

Products Review



形状を自由にデザインできる
Windows フォームコントロール集

FreeShapeButton.NET

せきぐち くみこ SEKIGUCHI, Kumiko

Software Information

OS	Windows 2000/XP
開発環境	.NET Framework 1.1 Visual Studio .NET 2003 (VB.NET/C#) Delphi8
価格	メディア販売：38,430円 ダウンロード販売：37,800円

問合先

株式会社コムラッド

FAX : 03-3290-0756

URL : <http://www.comrade.co.jp/>

MAIL : info@comrade.co.jp

はじめに

今回は、株式会社コムラッドの「FreeShapeButton.NET」を紹介します。

FreeShapeButton.NETは、.NET Framework 1.1上で動作するWindows フォームコントロール集で、形状を自由にデザインできることが特徴です(図1)。ただ単にコントロールの形状を変更できるだけでなく、コントロールの状態に応じた細やかな設定も用意されているので、ユーザーに親しみやすいインターフェイスを実現できます。

では、早速、FreeShapeButton.NETの機能について詳しく見ていきましょう。

FreeShapeButton.NETの機能

FreeShapeButton.NETは、Windows フォームの標準コントロールであるButton、CheckBox、RadioButton、Label、LinkLabel、Panel、Formを継承した7つのコントロールで構成されています。

- ① FreeShapeButton (Buttonを継承)
- ② FreeShapeCheckBox (CheckBoxを継承)
- ③ FreeShapeLabel (Labelを継承)
- ④ FreeShapeLinkLabel (LinkLabelを継承)
- ⑤ FreeShapePanel (Panelを継承)
- ⑥ FreeShapeRadioButton (RadioButtonを継承)
- ⑦ FreeShapeForm (Formを継承)

なお、FreeShapeButton.NETには、VB.NET、C#、Delphi8のサンプルプログラムが付属しており、これらの開発

Technology Tools

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

図1：FreeShapeButton.NET付属のサンプル

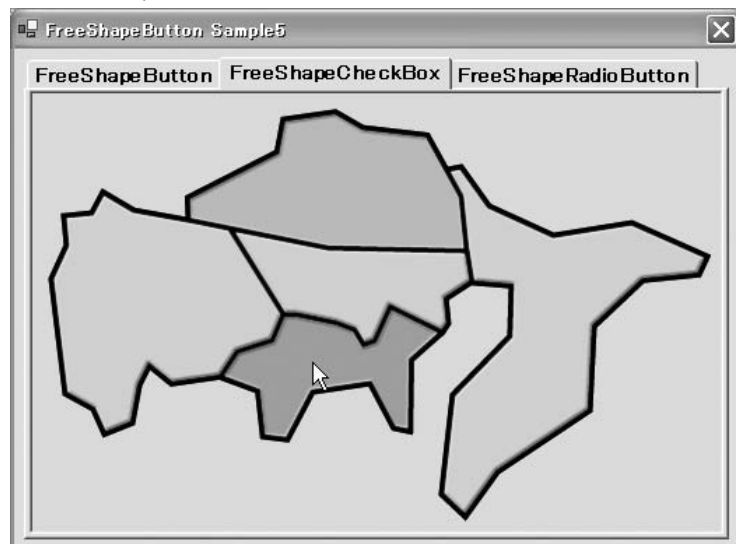


表1: FreeShapeButton.NETで指定できる状態

状態名	説明
Normal	通常時
Hover	マウス通過 (MouseHover) 時
Active	マウス左ボタン押下時 (ボタン系コントロール ^[*1] ではスペースキー押下時)
Disabled	無効時 (イメージ/フォントのみ)
Focused	フォーカス時 (ボタン系コントロール ^[*1] とFreeShapeLinkLabelのみ)
DefaultButton	デフォルトボタン時 (FreeShapeButtonのみ)


* 1) FreeShapeButton、FreeShapeCheckBox、FreeShapeRadioButton

環境で各コントロールを利用できます。

■状態対応プロパティ

コントロールの形状を変えるには、あらかじめ形状をイメージ (画像データ) として用意しておき、それをNormal Imageプロパティに指定して、コントロールの形状をイメージの有効部分 (非透過部分) に合わせるかどうかを指定するFitRegionToStateImageプロパティを「True」にするだけです。

また、FreeShapeButton.NETのコントロールは、単に形状を変えるだけでなく、コントロールの状態 (表1) に応じて形状を変えることができる「状態対応プロパティ」が用意されています。

コントロールの状態とはどのようなものかという点、Windows XPのシステムボタンを想像してもらえばよいでしょう。Windows XPスタイルの [閉じる] ボタンの背景色は、通常時は赤、マウス通過時には明るい赤、ボタン押下時には暗い赤、というようにわかりやすいインターフェイスを提供しています。

FreeShapeButton.NETのコントロールも、「状態対応プロパティ」を指定することで、Windows XPのシステムボタンと同様に、コントロールの状態に応じて表示が変化するユーザーライクなインターフェイスを実装できます。

「状態対応プロパティ」は、イメージを設定する「XxxxImage」(Xxxxは

状態名) のほか、フォントや前景色、背景色を設定するプロパティなど、コントロールの種類や状態に応じて4~6種類用意されています (表2・図2)。

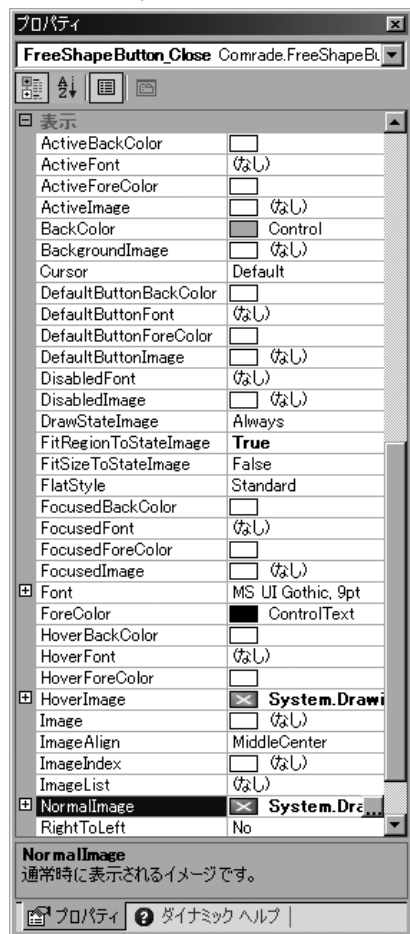
たとえば、「NormalImage」には通常時に表示する形状、「HoverFont」にはマウス通過時に表示するフォント、「FocusedForeColor」にはフォーカス時に表示する前景色、というようにコントロールの状態に応じて表示する形/フォント/色をそれぞれ指定します。

コントロールの表示を変えるという観点で言えば、通常のButtonコントロールなども、デザイン時にFlatStyleプロパティを「System」に設定しておいて、Application.EnableVisualStyles()メソッドを呼び出せばWindows XPスタイルにすることができます。FreeShapeButton.NETのコントロールも標準コントロールを継承しているのでFlatStyleプロパティが存在しますが、FreeShapeFormとFreeShapePanel以外ではFlatStyleプロパティを「System」に設定するとFreeShapeButton.NET独自の表示にはなりません。この点には注意が必要です。

■状態対応プロパティの適用順序

「状態対応プロパティ」は、ひとつのコントロールに対して最大で6種類あるので、少し多いのではないかと思います。しかし、す

図2: FreeShapeButtonの状態対応プロパティ



べての状態を設定する必要はありません。「Normal」時のイメージさえ設定しておけば、どのような状態であっても「Normal」時のイメージが適用されるからです。

「Normal」以外の「状態対応プロパティ」は、未設定 (既定値) の場合に表3の順序でイメージが適用されます。

たとえば、ActiveImageプロパティの値が未設定で、FocusedImageプロパティ、HoverImageプロパティ、NormalImageプロパティがそれぞれ設定されていた場合を考えてみましょう。コントロールの状態が「Active」のときの適用順は、

Active > Focused > DefaultButton > Hover