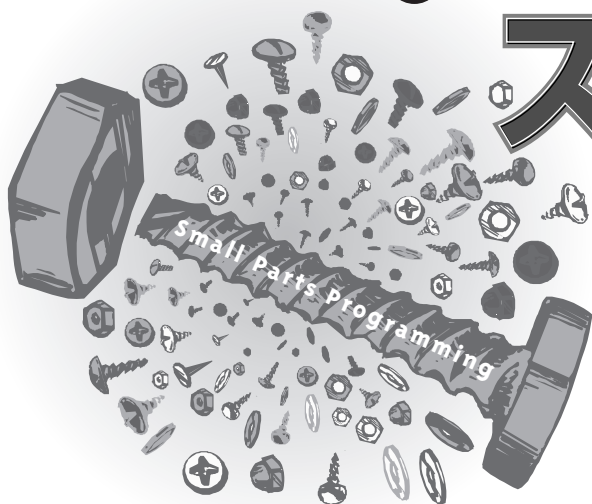


VB.NET

コンポーネント活用講座
～ 教養課程 ～



スモールパーツ プログラミング

第25回

小森 大輔 KOMORI, Daisuke

CommonDialogクラスの コントロール—Part 1—

Small Parts Programming

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¥SMALLディレクトリに収録しています。

¥COLORDIALOGSAMPLE01

カラーダイアログボックスの表示と選択された値の格納

¥COLORDIALOGSAMPLE02

ColorDialogコントロールのプロパティの使用

¥FONTDIALOGSAMPLE01

フォントダイアログボックスの表示と選択された値の格納

¥FONTDIALOGSAMPLE02

HookProcのオーバーライドによるWindowsメッセージの取得

¥HOOKPROCSAMPLE01

カラーダイアログボックスへの「適用」ボタンの追加

今回の主役はダイア ログボックス



本連載は、Visual Basic .NET (以下VB.NET) の標準コントロールにスポットを当て、とことん使い込んで、その機能や役割を調べてゆきます。

今回は、Windowsで標準的なダイアログボックスを表示し、アプリケーションに共通のUIと機能を提供する「CommonDialogクラスを継承するコントロール」を主役に据えてお話ししようと思います。

CommonDialog クラスを継承する コントロール



CommonDialogクラスはMustInherit宣言された基本クラスとなっており、このクラス自体をインスタンス化することはできません。CommonDialogクラスには、図1のような、このクラスから派生するさまざま

なクラスがあります。CommonDialogクラスを使用する場合、実際にはこれらの派生クラスを使用することになります。

これらのコントロールをひとつずつ見てゆくことにしましょう。

ColorDialog コントロール



ColorDialogは、色の選択を行なうことができるコントロールです。ユーザーが色を選択する必要があるアプリケーションすべてで、共通のUIから色の選択を行なうことができるというメリットがあります。また、色を選択するだけでなく、表示されたダイアログボックス内で独自の色を作成することも可能です。

カラーダイアログを 表示するには

ColorDialogコントロールの使い方はいたって簡単です。このコント

図1：CommonDialogクラス階層図

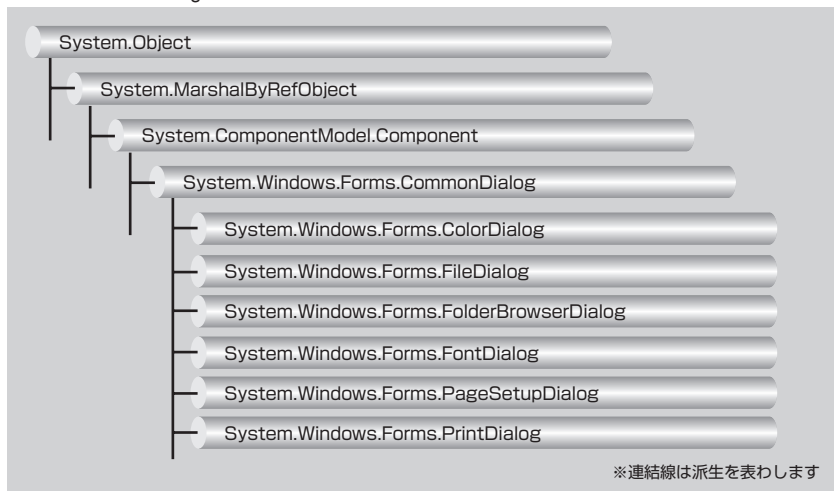
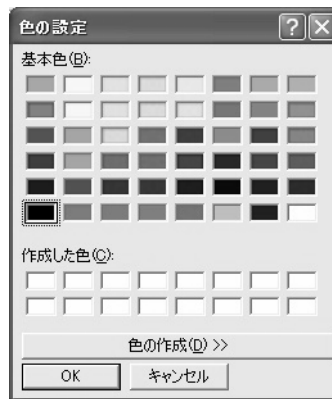


図2：カラーダイアログボックス



ロールのShowDialogメソッドをコールし、戻り値であるDialogResult列挙体の値を調べ、“DialogResult.OK”である場合にColorプロパティに格納された「ユーザーが選択した色」を取得するだけです。DialogResult列挙体の値は、日本語環境では表1のようにになっています。

リスト1は、フォームのクリックイベントでカラーダイアログボックスを表示し、フォームの背景色をユーザーが選択した色に設定する例です。図2は、このプログラムによって表示されたダイアログボックスです。

図2を見てわかるように、ColorDialogコントロールでダイアログボックスを表示した際のColorプロパティの初期値は、“Color.Black”になっています。これを変更するには、ShowDialogメソッドをコールする前に、

```
ColorDialog1.Color = Color.Red
```

表1：DialogResult列挙体

値	意味
DialogResult.About	[中止] ボタンが押された
DialogResult.Cancel	[キャンセル] ボタンが押された
DialogResult.Ignore	[無視] ボタンが押された
DialogResult.No	[いいえ] ボタンが押された
DialogResult.None	ダイアログボックスからNothingが返された (ダイアログボックスは実行中)
DialogResult.OK	[OK] ボタンが押された
DialogResult.Retry	[再試行] ボタンが押された
DialogResult.Yes	[はい] ボタンが押された

のように、Colorプロパティにあらかじめ色を設定しておきます。ですからリスト1の場合も、

```
ColorDialog1.Color = BackColor
```

というように、事前にColorプロパティに値をセットしておくほうが親切なUIかもしれません。ただしこの場合は、フォームの背景色、すなわち「Color.FromName ("Control")」で表わされるシステム定義色が純色ではないため、Colorプロパティの値としては有効ですが、ダイアログボックスには初期値として表示されません。

リスト1：カラーダイアログボックスの表示

```
Private Sub Form1_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles MyBase.Click
    With ColorDialog1

        ' 戻り値であるDialogResult型の変数を宣言し
        ' ダイアログボックスの戻り値を格納
        Dim dlgRes As DialogResult = ColorDialog1.ShowDialog

        ' [OK] ボタンが押された場合
        If dlgRes = DialogResult.OK Then
            ' フォームの背景色を変更
            BackColor = Color
        Else ' 他の戻り値の場合
            ' メッセージボックスを表示
            MessageBox.Show(dlgRes.ToString)
        End If

    End With
End Sub
```

