



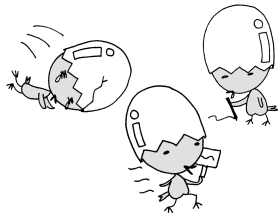
作って学ぶ
VISUAL BASIC .NET

瀬戸 遥

SETO, Haruka

<http://www.big.or.jp/~seto/>

<http://hp.vector.co.jp/authors/VA006682/>



MDIフォームを 使ってみようーその3ー

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¥TAMAHYOディレクトリに収録しています。

¥MDI_3
今回のサンプル



バグレポート

いきなりですが、前回作成したコードにミスがありました。ごめんなさい。

ミスがあったのは、開いたファイルをメニューに組み込む処理の部分です。前回のプログラムでは、ファイルを開くダイアログボックスの [キャンセル] ボタンを押しても、そのまま空白のメニューを組み込んでしまいます。

Sub プロシージャ「addmenu」を呼び出している位置が、以下のよう
にIfステートメントの外にあるため、ダイアログボックスで [OK] ボタンを押したとき以外にも、メニューの追加処理を実行してしまうのです。

```
Private Sub MenuItem3_Click(略) _
    Handles MenuItem3.Click
    (略)
    If Me.OpenFileDialog1.ShowDialog() _
```

```
= DialogResult.OK Then
(略)
End If
addmenu(Fname) ←ここで呼び出し
End Sub ている
```

この部分を以下のように修正してください。

```
Private Sub MenuItem3_Click(略) _
    Handles MenuItem3.Click
    (略)
    If Me.OpenFileDialog1.ShowDialog() _
        = DialogResult.OK Then
        (略)
        addmenu(Fname) ←Ifブロックの中
    End If 呼び出す
End Sub
```

正しくは、このように、Sub プロシージャ「addmenu」の呼び出しをIfステートメントの中に入れ、ダイアログボックスの [OK] ボタンを押したときにだけメニュー項目の追加を実行するようにしないと
いけません。



MDIフォームをパワーアップ！

さて、気を取り直して、今月は前回作成したMDIアプリケーションの機能拡張をさらに進めていきます。

組み込む機能は、次の3つです。

- ①ファイル名のメニュー表示を5つまでにする
- ②ファイル名のメニューをレジストリに保存する
- ③レジストリからファイル名を取得しメニューに登録する

以降でこれらの機能を実装していきましょう。



ファイル名だらけはいけません

前回のサンプルでは、ファイルを開くとそのファイル名をフルパスで「ファイル」メニューに追加する処理を組み込みました。

これはこれで便利な機能ですが、このままほうっておくとメニューの数が膨大なものになってしまいます。

そこで、メニューに登録する数を5つに制限し、新しくファイルを開くとメニューの最後尾に追加し、一番上の古いファイル名のメニュー項目を削除するようにします。

まず、MDIフォーム (MainForm.vb) に、追加したメニュー項目の個数を数えておく変数を、モジュールレベルで宣言します。この宣言は、以下のように、前回作成したLoad イベントプロシージャの前に記述してください。

```
Dim MenuNum As Byte = 0
Private Sub MainForm_Load(略) Handles MyBase.Load
    (略)
End Sub
```

次に、前回作成したSub プロシージャ「addmenu」の処理内容を変更します。

まず、メニュー項目の個数を示す変数MenuNumの値

をひとつ増やします。

```
Private Sub addmenu(ByVal fname As String)
    MenuNum += 1
```

続いて、この値が“5”より大きくなったら、ファイル名のメニュー項目の一番上を削除する処理を記述します。

MenuItem オブジェクトを削除するには、MenuItems コレクションのRemove または RemoveAt メソッドを使用します。Remove メソッドの場合は引数に「メニューの項目名」を指定し、RemoveAt メソッドの場合は「メニュー項目のインデックス番号」を指定します。

ここでは、「ファイル」メニューのメニュー項目「終了」の下にある区分線のすぐ下のメニュー項目を削除するので、インデックス番号を指定できる RemoveAt メソッドを使うことにします。

```
With MenuItem1
    If MenuNum > 5 Then
        .MenuItems.RemoveAt(7)
        MenuNum = 5
    End If
```

このコードでは削除するメニュー項目の位置として、“7”を指定しています。というのは、インデックス番号は先頭が「0」から始まり、区分線もその数に数えるからです。つまり、「終了」メニューの下にある区分線はインデックス番号が「6」になり、削除するメニュー項目は「7」になるというわけです。

メニュー項目を削除したら、前回作成したAdd メソッドをそのまま実行します。

```
.MenuItems.Add(fname, _
    New EventHandler(AddressOf addmenu_Click))
End With
End Sub
```

これで、5つ以上ファイルを開くと、自動的に先頭にあるファイル名がメニューから削除され、開いたファイル名がメニュー項目の最後尾に追加されるようになります (図1)。