

# ロジックを分離して DLL化するには

どんなアプリケーションでも使える  
変換ルーチンの作成法

大澤 文孝 OSAWA, Fumitaka

## Level

1 2 3 4 5

## Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- Excel
- ASP.NET
- Other:

## 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、  
<http://www.shoeisha.com/mag/windev/>  
からダウンロード可能です。

## はじめに

前の記事「文字コードに勝つ！」では、さまざまな文字コードで書かれたテキストファイルを読み書きする方法を説明しました。

文字コード変換を必要とする場面は、Windowsアプリケーションだけではなくありません。Webアプリケーションでも文字コード変換が必要になることがあります。またファイルではなくて、ネットワークから取得したデータ列を変換したいこともあるでしょう。

このように文字コードの変換処理は頻用されるため、処理をクラスにしてDLL化し、ほかのアプリケーションか

らも使えるようにすると便利です。

そこで本稿では、前稿で作成したテキストエディタの文字コード変換ロジック部分を分離して、DLLにする方法を説明します。

## DLLを作るには

DLLは「クラスライブラリ」というプロジェクトとして構成します。

その手順は、次のようになります。  
ソリューションに新しいプロジェクトを追加する

ソリューションエクスプローラでソリューションを右クリックし、「追加」-「新しいプロジェクト」を選んでプロジェクトを追加します(図1)。

### クラスライブラリを追加する

クラスライブラリとしてプロジェクトを追加します。

プロジェクト名は任意でかまいませんが、ここでは「KanjiLib」という名前を付けることにします(図2)。

プロジェクト名は、作成されるDLLファイル名にもなります<sup>[注1]</sup>。

### クラス名を変更する

先に作成したプロジェクトのなかに「Class1」というクラスができるので、適当な名前に変更します。

ここでは、「Convert」という名前に変更することにします。「ファイル名」と「ソースのクラス名」の2箇所を変更することになります(図3)。

注1) 作成されるDLLファイル名は、あとでプロジェクトのプロパティの「アセンブリ名」から変更することもできます。

## 汎用的な設計を考える

このようにして作成したConvertクラスに、前稿で実装した文字コード変換処理を実装すれば、文字コード変換処理をDLLファイルへと分離し、他のアプリケーションからも使えるようになります。

とはいえ、前稿で実装したコードを、そのままConvertクラスに移すのは、うまくありません。

たとえば前稿では、ダイアログボックスを使ってファイルを開く処理をしているので、コードをそのまま移しただけでは、ダイアログボックスではない方式でファイルを指定したいときに使えなくなってしまいます。

そこでクラスとして実装する場合には、「どんな値を与えて、どんな値が戻ってくると使いやすいのか」という汎用的な設計が求められます<sup>[注2]</sup>。

## 何を与えると何が出てくるのか

クラスを設計するときには、まず、「何を与えると何が出てくるのか」を考えます。すなわち、「どんな引数を渡すと、どんな処理が行なわれ、結果としてどんな戻り値が得られるのか」を考えます。そうすることで、クラスに実装すべきメソッドが決まります。

今回作成するConvertクラスは、文字コード変換をするのが目的です。

注2) どのようなクラス構成が望ましいのかは、想定される使われ方によります。以下の説明は、あくまでもクラスの実装例としてとらえてください。

図1：ソリューションに新しいプロジェクトを追加する

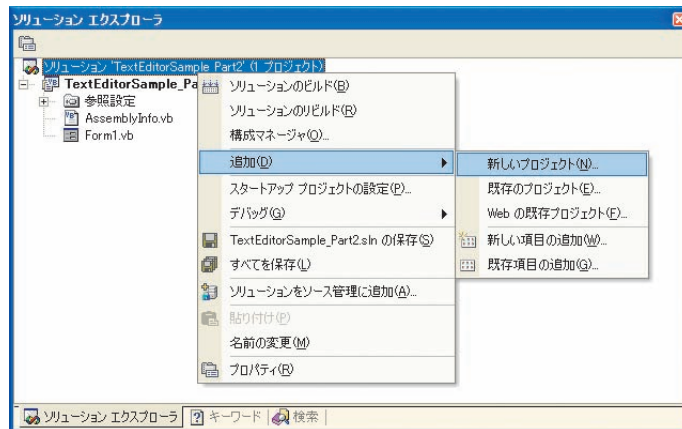


図2：クラスライブラリを追加する

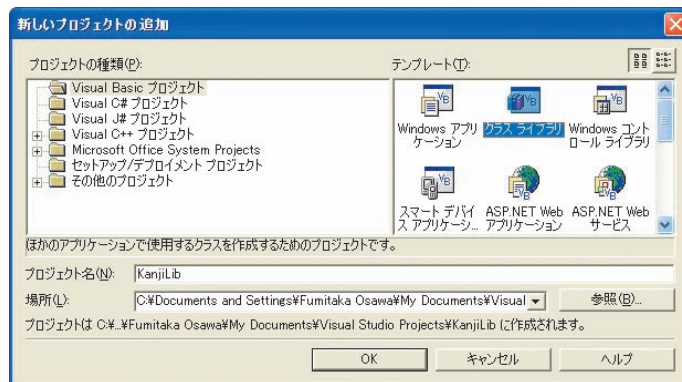


図3：クラス名を変更する

