

Visual Basic NET

のツボ

第26回 ADO.NETを利用した データベース処理 —その6—

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。

¥ADOCODE3
¥DATA
サンプルデータベース (Shimei.mdb)

*) サンプルプログラムは、データファイルを[E:\¥dotNETMagazine¥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。サンプルの実行時には、上記のディレクトリにデータファイルを配置するか、以下の2つを自分の環境に合わせて修正する必要があります。

- ・「btnConnect_Click」プロシージャの接続文字列
- ・「OleDbConnection」コントロール (OleDbConnection1) の「ConnectionString」プロパティ

本連載では、ADO .NETを使ったデータ処理プログラムを作成しています。今回は、データベース内を移動するボタンとコマンドを付け加えることにします。



フォームの改良

まず、前回までのプログラムのフォームにボタンを3つ (表1) 追加し、以前のボタンの位置を少し変更してください。これは、3つのコマンドを生成するためのボタンです。この3つのボタンの「Text」プロパティの設定値はボタンの名前からわかりますね。

さらに、ボタンを8つ (表2) とラベルを2つ (表3)、図1のように配置してください。これは、テーブルの中のレコードを移動するためのボタンです。ボタンのプロパティで重要なのはフォ

表1：コマンド生成用のボタン

	プロパティ設定	
	(Name)	Text
Button1	btnDelete	Delete
Button2	btnInsert	Insert
Button3	btnUpdate	Update

ントを「太字」にしていることです。

なお、ラベルには、テーブルの中にある総行数と、現在参照している行位置を表示します。



データを 移動する処理

まずデータの移動に関するボタンの内容を作成しましょう。

困ったことに、ADO.NETでは、カレントレコードという概念がありません。そこで、Windowsフォーム上のデータ連結についてお話する必要があります。

Visual Basic .NET (以下VB.NET) のデータ連結は、配列やコレクションについてもデータベースと同じように連結処理を行なうことが可能です。

コントロールに連結するデータソースには、必ず「CurrencyManager」オ

表2：ナビゲーション用のボタン

	プロパティ設定			備考
	(Name)	Text	Font(Bold)	
Button1	btnFirst1	<<	True	データセットDsShimei1用 ([Fill1] ボタン)
Button2	btnPrevious1	<		
Button3	btnNext1	>		
Button4	btnLast1	>>		
Button5	btnFirst2	<<		データセットmdstShimei用 ([Fill2] ボタン)
Button6	btnPrevious2	<		
Button7	btnNext2	>		
Button8	btnLast2	>>		

表3：ナビゲーション用のラベル

	プロパティ設定			
	(Name)	BackColor	BorderStyle	TextAlign
Label1	lblPosition1	White	Fixed3D	MiddleCenter
Label2	lblPosition2			

図1：変更後のフォーム

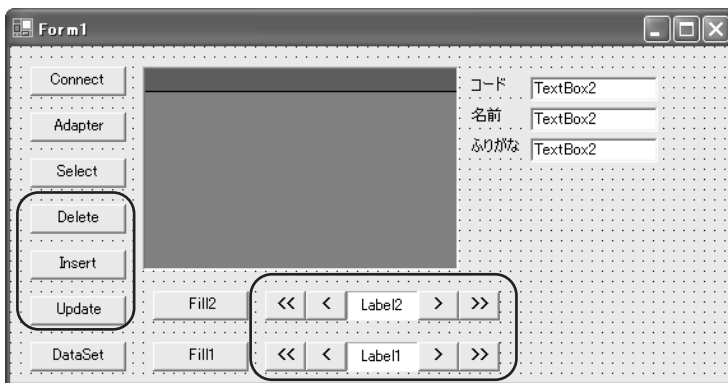
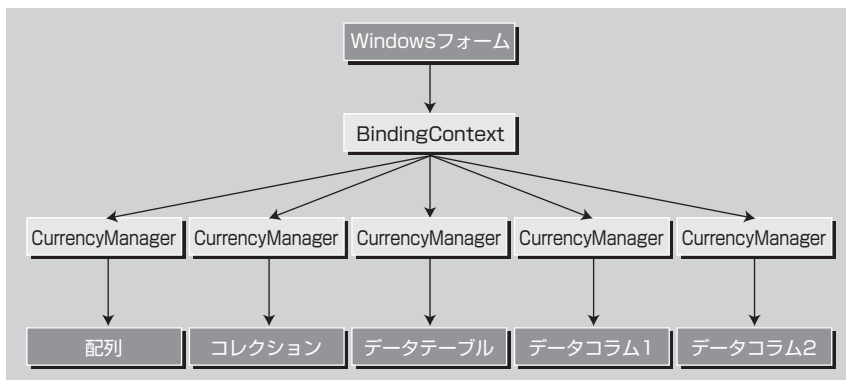


図2：「BindingContext」オブジェクトは、Windowsフォーム上のすべての「CurrencyManager」オブジェクトを管理する



プロジェクトが関連づけられています。このオブジェクトは、データソースとコントロールの連結を監視し、コントロールがどのレコードまたはデータ要素を使用しているかを追跡しています。複数のデータソースがあれば、その数だけ「CurrencyManager」オブジェク

トがあるのです。

「CurrencyManager」オブジェクトには「Position」プロパティがあり、データソース内の現在の位置を示します。

一方Windowsフォーム（またはコンテナコントロール）には、「BindingContext」オブジェクトがあり、フォー

ム上のすべての「CurrencyManager」オブジェクトを管理しています（図2）。

実際のコードでは、「BindingContext」クラスの「Item」プロパティ（省略可能）を使って、

```
BindingContext(mdStShimei, "tbShimei")
'Itemプロパティ省略しない場合
'BindingContext.Item(mdStShimei, "tbShimei")
```

のように記述して「CurrencyManager」オブジェクトを呼び出します。

ヘルプでは、このプロパティの値は「BindingManager」クラスということになっていますが、「BindingManager」クラスも継承された「CurrencyManager」オブジェクトを呼び出すこととなります。

ですから、最初のレコードに移動するボタンの「Click」イベントハンドラには、

```
BindingContext(_
mdstShimei, "tbShimei").Position = 0
```

のように記述すればよいのです。

参考

「Position」プロパティの値が最初や最後のレコードを越える値になってもエラーは発生しません（そのような値にはならないのです）。したがってADOのときのような「EOF」プロパティや「BOF」プロパティのチェックは必要ありません。