

# Visual Basic .NETで作る カスタムコンポーネント

新連載

## 標準コントロールの拡張

ウェブデ・ネット有限会社  
黒川 洋二 *KUROKAWA, Youji*  
<http://www.webde.net/>

### 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¥NETARCH\_CTLディレクトリに収録しています。

¥SAMPLE01.SLN  
カスタムコントロール/動作確認用プロジェクトを含むソリューションファイル

### はじめに

プログラムを実装していると、“以前も同じように実装したような……”なんて思って以前書いたソースを探してしまうことがあります。こんなとき私はコンポーネントをカスタマイズしたり、時には新たなコンポーネントを作成したり、クラスにしたりして、できるだけ手間を省いています。

部品化するメリットは多数ありますが、実装量削減、テスト不要（バグ数減少）などを享受できます。

今回から、カスタムコントロールや、オリジナルコンポーネントを作成する方法を数回に分けてご紹介します。

### .NETのコンポーネントとコントロールの定義

たとえば、Borland社のDelphiでは、ボタンやテキストボックスなどユーザーインターフェイス（UI）に使用する部品からデータベースに接続する部品まですべてを「コンポーネント」と定

義します。しかし、.NETでは、ある特定のクラスの基底クラスかつUIの部品を「コントロール」とも言います。

では、コンポーネントとコントロールはどのように違うのでしょうか。

### コンポーネントとは？

.NET Frameworkにおけるコンポーネントとは、

- System.ComponentModel.IComponent インターフェイスを実装するクラス
- IComponent インターフェイスを実装するクラスから直接または間接に派生したクラス

のいずれかを指します。

プログラミングで「コンポーネント」というと、一般的に、「再利用可能で、ほかのオブジェクトと対話できるオブジェクト」という意味があります。.NET Frameworkのコンポーネントは、この一般的な要件を満たすだけでなく、外部リソースの制御やデザイン時サポート

などの機能も備えています。

## コントロールとは？

コントロールとは、「UI機能を提供する（または有効化する）コンポーネント」です。.NET Frameworkには、コントロールの基本クラスが2種類用意されています。

- ①クライアント側のWindows フォームコントロールの基本クラス
- ②ASP.NETサーバーコントロールの基本クラス

これらの基本クラスは、System.Windows.Forms.Control および System.Web.UI.Control です。.NET Framework クラスタライブラリのすべてのコントロールは、これら2つのクラスから直接または間接的に派生しています。

System.Windows.Forms.Controlは、System.ComponentModel.Componentから派生し、それ自身でUI機能を備えています。System.Web.UI.ControlはIComponent インターフェイスを実装しており、UI機能を簡単に追加できるようにする環境を提供します。

## カスタムコントロールの作成方法

今回作成するコントロールは、Windows フォームコントロールです。主に、カスタムコントロールを作成するには次の3つの方法があります。

- 方法 1** 既存のコントロールを継承する
- 方法 2** 複数のコントロールを複合する
- 方法 3** 独自のコントロールを作成する

今回は、これら方法の中からもっとも簡単にカスタムコントロールを作成することができる「既存のコントロールを継承する」という方法を用いて解説してゆきます。この方法は、既存のコントロールに新たにプロパティ、メソッド、イベントなどの機能を追加したいときや既存機能の動

作を変更したいときに便利です。

たとえば、テキストボックスの入力内容を数値のみにしたい場合や、文字列のみにしたい場合は、テキストボックスのKeyDown イベントもしくは、そのイベントを実行するメソッドに数値（または文字列）のみ入力を許可するように実装すればいいのです。また、テキストボックスのフォーマットを独自に定義したい場合は、GotFocus イベントにフォーマットを解除する処理を実装し、Leave イベントまたはLostFocus イベントなどにフォーマットを設定する処理を実装すれば、テキストは自動的にフォーマットされます。

## TextBoxを拡張する

今回は、[Tab] キーを押下したときにフォーカス移動を行なう動作を、「任意のキーを利用して、指定したコントロールへフォーカス移動できる」ように変更し、その機能をTextBox コントロールに追加してみましょう。

まずは、「Windows コントロールライブラリ」のプロジェクトのテンプレートを選択して「CustomCtrls」という名前を付けて（図1）、プロジェクトを新規作成します。

このプロジェクトはクラスライブラリプロジェクト（DLLを生成します）なので、デフォルトで「UserController1.vb」という名前のファイルが自動作成されています。このファイル名を「Ch01TextBox.vb」に変更し、コードを表示して次のようにコードを修正します。

図1：「Windows コントロールライブラリ」プロジェクトを作成

