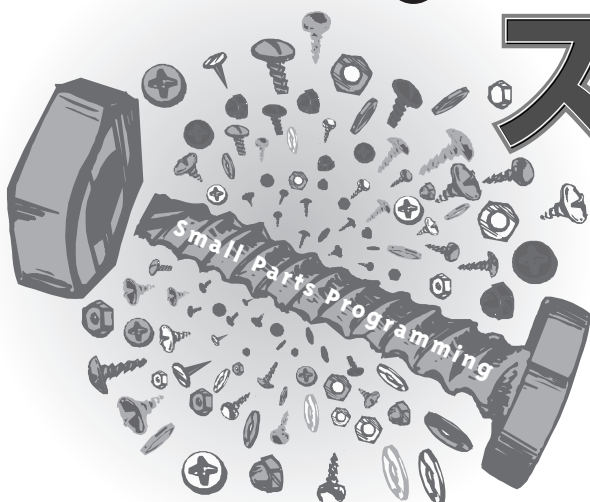


# VB.NET

コンポーネント活用講座  
～ 教養課程 ～



# スモールパーツ プログラミング

第22回

小森 大輔 KOMORI, Daisuke

## リソース活用法 [実践編]

### 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¥SMALLディレクトリに収録しています。

¥RESOURCESAMPLE1  
カルチャー別のリソースを使用するサンプル

¥RESOURCESAMPLE2  
Visual Studio .NETのIDEを使ったリソースの利用

本連載は、Visual Basic .NET (以下VB.NET) の標準コントロールにスポットを当て、とことん使い込んで、その機能や役割を探ってゆきます。

今回は前回と同様に、.resxファイルとResource ファイルといったリソースファイルの作成方法や、ResXResource ReaderクラスやResourceManagerクラスを使用した、リソースファイル内のリソースの基本的な取得方法について説明しました。今回は実践編として、カルチャー別のリソースの配置方法や、Visual Studio .NET (以下VS.NET) のIDEを使ったリソースの利用方法などについて見てゆきたいと思います。

### リソースファイルの作成



まずは前回の復習も兼ねて、今回のサンプルで使用するリソースファイルを作成することから始めてみましょう。

リソースの種類としてはResXファイル形式 (.resx) とResource ファイル形式 (.resources) があり、

- ・ ResXファイル形式の場合：コード上からResXResourceWriterクラスのAddResourceメソッド
- ・ Resourceファイル形式の場合：リソースファイルジェネレータ(Resgen.exe) を使用して.resxファイルを変換、またはコード上からResourceWriterクラスのAddResourceメソッド

を使用して、イメージなどから直接リソースファイルを作成するんですね。

今回は「resourcesファイルを作成する」ことにし、リソースファイルには日本の国旗のイメージリソースと、このイメージを説明する「日本」という文字列リソースを含めることにします。

リスト1は、実行ファイルのディレクトリ配下に存在する¥Resources

### リスト1：リソースファイルの作成

```
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    Try
        ' ファイル名を指定してResourceWriterのインスタンスを作成
        Dim rw As ResourceWriter = New ResourceWriter(_
            ".\Resources\ResourceSample1.resources")

        With rw
            ' イメージリソースを書き込み
            .AddResource("Flag", _
                Image.FromFile(".\Resources/JapanFlag.jpg"))

            ' テキストリソースを書き込み
            .AddResource("LabelText", "日本")

            ' リソースをファイルとして出力
            .Generate()

            ' ResourceWriterを閉じる
            .Close()
        End With

        ' ファイルの出力に成功したらメッセージを表示
        MessageBox.Show("Created Resources File!")

    Catch ex As Exception
        ' 例外発生時のメッセージ
        MessageBox.Show(ex.Message.ToString)
    End Try
End Sub
```

ディレクトリへリソースファイルを作成するサンプルです。

まず出力先のディレクトリとリソースファイル名を指定してResourceWriterのインスタンスを作成します。

次に、AddResourceメソッドで“Flag”という名前を付け、出力先と同じディレクトリに存在するイメージファイルをリソースとして追加します。そして、同じように“LabelText”という名前を付けて文字列リソースを追加し、Generateメソッドでファイルへ出力すればリソースファイルの出来上がりです。

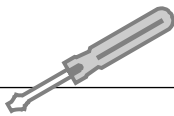
法についても復習しておきましょう。

リソースファイルからリソースを取り出すには、いろいろなクラスを使用する方法がありますが、今回は代表的な「ResourceManagerクラス」を使用します。

リスト2は、フォームのButtonコントロールのクリックイベントで、PictureBoxコントロールやLabelコントロールへ、リソースファイル内のリソースを表示するサンプルです。

このサンプルではまず、ResourceManagerクラスのCreateFileBasedResourceManagerメソッドで、ResourceManagerのインスタンスを作成しています。このメソッドに指定しているのは、リソースファイルのファイル名とファイルの位置（実行ファイルからの相対位置で指定）です。このサンプルでは、メインアセンブリのディレクトリ配下にあるResourcesディレクトリを指定し

## リソースファイル内のリソースを使用する



作成したリソースファイル内の“リソース”の使用方法

### リスト2：リソースファイル内のリソースの使用

```
Protected Sub SetInitializeValue()
    ' 各コントロールのプロパティを指定
    With Label1
        .Text = ""
        .Font = New Font(Font.FontFamily, 14, FontStyle.Bold)
    End With

    PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage

    Button1.Text = "CreateResources"
    Button2.Text = "ReadResources"
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button2.Click
    ' リソースファイルからResourceManagerのインスタンスを作成
    Dim rm As ResourceManager = _
        ResourceManager.CreateFileBasedResourceManager(_
            "ResourceSample1", ".\Resources", Nothing)

    ' PictureBoxにリソース内のイメージを表示
    PictureBox1.Image = CType(rm.GetObject("Flag"), Image)

    ' Labelにリソース内のテキストを表示
    Label1.Text = rm.GetString("LabelText")
End Sub
```