

# Visual Basic .NETで作る カスタムコンポーネント

第7回

## Webカスタムコントロール

ウェブデ・ネット有限公司  
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¥CONTROLディレクトリに収録しています。

- ¥DB  
テストデータ (Sample.mdb, User.mdb)
- ¥WEBCUSTOMCONTROL  
今回解説したカスタムコントロールのプロジェクト
- ¥TESTWEBCUSTOMCONTROL  
コントロールの動作確認用プロジェクト

### はじめに

今月は、「System.Web.UI.Control」をベースにした独自のコントロールを作成します。Tableコントロールを使用してDataTableオブジェクトの内容を表示する、DataGridのようなコントロールだと考えてください。また、後半では、前回作成したWebユーザーコントロールを、複合カスタムコントロールに作り変えてみたいと思います。

### ユーザーコントロールとの違い

まず、ユーザーコントロールとカスタムコントロールの違いについて考えてみましょう。

ユーザーコントロールは、実行するまでコントロールの状態を確認することができません。対して、カスタムコントロールは、デザイン時にレンダリングされてコントロールの状態を確認することができます。

ユーザーコントロールではプロジェ

クト内にユーザーコントロール (.ascx)を展開しなければなりませんでしたが、カスタムコントロールではVisual Studio .NETのツールボックスに登録することもできます。

カスタムコントロールには、大きく分けて3つの種類があります。

**Type1**：既存のWebコントロールに新しいプロパティ、メソッド、イベントなどを追加したり、元の動作を変更したカスタムコントロール。この場合、拡張元のコントロールを継承してカスタムコントロールを作成します。

**Type2**：既存のWebコントロールを拡張しないカスタムコントロール。この場合、Controlクラス(System.Web.UI.Control)または、WebControlクラス(System.Web.UI.WebControls.\*)の必要なクラスを使用して作成します。

**Type3**：TextBoxやButtonなど、複数のコントロールからなるカスタムコントロール。この場合、Controlクラス(System.Web.UI.Control)を継承して作成します。このようなカスタムコントロールは、「コンポジット

コントロール」とも呼ばれます。

今回は、Type2とType3のカスタムコントロールを作成します。

カスタムコントロールを作成するのはユーザーコントロールほど簡単ではないかもしれませんが、カスタムコントロールには、柔軟性や高度な動的出力ができるなどの利点があります。

## Tableを使用した カスタムコントロール

最初に、既存のWebコントロールを拡張しない「System.Web.UI.Control」を継承したカスタムコントロールを作成してみましょう。

WebフォームのTableコントロールは、データの内容を表示するのにとても便利ですが、DataGridとは違ってDataSourceプロパティを持っていません。そのため、動的にデータを表示したいときには、毎回、TableコントロールにTableRowコントロールとTableCellコントロールを設定しなくてはならないので面倒です。

そこで、DataTableを定義して値を設定し、そのDataTableをTableコントロールに設定して簡単にデータを表示できるようにします。

このカスタムコントロールの使用方法は、以下のとおりです。

- ① DataTableオブジェクトを新たに作成して、Tableコントロールに追加した「DataSource」プロパティに設定する
- ② Controlのオーバーライドメソッドの「DataBind」メソッドを実行して、内部オブジェクトのTableに展開する
- ③ レンダリング処理用オーバーライドメソッド「Render」を実行して、Tableの内容をレンダリングする

まず、カスタムコントロール作成用の「Webコントロールライブラリ」プロジェクトを「WebCustomControl」という名前で作成します。作成したプロジェクトに「Webカスタムコントロール」を追加し<sup>注1)</sup>、ファイル名を「TableView

図1：Webカスタムコントロールの追加



.vb」にします (図1)。

カスタムコントロールの名前は、「TableView」とします。

## 実装するプロパティとメソッド

カスタムコントロール「TableView」は、前述したように「DataSource」プロパティにDataTableを想定したオブジェクトを設定して、DataBindメソッドでDataTableの内容を展開し、Tableの内容をHtmlTextWriterで出力する、という仕様になります。

DataTableの内容を展開する前にBoolean型の変数「HasSeqColumn」を“True”に設定しておけば、Tableの左端の項目を連番項目として出力します。

その他にColor型のプロパティ「FixedBackColor」と「FixedForeColor」を実装しており、これらにColorを設定すれば、列見出し、行見出しの色設定ができます。

「TableView」に定義したプロパティを表1に、定義またはオーバーライドしたメソッドを表2に示します。

「DataSource」プロパティに設定された値は、「SetDataSourceToDataTable」メソッドを実行することで内部のDataTableメンバにキャストされ設定されます。今回はDataTableだけを設定対象にしましたが、カスタマイズすればDataGridViewやDataRowなども設定対象にできるので、興味のある方は試してみてください。

「CreateTable」メソッド内では、前述の変数「HasSeqColumn」の値を判定して連番項目を作成しています (リスト1)。

注1) クラスファイルの追加でも問題ありません。