

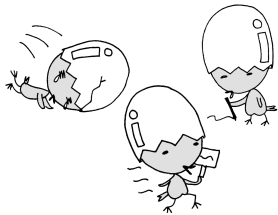
作って学ぶ
VISUAL BASIC .NET

瀬戸 遥

SETO, Haruka

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

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



第15回 MDIフォームを使ってみよう

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
今回のサンプル



ところで MDIってなに？

本誌2004年6月号と7月号でRichTextBoxコントロールを使って、ワープロソフトを作成しました。このときは、「ひとつのフォームからひとつの文書を編集」するように作成しました。このようなイ

ンターフェイスをSDI (Single Document Interface) と呼びます。しかし、市販のワープロソフトや表計算ソフトなどでは、「ひとつのアプリケーションから複数のウィンドウを開いて、文書やワークシートを編集」できるようになっています。このようなインターフェイスをMDI (Multi Document Interface)

図1：作成するアプリケーション。複数のテキストファイルの編集やJPEGファイルの表示を実行できる



と呼びます。Visual Basic .NETに実装されているフォームも、実はこのMDIという機能を使うことができます。

そこで、今回はMDI機能を使って、複数のテキストファイルの編集や画像表示ができるアプリケーションを作成してみます (図1)。



さらに詳しくMDI

フォームのMDI機能を使うと、図1のようにフォーム内に複数のフォームを表示させることができます。

このような形態のフォームを**MDIフォーム**と呼び、プログラムのユーザーインターフェイスを担当しているフォームを**親フォーム**、親フォーム内に表示するフォームを**子フォーム**と呼びます。そして、ひとつのプログラムに親フォームはひとつ、子フォームはいくつも持たせることができます。

フォームをMDIの親フォームにするには、フォームの**IsMdiContainer**プロパティを使用します。プロパティウィンドウからこのプロパティの値を“True”に設定すると、そのフォームはMDIの親フォームとなります。

親フォーム内に表示する子フォームを設定するには、プロジェクトにフォームを追加し、**MdiParent**プロパティで親フォームを指定します。この操作はプロパティウィンドウではなく、コードから行ないます。設定値は、親フォームを参照するFormオブジェクトです。



こんな機能を組み込みます

MDIフォームを使うテクニックはとて多く、用途もいろいろ考えられるだけに、実装したい機能は数多くありますが、一度に多くの機能を組み込んで混乱するだけです。そこで、今回は、MDIを使うための基本的な機能のみを実装することにします。

機能1：子フォームにテキストデータを入力/表示を行なう

機能2：メニューを使って、新規文書の作成とデータの表示 (ファイルの読み込み) を行なう。表示できるファイルは、テキストファイルとJPEGファイル

機能3：「ウィンドウ」メニューで、子フォームのアクティブウィンドウの切り替えを行なう

テキストデータの操作はRichTextBoxコントロールで行ない、JPEG画像の表示はPictureBoxコントロールで行ないます。データを表示する子フォームと同様に、Rich TextBoxコントロールとPictureBoxコントロールもその都度コードから作成し、DockプロパティとControlsプロパティを使ってフォームにドッキングさせます。



最初に親フォームを作りましょ

では、早速プログラムのベースとなる親フォームを作成しましょう。プロジェクト名は「MDIアプリケーション」とします。

新しいプロジェクトを作成したら、Form1.vbファイルのファイル名と、フォームのNameプロパティ (オブジェクト名)、IsMdiContainerプロパティの値を変更します。

プロパティ	設定値
ファイル名	MainForm
(Name)	MainForm
IsMdiContainer	True
Text	MDIアプリケーション

フォームのオブジェクト名を「Form1」から「MainForm」に変更したのは、親フォームと子フォームを区別できるようにするためです。

また、IsMdiContainerプロパティを“True”にすると、図2のようにフォームの背景色が黒くなり、少しくぼんだような形状に変わります。なんとなく、どこかのアプリケーションでみたような形になりましたね。

フォームのオブジェクト名を変更したら、プロジェクトのスタートアップフォームの設定も変更しておきます。