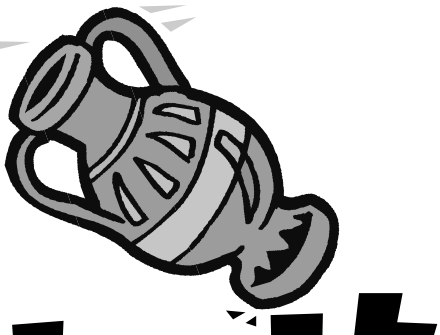
VB.NET

Visual Basic .NETのツボ



第12回 自動読み込み機能の実装と追加／レコード移動用ボタンの機能

西田 雅昭
NISHIDA, Masaaki

筆者はデータフォームウィザードが生成してくれたコードを検討することが、ADO.NETを理解する近道だと考えています。そこで、先月号に引き続き、このコードの検討を行ないます。

前回は［読み込み］ボタンの機能をすべて調べました。これで、データベースの処理をどのように始めるかをご理解いただけたと思います。

今回はまず、［読み込み］ボタンを使う必要がないようにプログラムを修正し、次に［追加］ボタンの機能を調べてみることにします。今回の内容は、前回までに作成したDataBase1プロジェクトを調べたり改造したりします。このDataBase1プロジェクトは付録CD-ROMに収録してありますから、参照してください。

自動実行にする

ところで、考えてみるとプログラムを使う際に、いちいち［読み込み］ボタンをクリックするのは面倒です。DataForm1フォームを開いた際に、自動的にデータベースの処理を始めることができるようにしたいものです。

これは簡単に実現できますね。DataForm1フォームのbtnLoad_Clickイベントハンドラに記述されているコードを、DataForm1_Loadイベントハンドラに移せばよいのです（リスト1）。まず、btnLoad_Clickイベントハンドラのコード部分を選択して、［Ctrl］＋［C］キーを押します。次に、DataForm1フォームのデザインモードでフォームのどこかをダブルクリックすると、コードウィンドウでDataForm1_Loadイベントハンドラを見ることができるので、ここで［Ctrl］＋［V］キーを押せば終了です。

［F5］キーを押してプログラムをデバッグ実行します。開いたメニューフォームで以前に作成した“Bu”という文字が見える小さなボタン（図1）をクリックするとDataForm1フォームが表示されて、［読み込み］ボタンをクリックしなくてもデータを表示することを確認できます（図2）。

デバッグを終了してデザインモードに戻り、DataForm1の［読み込み］ボタン（btnLoad）を消去します。注意しておきますが、デザインモードで［読み込み］ボタンを消去しても、btnLoad

本稿で前提となるもの

OS　Windows XP Professional（SP1）

開発環境　Visual Basic.NET

初級　中級　上級

この記事で解説しているサンプルプログラムは、付録CD-ROMの¥DMAG¥TUBOフォルダ以下に収録しています。

¥DATABASE1：前回までに作成したサンプルプログラム

図1：DataBase1プロジェクトのfrmMenuフォーム

図2：DataForm1フォームにデータが読み込まれている

_Clickイベントハンドラは消えません。コードウィンドウでbtnLoad_Clickイベントハンドラをすべて消去しておいてください。

[追加] ボタンの機能

デザインモードで［追加］ボタン（btnAdd）をダブルクリックすると、コードウィンドウでbtnAdd_Clickイベントハンドラ（リスト2）を見ることができます。

「Try…Catch」構文については、前々回（本誌2003年2月号）に詳しく説明しましたね。VB.NETの「構造化例外処理」と呼ばれるものでした。

Tryブロックの1行目、

リスト1：読み込み処理を自動的に行なう

```
Private Sub DataForm1_Load(ByVal sender As System.Object, _
 ByVal e As System.EventArgs) Handles MyBase.Load

  Try
    'データセットを読み込みます。
    Me.prcLoadDataSet()
  Catch eLoad As System.Exception
    'エラー処理コードをここに追加します。
    'エラーメッセージがある場合は表示します。
    System.Windows.Forms.MessageBox.Show(eLoad.Message)
  End Try
  Me.objdstTest1_PositionChanged()

End Sub
```

リスト2：DataForm1のbtnAdd_Clickイベントハンドラ

```
Private Sub btnAdd_Click(ByVal sender As System.Object, _
 ByVal e As System.EventArgs)Handles btnAdd.Click
  Try
    '現在の編集結果を消去します。
    Me.BindingContext(objdstTest1, "Friend").EndCurrentEdit()
    Me.BindingContext(objdstTest1, "Friend").AddNew()
  Catch eEndEdit As System.Exception
    System.Windows.Forms.MessageBox.Show(eEndEdit.Message)
  End Try
  Me.objdstTest1_PositionChanged()

End Sub
```

```
Me.BindingContext(objdstTest1, _
 "Friend").EndCurrentEdit()
```

を見てみましょう。「Me」は自分自身、すなわちDataForm1フォームを指すことも、もうおわかりですね。では、続く「BindingContext」は何でしょう。ここにカーソルを置いて［F1］キーを押すと、なんと「BindingManagerBaseメンバ」のヘルプが開きました。便利なダイナミックヘルプ機能ですが間違いもあるようです。どうやらMSDNではヘルプの訂正が行なわれるようですから、それに期待しましょう。

BindingContextプロパティ

私の記憶では、「BindingContext」はFormのメンバであるはずです。ヘルプで「Formメンバ」を見てみると、BindingContextはプロパティであることがわかります。さらにFormメンバのヘルプで「BindingContext」をクリックすると、「Control.BindingContextプロパティ」のヘルプが開きました（図3）。ヘルプの先頭には、

「コントロールのBindingContextを取得または設定します。」

と書いてあります。