

Visual Basic NET

ツボ

データビューによるデータ表示—その2—

西田 雅昭

NISHIDA, Masaaki

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¥TUBOディレクトリに収録しています。

¥VIEWTEST2

今回作成したデータビュー実験プログラム

¥DB

サンプルデータベース (Human1.mdb)

¥OLD

前回作成したプログラム

*) 今回のサンプルプログラムは、データファイルを「E:\¥dotNETMagazine¥VB31¥」ディレクトリに配置しているという前提で作成しています。サンプルを実行するには、上記のディレクトリにデータファイルを配置するか、「OleDb Connection」コントロール (ocnHuman) の「ConnectionString」プロパティの値を自分の環境に合わせて修正する必要があります。

前回は、「DataView」クラスを利用して、処理中のデータベースの表示にフィルタをかけたり、並べ替えを行なう方法を紹介しました。

今回は、もう少し「DataView」クラスの凝った使い方を紹介しましょう。



検索機能をつける

前回作成したフォーム (図1) では、フィルタと並べ替えの機能をつけました。続いて、つけたい機能といえば「検索」ですね。そのために、まずはフォームにコントロールを追加しましょう。今回から読み始めた方は、前回のプログラムを付録CD-ROMに収録していますので (OLDフォルダ)、ご利用ください。

ラベルを2つ、コンボボックスを2つ、テキストボックスを2つ、ボタンをひとつ追加します (図2)。

「検索条件」ラベルと「検索結果」ラベルは「AutoSize」プロパティと「Font」プロパティの「Bold」を“True”に設定しています。

また、データビューにテーブルを設

定する ([DataView] ボタンを選択する) 前に、検索を行なうことができないように、追加したコントロールすべての「Visible」プロパティを“False”にしておきます。

●列の選択

検索は、特定の列に対して行なうこととなりますが、検索する際には列名を忘れていた場合もありますから、列名の一覧をコンボボックスに表示して、選択することにしました。

まず、列名一覧を持つコンボボックスを作成します (リスト1)。「mdtvHuman.Table」の“Table”は、データビューのソースとなっている「DataTable」オブジェクトを取得するプロパティです。「mdtvHuman.Table.Columns」で、このデータビューの持つすべての列のコレクションを取得できます。

図1：前回作成したフォーム

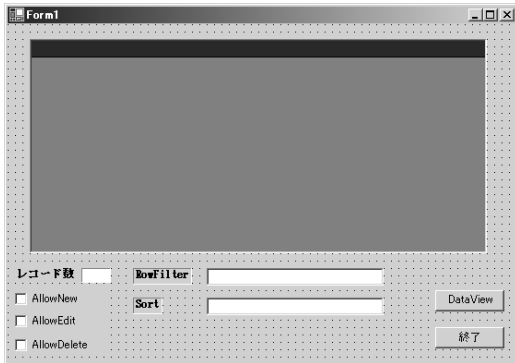
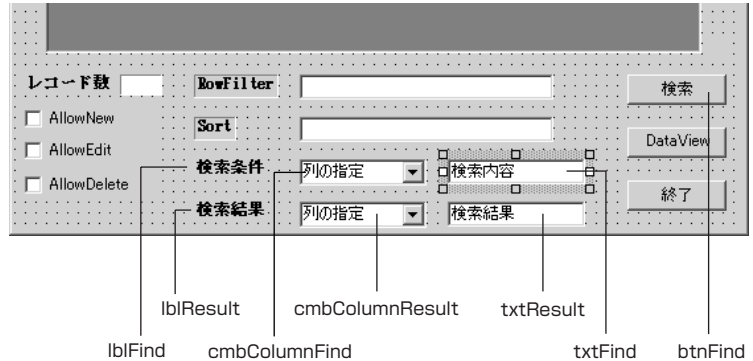


図2：検索用のコントロールを追加



For Each…Next構文は、もうわかりですね。すべての列の名前をコンボボックスに設定しているのです。

● 検索の手順

続いて、「btnDataView_Click」イベントハンドラの最後に、次のコードを追加します。

```
mprcComboMake()

lblFind.Visible = True
cmbColumnFind.Visible = True
```

コンボボックスに列の一覧を書き込み、検索の最初の手順に関するコントロールを見えるようにしています。

次は、検索を行なう列の指定です(リスト2-①)。「txtFind」テキストボックスを見えるようにし、フォーカスを移します。これによって、テキストボックスの内容が反転し、この内容を消去

せずに新しい内容を書き込むことができます。

参考

VB.NETでは、「ComboBox_Changed」イベントではなく、「ComboBox_SelectedIndexChanged」イベントを使うことになりました。

テキストボックスに検索する内容を書き込んだら、次は検索結果として、どの列の内容を表示するかの指定です(リスト2-②)。データが入力されていたら、検索結果としてどの列の内容が必要かを指定するためのコントロールを見えるようにします。

● 奇妙なValidate

次に、ユーザーが操作の途中で「終了」ボタンを選択することも考慮して、「btnQuit」ボタンの「CausesValidation」プロパティを“False”に設定しておき

ます。

実は、ここで奇妙な現象に遭遇しました。「txtFind」テキストボックスに入力して「Tab」キーを押すと、フォーカスが「終了」ボタンに移り、もう1度「Tab」キーを押して初めて「txtFind_Validating」イベントが発生します。もちろん「TabIndex」は、

txtFind (4) → cmbColumnResult (5) → btnFind (6)

の順になっており、「終了」ボタン (btnQuit) の「TabIndex」の値ははるか後ろです(図3)。

「TabIndex」の順序を見ると、「txtFind」から「btnQuit」までの間に、「Text」プロパティが“空”に設定されている入力を許すコントロール(ここではテキストボックス)がないためだと思われます。

リスト1：列名一覧を表示するコンボボックスの作成

```
Private mdtvHuman As New DataView()

Private Sub mprcComboMake()
    Dim strColumnName As String
    Dim clmNow As DataColumn
    Dim colColumn As DataColumnCollection

    colColumn = mdtvHuman.Table.Columns

    For Each clmNow In colColumn
        cmbColumnFind.Items.Add(clmNow.ToString)
        cmbColumnResult.Items.Add(clmNow.ToString)
    Next
End Sub
```