

こんなときVB.NETでは どうする？

VB6.0でやっていた処理をVB.NETで実現する
実践的なコードを紹介

大澤 文孝

OSAWA, Fumitaka

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
Excel 2003

Level



Samples

はじめに

VB.NETでは、VB6.0で関数やコンポーネントを使っていた処理を、.NET Frameworkのクラスライブラリを使った処理に置き換える必要があります。

とはいえ.NET Frameworkのクラスライブラリは膨大で、最初のうちはどこから手を付けてよいのかわからないこともあります。

そこで本稿では、アプリケーションでよく必要となる処理を、VB.NETではどのように実装すれば良いのか、その具体的なコードを紹介します。

ファイルの読み書き

VB6.0では、ファイルを読み書きするのに、FileSystemObjectオブジェクトを使っていました。

それに対してVB.NETでは、System.IO名前空間に存在するクラス群を使います。

```
Imports System.IO
```

テキストファイルの読み取り

まずは、テキストファイルを読み取る方法から説明します。

テキストファイルを読み取るには、StreamReaderオブジェクトを使います(リスト1)。

手順1 ▶ StreamReaderオブジェクトを作る

まずはNewキーワードを使って、StreamReaderオブジェクトを作ります。

このときコンストラクタの引数に、読み取りたいファイル名を渡します。

```
Dim reader As New StreamReader(ファイル名)
```

手順2 ▶ 読み取る

StreamReaderオブジェクトのメソッドを使ってファイルを読み取ります。

たとえば、ReadToEndメソッドを使うと、全行を読み取れます。

```
Dim load As String  
load = reader.ReadToEnd()
```

全行読み取るのではなく、1行ずつ読み取るには、ReadLineメソッドを使

リスト1: テキストファイルの読み取り

```
Dim load As String
Dim reader As StreamReader

Try
    reader = New StreamReader(ファイル名)
    load = reader.ReadToEnd()
Catch ex As IOException
    Debug.WriteLine("エラー:" & ex.Message)
Finally
    If Not IsNothing(reader) Then
        reader.Close()
    End If
End Try
```

います。

Peekメソッドを使うと、あと何文字読み取れるのかを参照できるため、次のようなループ処理で、全行取得することもできます。

```
While reader.Peek() >= 0
    load = load & reader.ReadLine()
End While
```

手順3▶ ファイルを閉じる

ファイルを閉じるには、Closeメソッドを呼び出します。

```
reader.Close()
```

ファイルの読み取りに失敗すると、IOException例外が発生します。そこでリスト1では、Try…Catchで括った例外処理を実装しています。

文字のエンコード

.NET Frameworkでは、文字コードのデフォルトが「UTF-8」であるため、リスト1のコードでは、シフトJISコードなど、他の文字コードで書かれたファイルを読めません。

他の文字コードで書かれたファイルを読むには、明示的に文字コードの種

類をStreamReaderのコンストラクタに指定します。

文字コードのエンコードはSystem.Text.Encodingオブジェクトで表現されます。

任意の文字コードに対応するSystem.Text.Encodingオブジェクトを取得するには、次のようにします。

```
System.Text.Encoding.GetEncoding(_
    文字コード)
```

このオブジェクトをStreamReaderのコンストラクタの第2引数に渡すと、その文字コードで読み取れるようになります。

たとえば、シフトJISコードで読み取るのであれば、次のようにします。

```
reader = New StreamReader(_
    ファイル名, _
    System.Text.Encoding.GetEncoding(_
    "Shift_JIS"))
```

テキストファイルの書き込み

テキストファイルを書き込むには、StreamWriterオブジェクトを使います。その処理は、リスト2のようになります。

StreamWriterオブジェクトもStream

リスト2: テキストファイルの書き込み

```
Dim writeText As String = "書き込みたいデータ"
Dim writer As StreamWriter

Try
    writer = New StreamWriter(ファイル名)
    writer.Write(writeText)
Catch ex As IOException
    Debug.WriteLine("エラー:" & ex.Message)
Finally
    If Not IsNothing(writer) Then
        writer.Close()
    End If
End Try
```

Readerオブジェクトと同様に、コンストラクタに開きたいファイル名を渡してファイルを開きます。

```
Dim writer As New StreamWriter(ファイル名)
```

ファイルに書き込むには、リスト2に示したように、Writeメソッドを使います。

```
writer.Write(書き込みたい文字列)
```

ちなみにWriteLineメソッドを使うと、後ろに改行を付けて書き込むこともできます。

ファイルの書き込み操作が終わったら、Closeメソッドを呼び出します。

```
writer.Close()
```

StreamWriterオブジェクトもStreamReaderオブジェクトと同様、デフォルトの文字コードは「UTF-8」です。

シフトJISコードで書き込みたい場合には、次のようにします。

```
writer = New StreamWriter(_
    ファイル名, _
    False, _
    System.Text.Encoding.GetEncoding(_
    "Shift_JIS"))
```