

Visual Studio .NETで  
究めるコンポーネント開発

## 第2回

## イベントハンドラの登録と独自イベントの作成

株式会社コムラッド  
辻慶 TSUJI, Kei  
<http://www.comrade.co.jp/>

## Technology Tools

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

## Level



## Samples

・本稿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¥NETARCHO1ディレクトリに収録しています。

¥DIGITEXTFORM

「イベントハンドラの登録」サンプル

¥EXTENDEDNUMERICUPDOWN

「独自イベントの作成」サンプル

## はじめに

今回は、TextBoxコントロールを拡張し、数字以外の入力を受け付けないコントロールを作成しながら、プロパティやメソッドのオーバーライドによる処理の変更と、イベントの処理に関して簡単に触れました。今回は、より詳しくイベントに関して説明してゆくことにします。

今回は最初にイベントハンドラの登録に関して説明し、その後独自イベントの実装および使用方法を説明します。

## イベントハンドラの登録

イベントを処理するメソッドのことを「イベントハンドラ」といいます。TextBoxコントロールのKeyPressイベントなら、前回のよう、派生クラスを作成し、OnKeyPressメソッドをオーバーライドすることができます。しかし、派生クラスが作れないSealedクラスのコントロールや、処理を記述したいイ

ベントに対応するOnXXXXメソッドがないなど、オーバーライドできない場合には、今回説明するイベントハンドラの登録を使用することになります。なおイベントハンドラの登録の説明には、前回に引き続き、TextBoxコントロールを使用します。

## 登録の手順

前回同様、「新しいプロジェクト」ダイアログで「Visual C#プロジェクト」のテンプレートから「Windowsアプリケーション」を選択し、プロジェクト名を付けて[OK]ボタンを押し、新しいプロジェクトを作成します。

Form1.csのデザイン画面が現われるので、ツールボックスから「TextBox」を選択して、フォームにドラッグします。

「textBox1」というTextBoxがForm1上に作成されます。このtextBox1を選択して、右クリックすると表示されるショートカットメニューから[プロパティ]を選択します。プロパティウィンドウが表示されるので、前回と同様、イベント一覧ボタン (☑) をクリック

図1：キーカテゴリのKeyPress イベント



し、イベントのプロパティを表示します。

アイコンをクリックした直後は「プロパティ変更」というカテゴリのTextChanged イベントにカーソルがあるので、上のほうにある「キー」というカテゴリの中から、目的のKeyPress イベントを選択します（図1）。KeyPress イベントをダブルクリックすると、「textBox1\_KeyPress」というメソッドが自動生成されます。

コードエディタで「Windows フォームデザイナーで生成されたコード」の左側にある「+」部分をクリックすると、「#region……#endregion」というコードで記述された、一時的に隠れていた部分が開きます。textBox1\_KeyPressの生成と同時に、textBox1\_KeyPressがKeyPress イベントに登録されたことが確認できます。

```
//
// textBox1
//
(略)
this.textBox1.KeyPress +=
    new System.Windows.Forms.KeyPressEventHandler(
        this.textBox1_KeyPress);
```

次にここで自動的に行なわれた、イベントの登録に関して説明してゆくことにします。

## イベントを登録？

前述の手順により、Visual Studio .NETがイベントハンドラを自動生成して、イベントに登録できることがわかりま

### 自分好みの

デフォルトの設定では#regionの部分は、閉じた状態で表示されます。もしこの隠れた部分を常に開いておきたいのであれば、[ツール] メニューから [オプション] を選択して「オプション」ダイアログを表示します。そして、ダイアログの左側のツリーを「テキストエディタ」→「C#」→「書式設定」の順でたどり、「ファイルを開いたときに#regionブロックを縮小する」というチェックを外せば、#regionブロックを開いた状態でコードを表示することができます。「オプション」ダイアログには、このほか、タブ幅の設定や、行数の表示などさまざまな設定があるので、自分好みにカスタマイズするのも面白いでしょう。

したが、ここではその自動生成されたKeyPress イベントに関して説明します。

KeyPress イベントは、C#では以下のように宣言されています。

```
public event KeyPressEventHandler KeyPress;
```

さらに、KeyPressEventHandlerの定義は、

```
public delegate void KeyPressEventHandler(
    object sender, KeyPressEventArgs e
);
```

となっています。

ここからわかるとおり、KeyPress イベントの実体は、「KeyPressEventHandler」というデリゲートになっています。デリゲートを使用するには、同じ引数のメソッドを作成します。このメソッドに、前回OnKeyPressメソッドで行っていた処理を記述します（リスト1）。

これで、前回派生クラスを作成して行っていたKeyPress イベントの処理を、デフォルトのTextBoxだけを使用して実現する方法がわかりました。

続いて、前回TextプロパティのSetアクセサをオーバーライドして行っていた「コードからの代入を阻止する部分」の実装を説明します。

まず、先ほどのKeyPress イベントと同様に、Visual Studio