

# 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24回

小森 大輔 KOMORI, Daisuke

### LinkLabelコントロール —Part2—

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

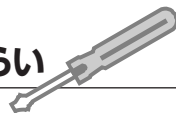
・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥SMALLディレクトリに収録しています。

¥ENVIRONMENTSAMPLE  
Environmentクラスを使用するサンプル

¥LINKLABEL2A  
アイコンの表示

¥LINKLABEL2B  
関連付けアイコンの表示

#### 前回のおさらい



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、その機能や役割を探ってゆきます。

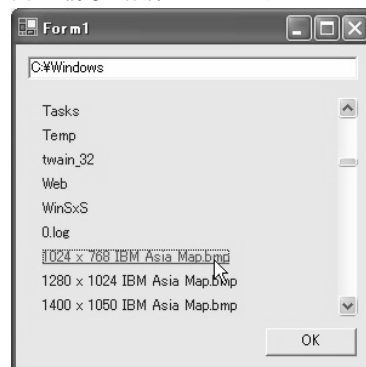
前号ではLinkLabelコントロールを使用して、以下のような処理を行ないました。

- ① LinkLabelコントロールを使ったリンク先の表示
  - ・コントロールの初期値の設定
  - ・リンクの表示
  - ・リンク先の表示 (System.Diagnostics.ProcessクラスのStartメソッドを使用したプロセスの起動)
  - ・設定した文字列内のリンクエリアの設定
  - ・単一のLinkLabelコントロールへの複数のリンクの表示

- ② LinkLabelコントロールを使用したランチャーの作成
  - ・ System.IOネームスペース内のクラスを使用したファイルやフォルダの列挙
  - ・ LinkLabelクラスの動的なロード
  - ・ Panelコントロールをコンテナにする方法
  - ・ AddHandlerステートメントを使用したイベントハンドラへの関連付け

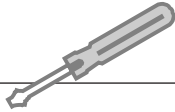
作成したのは図1のようなUIのランチャーです。ですがこのランチャー、サンプルで作成したものにしろ

図1：前号で作成したランチャー



何かすごく物足りない気がしたので、今回はもう少し手を加えてみたいと思います。

## ディレクトリの指定を ComboBoxで!



今の状態では、テキストボックスへディレクトリを手入力して指定しなければなりません。これをComboBoxにして、クリックだけでディレクトリを指定できるようにすると便利です。まずはこれを実装してみましょう。

### 仕様を決めてみる

「ComboBoxでの指定」という小さな処理ですが、このような処理を行なう際にも仕様はしっかりと決めておいたほうが後々楽です（はじめにしっかり決めてないから、今ごろテキストボックスからComboBoxへ変更したりしています……）。

ここでの仕様は、以下のようなものにしましょう。

仕様1▶ ComboBoxの初期値はルートのドライブ表示

仕様2▶ ComboBoxが開かれたときにすべてのアイテムを削除し、ComboBoxの現在の値をルートにしてディレクトリを列挙し、アイテムを再構築

仕様3▶ Buttonのイベントではなく、ComboBoxの値が変更された際にファイルやフォルダを列挙し、LinkLabelで表示

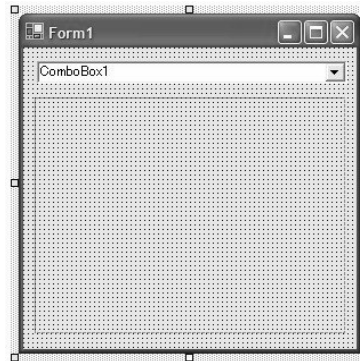
### UIの変更

仕様がめでたく決まったところで、

- ・フォームのテキストボックスを削除する
- ・新たにComboBoxコントロールを追加する
- ・Buttonコントロールを削除する

といったUIの変更を行ないます（図2）。ComboBoxの初期値を“ルートのドライブ表示”とするのは、コントロールの初期値を設定するSetInitialValueプロシージャで、テキストボックスへ値を代入していた部分をComboBox

図2：新しいUI



リスト1：コントロールの初期値の設定

```
Protected Sub SetInitialValue()  
    ' Panelのスクロールバーを表示  
    Panel1.AutoScroll = True  
  
    ' ComboBoxのデフォルト値を  
    ' システムドライブのルートパスにする  
    ComboBox1.Text = _  
        Path.GetPathRoot(Environment.SystemDirectory.ToString)  
End Sub
```

に変更するだけです（リスト1）。

ルートのドライブを取得するには、EnvironmentクラスのSystemDirectoryプロパティで“システムドライブのパス”を取得し、その値をPathクラスのGetPathRootメソッドの引数にすればよいのでしたね（Path.GetPathRootメソッドは、指定したパスのルートディレクトリ情報を取得できます）。

### ComboBoxへアイテムを設定する

続いて、ComboBoxコントロールのテキストに設定されたパス文字列をルートにし、ディレクトリを列挙してアイテムを再構築する処理を作成します。この処理は、ComboBoxのDropDownイベントプロシージャで行なうことにします（リスト2）。

ここではまず、ComboBoxコントロールの現在のアイテムをClearメソッドですべて削除しています。

次に、DirectoryInfoクラスのコンストラクタへComboBoxコントロールのTextプロパティの値を渡して初期化し、インスタンスを作成します。DirectoryInfoクラスのコンストラクタへ渡したパスが不正である場合には、こ