

データプロバイダーとは

■ 機能

- データソースとの接続
- クエリの実行

■ 特徴

- 高速なアクセス
- リソースの節約

■ 機能

データプロバイダーは、データベースに接続したり、クエリを実行したりするときに利用します。この際、複数のオブジェクトを組み合わせで使用します。たとえば、データベースとの接続は Connection オブジェクトでおこない、選択クエリや更新クエリの実行は Command オブジェクトでおこないます。また、クエリによって返される結果は、DataReader オブジェクトで読み取ります。

■ 特徴

データプロバイダーが提供する DataReader オブジェクトは、高速に読み取り専用のアクセスができるため、データベースの検索結果を表示するのに適しています。また、DataReader オブジェクトは、1 行ずつ処理するので、メモリには常に 1 件分のデータしか存在しません。そのため、選択クエリの実行によって大量のデータが検索されても、使用されるメモリは最小限ですみます。

標準データプロバイダーの種類			
	対象	プリフィックス	名前空間
SQL Server データプロバイダー	SQL Server 7.0以降	Sql	System.Data.SqlClient
Oracle データプロバイダー	Oracle	Oracle	System.Data.OracleClient
OLE DB データプロバイダー	OLE DBプロバイダーが提供されているデータソース	OleDb	System.Data.OleDb
ODBC データプロバイダー	ODBCドライバーが提供されているデータベース	Odbc	System.Data.Odbc

※Oracle データプロバイダーのサポートは.NET Framework 4 までです。

データプロバイダーは、データソースとのインターフェイスであるため、データソースごとに用意する必要があります。.NET Framework 2.0 以降では、以下の 4 つのデータプロバイダーが含まれています。

■ SQL Server データプロバイダー

SQL Server 7.0 以降のために開発されたデータプロバイダーです。アンマネージコードを利用しないため、OLE DB データプロバイダーを利用するよりも高速です。

■ Oracle データプロバイダー

Oracle DB のために開発されたデータプロバイダーです。.NET Framework 4.5 以降ではサポートされませんので、Oracle 社が提供している ODP.NET (Oracle Data Provider for .NET) を利用します。

■ OLE DB データプロバイダー

OLE DB プロバイダーが提供されているデータソース用のデータプロバイダーです。OLE DB プロバイダーを利用するため、マネージコードとアンマネージコードの境界越えのオーバーヘッドがあります。

■ ODBC データプロバイダー

ODBC ドライバーが提供されているデータソース用のデータプロバイダーです。ODBC ドライバーを利用するため、マネージとアンマネージコードの境界越えのオーバーヘッドがあります。

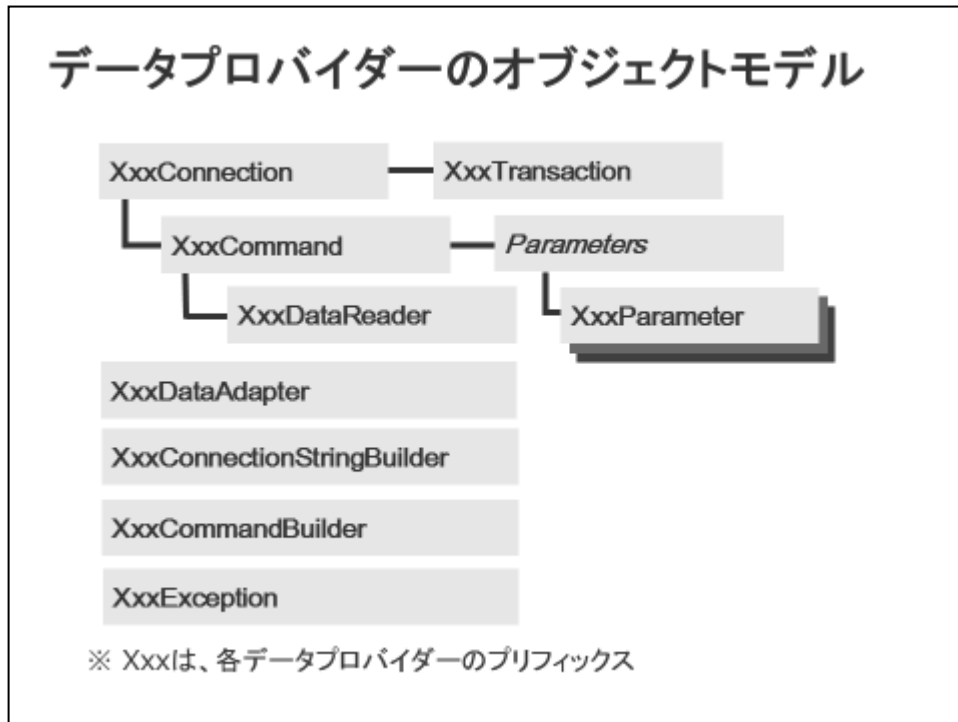
これらの .NET プロバイダー以外にも、さまざまなベンダーやサードパーティから、データプロバイダーが提供されています。

第 2 章 データプロバイダーを利用したデータアクセス

ポイント！ : SQL Server や Oracle には、OLE DB プロバイダーや ODBC ドライバーが提供されているため、OLE DB、ODBC データプロバイダーを利用することもできます。ただし、これらはアンマネージコードのため、アンマネージとマネージの境界越えが起こり、オーバーヘッドが生じます。専用のデータプロバイダーが提供されている場合は、それを利用したほうが、パフォーマンスが向上します。

ポイント！ : データプロバイダーの各クラス名には、プロバイダー毎のプリフィックスがついています。たとえば、Connection オブジェクトのクラス名は、以下のようになります。

データプロバイダーの種類	Connection のクラス名
SQL Server データプロバイダー	SqlConnection
Oracle データプロバイダー	OracleConnection
OLE DB データプロバイダー	OleDbConnection
ODBC データプロバイダー	OdbcConnection



■ Connection

特定のデータソースとの接続機能を提供します。

■ Transaction

データソースのトランザクション機能を利用するときに使用します。(5 章)

■ Command

データソースに対して、コマンドの実行を要求します。SQL クエリやストアドプロシージャを実行するときに使用します。

■ Parameter

SQL クエリやストアドプロシージャの入力パラメーター、出力パラメーター、および戻り値パラメーターを定義するときに使用します。

■ DataReader

データソースから前方向、読み取り専用のデータストリームを取得します。

■ DataAdapter

DataSet オブジェクトとデータソースの橋渡しをします。(3 章)

■ ConnectionStringBuilder

Connection オブジェクトの ConnectionString プロパティが、簡単に設定できます。

■ CommandBuilder

DataAdapter オブジェクトが使用する更新クエリを自動生成します。(3 章)

■ **Exception**

データソースで例外が発生したときに生成されます。

■ データベースとの接続 ■

- 接続の開始と終了
- SQL Server での認証
- 動的プロパティ
- 動的プロパティの利用 –Windows AP-
- 動的プロパティの利用 –Web AP-

接続の開始と終了

- Connection オブジェクトの生成
 - New / new
- 接続文字列の設定
 - ConnectionString プロパティ
- 接続・切断
 - Open メソッド
 - Close メソッド

■ Connection オブジェクトの生成

データソースに接続するためには、Connection オブジェクトを使用します。Connection オブジェクトは New / new で生成します。

■ 接続文字列の設定

データソースに接続するためには、ユーザー名やパスワードなどの情報が必要です。これらの情報は、接続文字列として ConnectionString プロパティに設定します。

接続文字列は、「パラメーター=値」の組み合わせを「;」(セミコロン)で区切って作成します。接続文字列はデータプロバイダーに直接渡される値であるため、データプロバイダーの種類によって必要なパラメーターは異なります。おもなデータプロバイダーの接続文字列の例は、以下のとおりです。

● SQL Server データプロバイダー

```
Data Source=localhost; Initial Catalog=coffeeDB;  
User Id=sa; Password=P@ssw0rd
```

● Oracle データプロバイダー

```
Data Source=localhost; User ID=scott; Password=tiger
```

● OLE DB データプロバイダー

Microsoft OLE DB Provider for SQL Server の場合

```
Provider=SQLOLEDB ; Data Source=localhost;  
Initial Catalog=coffeeDB; User Id=sa; Password=pass
```

Microsoft OLE DB Provider for Oracle の場合

```
Provider=MSDAORA; User Id=scott ; Password=tiger;
```

第2章 データプロバイダーを利用したデータアクセス

```
Data Source=localhost
```

Microsoft Jet 4.0 OLE DB Provider の場合

```
Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\¥NWind.mdb
```

参考:ConnectionString プロパティを設定する代わりに、インスタンスを生成する際の引数として、接続文字列を指定することもできます。

【VB】

```
1. cn = New SqlConnection("Data Source=localhost;" _  
2.                        & "Initial Catalog=coffeeDB;" _  
3.                        & "User Id=sa; Password=pass;")
```

【C#】

```
1. cn = new SqlConnection("Data Source=localhost;" _  
2.                        + "Initial Catalog=coffeeDB;" _  
3.                        + "User Id=sa; Password=pass;");
```

■ 接続・切断

データベースと接続するには Connection オブジェクトの Open メソッドを実行します。データベースを使い終わったら切断します。切断するには Close メソッドを実行します。

【VB】

```
1. Dim cn As New SqlConnection  
2. cn.ConnectionString = "Data Source=localhost;" _  
3.                    & "Initial Catalog=coffeeDB;User Id=sa; Password=pass"  
4. cn.Open()  
5. 'データベースの利用  
6. cn.Close()
```

【C#】

```
1. SqlConnection cn = new SqlConnection();  
2. cn.ConnectionString = "Data Source=localhost;"  
3.                    + "Initial Catalog=coffeeDB;User Id=sa; Password=pass";  
4. cn.Open();  
5. //データベースの利用  
6. cn.Close();
```


(参考) コントロールを利用した接続

- Connection コントロール
 - オブジェクト生成は自動的におこなわれる
- プロパティウィンドウで接続文字列を設定可能
- Open メソッド
- Close メソッド

Connection オブジェクトは、コントロールで作成することもできます。これにより、コードで生成する必要がなくなったり、プロパティをプロパティウィンドウで設定することができるようになったりするため、開発の効率が向上します。

■ Connection コントロールの表示

Connection コントロールは、既定ではツールボックスに表示されないため、ツールボックスを右クリックし、[アイテムの選択]を選びます。

「ツールボックスアイテムの選択」ダイアログボックスが表示されたら、「XxxConnection」を選択します。



ポイント！ : Connection コントロールは、ASP.NET アプリケーションでは表示できません。

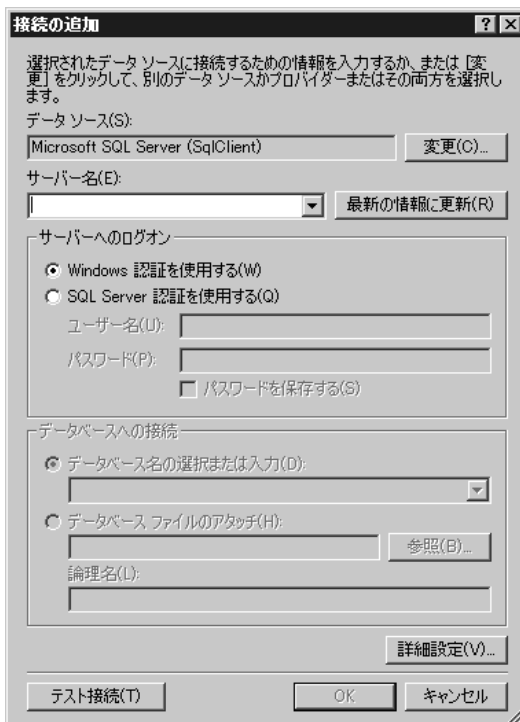
■ Connection コントロールの追加

ツールボックスにコントロールが表示されたら、フォーム上にドラッグして配置します。これにより、Connection オブジェクトが生成されるため、New(new)キーワードを記述する必要がなくなります。



■ 接続文字列の設定

プロパティウィンドウから、ウィザードを使用して、ConnectionString プロパティを設定することができます。



■ 接続・切断

Connection コントロールの Open メソッド / Close メソッドを実行します。