

# Visual Basic NET



## 第8回 データベースフォームの作成と 親切なユーザーインターフェイス

西田 雅昭  
NISHIDA, Masaaki

前回は、ログインのためのフォームを作成しました。今回はまず、この「ログイン」フォームを「メニュー」フォームから開くための処理から始めましょう。

また、今回はデータベース処理のフォームのやさしい作り方も紹介します。



### はじめに

新しい内容に入る前に、お詫びして訂正しなければならないことがあります。

前回作成した「ログイン」フォームで、パスワードを入力した後の操作がわかりにくい、とのご意見をいただきました。パスワードを入力し終わった状態でのコメントは「パスワードを入力してください」のままだし、[OK] ボタンが見あたらないので何をすればよいかわからない (図1)、というご意見です。

そこで、フォーム「frmLogin」の内容を少し変更することにしました。まず、このフォームにメッセージを追加することから始めたいと思います。

フォーム「frmLogin」に「mcstr

MSGTab」という定数を追加し、さらに、定数「mcstrMSGOK」の宣言を書き換えます (リスト1)。

続いて、フォーム「frmLogin」のテキストボックス「txtPassword」のTextChanged イベントハンドラをリスト2のように作成してください。

これでプログラムを実行してみます。今度は、パスワードの入力を始めるとメッセージが、

「これでよければ、「Tab」キーを打ってください」

に変わるようになります (図2)。これで操作がわかりやすくなりました。正しいパスワード「dotnet」を入力した状態で [Tab] キーを押すと、[OK] ボ

#### 本稿で前提となるもの

OS | Windows XP

開発環境 | Visual Basic.NET

初級

中級

上級

本記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥TUBOフォルダ以下に収録しています

¥0211：前回までに作成したプログラム

¥0212：今回作成したプログラムとmdbファイル

図1：ユーザーが何をすればよいかわからない

リスト1：表示されるメッセージの追加と変更

```
' 確認メッセージの追加
Const mcstrMSGTab As String = "これでよければ、「Tab」キーを打ってください"

' パスワード入力後のメッセージの変更
Const mcstrMSGOK As String = "「OK」ボタンを選択してください"
```

リスト2：パスワード入力後にメッセージを表示

```
Private Sub txtPassword_TextChanged(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles txtPassword.TextChanged

    lblMSG.Text = mcstrMSGTab

End Sub
```

タンが現われてフォーカスもボタンに移ります<sup>[註1]</sup>。

ここで気が付いたのですが、名前を入力した後も [Tab] キーを押さない人がいるかもしれません。同じようにテキストボックス [txtUser] の Text Changed イベントハンドラも作成しておきます。



## ログイン時のユーザーインターフェイス

前回は、何も考えずに「ログイン」フォームを作成してしまいました。実際の操作を考えた場合、ログインしな

注1) 先月号の付録CD-ROMでは、txtPassword\_Validating イベントハンドラの [btnOK.Visible = False] がコメント化されているため、誤ったパスワードを入力した場合でも [OK] ボタンが現われます。

図2：的確な指示をユーザーに示す

いと何も操作できないようにしなければなりません。そのためには、[ログイン] ボタン以外のすべてのボタンとメニューを使えないようにする必要があります。そのためのプロシージャを作成することにしましょう。

### ●ログインする前は

フォーム [frmMenu] のモジュールレベルのプロシージャですから、最初に記述することにしましょう。

[Windows フォームデザイナーで生成されたコード] という行のすぐ下に、[Private Sub mprcInit] と入力して [Enter] キーを押します。すると、エディタが1行空けて [End Sub] を作ってくれます。その上、ちゃんとインデントを付けた位置にカーソルがあります。Visual Basic.NET (以下VB.NET) は、本当に親切ですね。

リスト3：frmMenuのLoadイベントハンドラ

```
Private Sub frmMenu_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load

    mprcInit()

End Sub
```

さて、そのmprcInitプロシージャの内容は、

```
btnAddNew.Enabled = False
btnEdit.Enabled = False
:
```

のように、片端からコントロールの [Enabled] プロパティを [False] にすればよいのですが、コントロールがたくさんあるので面倒です。

そこで、“コレクション”を使った面白いテクニックを紹介しましょう。

```
Dim ctrNow As Control

For Each ctrNow In Controls
    ctrNow.Enabled = False
Next ctrNow
```

これで、すべてのコントロールの [Enabled] プロパティを [False] にすることができます。しかし、このままでは何もできなくなってしまうので、[ログイン] ボタンと [終了] ボタンだけは使えるようにしましょう。このプロシージャの最後に、

```
btnLogIn.Enabled = True
btnQuit.Enabled = True
```

を付け加えます。あとはこのプロシージャを、フォーム [frmMenu] のロード時に実行するように指定すればよいのです (リスト3)。