectOption ()	はい
SQLGetData ()	はい
SQLGetFunctions ()	はい
SQLGetInfo ()	はい
SQLGetStmtOption ()	はい
SQLGetTypeInfo ()	はい

[次のページに続く](#)

表 3 – 前のページからの続き

ODBCAPI 関数名	EDB-ODBC でサポートされていますか？
SQLParamData ()	はい
SQLPutData ()	はい
SQLSetConnectOption ()	はい
SQLSetStmtOption ()	はい
SQLSpecialColumns ()	はい
SQLStatistics ()	はい
SQLTables ()	はい
SQLBrowseConnect ()	番号
SQLColumnPrivileges ()	番号
SQLDataSources ()	はい
SQLDescribeParam ()	番号
SQLExtendedFetch ()	はい
SQLForeignKeys ()	はい
SQLMoreResults ()	はい
SQLNativeSQL ()	はい
SQLNumParams ()	はい
SQLParamOptions ()	はい
SQLPrimaryKeys ()	はい
SQLProcedureColumns ()	はい
SQLProcedures ()	はい
SQLSetPos ()	はい
SQLSetScrollOptions ()	番号
SQLTablePrivileges ()	はい
SQLDrivers ()	はい
SQLBindParameter ()	はい
SQLAllocHandle ()	はい
SQLBindParam ()	はい
SQLCloseCursor ()	はい
SQLColAttribute ()	はい
SQLCopyDesc ()	はい
SQLEndTran ()	はい
SQLFetchScroll ()	はい
SQLFreeHandle ()	はい
SQLGetConnectAttr ()	はい
SQLGetDescField ()	はい
SQLGetDescRec ()	はい
SQLGetDiagField ()	はい

次のページに続く

表 3 – 前のページからの続き

ODBCAPI 関数名	EDB-ODBC でサポートされていますか？
SQLGetDiagRec ()	はい
SQLGetEnvAttr ()	はい
SQLGetStmtAttr ()	はい
SQLSetConnectAttr ()	はい
SQLSetDescField ()	はい
SQLSetDescRec ()	番号
SQLSetEnvAttr ()	はい
SQLSetStmtAttr ()	はい
SQLBulkOperations ()	はい

6.7 サポートされているデータ型

EDB-ODBC は、次の ODBC データ型をサポートしています。

ODBC データ型	対応する AdvancedServer データ型
SQL_BIGINT	PG_TYPE_INT8
SQL_BINARY	PG_TYPE_BYTEA
SQL_BIT	PG_TYPE_BOOL または 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 または 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

6.8 スレッドセーフティ

EDB-ODBC はスレッドセーフです。

第 7 章

セキュリティと暗号化

7.1 スクラムの互換性

EDBODBC ドライバーは、AdvancedServer バージョン 12、11、10 に SCRAM-SHA-256 サポートを提供します。このサポートは、EDBODBC10.01.0000.01 リリース以降で利用できます。

第 8 章

結論

EDBPostgresODBC コネクタガイド

Copyright©2007-2019EnterpriseDBCorporation。

全著作権所有。

EnterpriseDB®Corporation

34CrosbyDrive、Suite201、Bedford、MA01730、USA

T+17813573390F+19784671307E

info@enterprisedb.com

www.enterprisedb.com

- EDB は、EDBPostgres 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、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最小限のリスクを提供できるように設計およびテストされることを保証します。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

Adding data source definition in linux, 39
Adding data source definition in windows, 29

EDB-ODBC ドライバーの機能, 47
EDB-ODBC のインストール, 7
EDB-ODBC の概要, 7
EDB-ODBC 接続プロパティ, 21

Installing ODBC on Debian, 13
Installing ODBC on SLES, 11
Installing ODBC on Ubuntu, 13
Installing ODBC using RPM, 8
Installing ODBC with graphical installer, 14

ODBC API functions, 78
ODBC connection attributes, 64
ODBC environment attributes, 68
ODBC statement attributes, 72

Status and error data, 76
Supported data types, 81
supported server versions, 5

Using ODBC functions, 47

スクラムの互換性, 83

セキュリティと暗号化, 83

データソースの作成, 19

結論, 85
新機能, 3
要件の概要, 5