

Visual Basic NET

のツボ

第29回 データ連結

西田 雅昭

NISHIDA, Masaaki

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥TUBOディレクトリに収録しています。

¥OLD

前回作成したサンプルプログラム

¥ADOTEST4

今回作成したサンプルプログラム

¥DATA

サンプルデータベース (Shimei.mdb)

*) 今回のサンプルプログラムは、データファイルを「E:\dotNET\Magazine¥VB21¥Data¥」ディレクトリに配置しているという前提で作成しています。サンプルを実行するには、上記のディレクトリにデータファイルを配置するか、「mprConnectMake」プロシージャの接続文字列を自分の環境に合わせて修正する必要があります。

前回は、データベースの処理を行なうプロジェクトを新たに作成しました。今回は、そのプロジェクトにいろいろな機能を付け加えるとともに、ユーザーに優しいインターフェイスを作ってゆこうと思います。



データ連結とは？

前回まで、いろいろコードをいじってきましたが、連結コントロールに関しては断片的にしか説明していませんでした。そこで、今回は、連結コントロールに関して、きちんとまとめて説明することになります。

Windowsフォーム上に、データソースの内容を表示し、データソースの処理を行なうための、最も簡単な方法は、「データ連結」という機能を利用することです。

VB6.0の時代から連結コントロールを使っていた方は、データ連結という「データの内容（文字列）を表示すること」というイメージを持っているでしょう。

しかしVB.NETのデータ連結は、Win

dowsフォーム上の任意のコントロールに対して、実行時にアクセスできる任意のプロパティを自動的に設定する手段なのです。データの内容（文字列）を表示するには「Text」プロパティをデータソースに連結します。また、コントロールの背景色を変えとか、「Image」コントロールに画像を表示する、などの目的にもデータ連結が使えます。

データ連結には、「単純連結」と「複合連結」という2つの種類があります。これらの違いについて見てゆきましょう。

●単純連結

「単純連結」は、テーブル内の列の値などのひとつのデータ要素にコントロールを連結する機能です。「TextBox」や「Label」にデータを表示する際になどに使います。

この場合、フォームのデザインモードで、プロパティウィンドウの「(Data Bindings)」の「Text」から必要なものを選択することで、データ連結を簡単に作成できます。

これをコードで記述すると、

```
txtCode.DataBindings.Add _
(New Windows.Forms.Binding( _
"Text", mdstShimei, "tbShimei.clCode"))
txtName.DataBindings.Add _
(New Windows.Forms.Binding( _
"Text", mdstShimei, "tbShimei.clName"))
txtFName.DataBindings.Add _
(New Windows.Forms.Binding( _
"Text", mdstShimei, "tbShimei.clFName"))
```

のようになります。

このコードは、「txtCode」テキストボックスの「DataBindings」コレクションに新しい「Binding」オブジェクトを追加しています。

「Binding」オブジェクトの内容はコンストラクタで指定しています。「Binding」クラスのコンストラクタの書式については、表1に掲載しておきます。

これで、この「txtCode」テキストボックスの「Text」プロパティに「mdstShimei」データセットの「tbShimei」テーブルの列「clCode」を連結していることがわかりますね。

先ほど「Text」プロパティ以外との連結もできると言いましたが、データソースもADO.NETデータオブジェクト以外に、配列やコレクションを指定することができます。

● 複合連結

「複合連結」は、複数のデータ要素にコントロールを連結する機能です。複数のデータ要素とは、通常、データ

表1: 「Binding」クラスのコンストラクタの書式

Public Sub New(ByVal propertyName As String, ByVal dataSource As Object, _ ByVal dataMember As String) As Binding		
機能	コントロールのプロパティに、データソースのメンバを単純連結する「Binding」クラスのインスタンスを作成する	
パラメータ	propertyName	連結するコントロールのプロパティ名
	dataSource	データソースを表わすObject
	dataMember	連結先のプロパティまたはリスト

ベース内の複数のレコードである場合が多いでしょう。

「DataGrid」「ListBox」「ComboBox」などのコントロールが複合連結をサポートしています。

今回のサンプルプログラムでは「DataGrid」を使っているので、フォームのデザインモードで、プロパティウィンドウの「DataSource」プロパティと「DataMember」プロパティを設定することで複合連結を作成することができます。

これをコードで書くと、

```
With dgdShimei
.DataSource = mdstShimei
.DataMember = "tbShimei"
End With
```

のようになります。

ここで、データソースとして連結できるADO.NETデータオブジェクトに、どんなものがあるか列挙しておきましょう。

- DataColumn
- DataTable
- DataView
- DataSet
- DataViewManager

本稿ですべてを説明することはできませんので、いろいろ実験してみてください

ださい。



連結を受ける側

では、連結したデータを受け取る側(Windowsフォーム)のメカニズムはどうなっているのでしょうか。

データ連結を持つWindowsフォームには、データソースごとに「CurrencyManager」というオブジェクトがあります。このオブジェクトは、データソースの場所を追跡し、データソースへの連結を監視しています。

フォーム上にデータ連結コントロールが複数あっても、ひとつの「CurrencyManager」オブジェクトを共有します。しかし、データソースが異なると(たとえば2つのテーブルと連結している場合など)、データソースの数だけ「CurrencyManager」オブジェクトを持つこととなります。

Windowsフォームには、複数の「CurrencyManager」オブジェクトを管理するために「BindingContext」オブジェクトがあります。「BindingContext」オブジェクトが、特定のコントロールと特定のデータソースとのデータ連結を管理している「CurrencyManager」を指定してくれるのです。

なお、先ほど説明した「Binding」オ