

# Visual Basic NET

# のツボ

## 第27回 ADO.NETを利用した データベース処理 —その7—

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。

#### ¥OLD

前回作成したサンプルアプリケーション

#### ¥ADOCODE3

今回作成したサンプルアプリケーション

#### ¥DATA

サンプルデータベース (Shimei.mdb)

\*) サンプルプログラムは、データファイルを「E:\dotNETMagazine¥VB21¥Data¥」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。サンプルの実行時には、上記のディレクトリにデータファイルを配置するか、以下の2つを自分の環境に合わせて修正する必要があります。

- ・ [btnConnect\_Click] プロシージャの接続文字列
- ・ [OLEDBConnection] コントロール (OLEDBConnection1) の [ConnectionString] プロパティ

本連載では、ADO.NETを使ったデータ処理プログラムを作成しています。今回は、「データの更新」について考えてみることにします。



### データを表示する

まずは、前回までに作成した「ADO Test3」プロジェクトを起動して、実行してください。前回のプログラムは、付録CD-ROMに収録していますので(OLDフォルダ)、今回から読み始めた方はご利用ください。

[Fill1] ボタンをクリックすると、「データソースとの接続」を開いたことを表わすメッセージボックスが現われますから、[OK] ボタンをクリックしてください。すると、データを表示し、

図1：グリッド上でデータを編集



今回実験に使用する  
[Fill1] ボタンとナビゲータボタン

またメッセージボックスが現われます。再度 [OK] ボタンをクリックすると、データを見ることができます。



### データを変更する

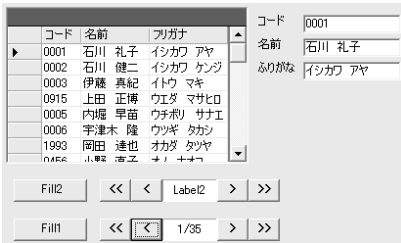
データを表示したら、「ふりがな」テキストボックスで「イシカワ アヤコ」の「コ」を消去してみましょう(図1)。不思議なことに、データグリッドの中のデータは変わっていません。

[>] ボタンでひとつ移動してから [<] ボタンで元に戻ると、確かにデータを変更したことがわかります(図2)。

フォーム右上の「終了」ボタン ([X]) で終了して、もう一度実行します。なんと、データは元のままの「イシカワ アヤコ」と変わっていません。これは、どうしたことでしょう。

以前に、データセットは、「データソースを元に作成した、メモリ上に常駐

図2：データを変更できた



するデータの表現である」ということをお話しました。データソースとの接続をクローズしてしまっても、データベース内の表示するレコードの位置を変更することができましたね。

テキストボックスなどのコントロールは、このメモリ上のデータセットのテーブルや列に連結されているのです。したがって、データセットの内容を変更しても、データソースの内容は変わりません。データアダプタの力を借りてデータソースの内容を変更しなければならぬのです。

データソースの内容をデータセットに取り込むのに「Fill」メソッドを使用したように、データセットの内容にあわせてデータソースの内容を変更するには「Update」メソッドを使います。



## 「Update」メソッドを使う

まず、フォームに「Save1」というボタン（名前は「btnSave1」）をひとつ追加しましょう。

リスト1：「Save1」ボタンをクリックしたときの処理

```
Private Sub btnSave1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnSave1.Click
    OleDbDataAdapter1.Update(DsShimei.tbShimei)
    mprcDisplayPosition1()
End Sub
```

表1：DataAdapter.Updateメソッドの書式

Overloads Public Function Update(ByVal dataRows() As DataRow) As Integer		
パラメータ	dataRows	データソースの更新に使用するDataRowオブジェクトの配列
戻り値	DataSet内で正常に更新された行数	
Overrides Overloads Public Function Update(_ ByVal dataSet As DataSet) As Integer Implements IDataAdapter.Update		
パラメータ	dataSet	データソースの更新に使用するDataSet
戻り値	DataSet内で正常に更新された行数	
Overloads Public Function Update(ByVal dataTable As DataTable) As Integer		
パラメータ	dataTable	データソースの更新に使用するDataTable
戻り値	DataSet内で正常に更新された行数	
Overridable Overloads Protected Function Update(ByVal dataRows() As DataRow, _ ByVal tableMapping As DataTableMapping) As Integer		
パラメータ	dataRows	データソースの更新に使用するDataRowオブジェクトの配列
	tableMapping	使用するTableMappingsコレクション
戻り値	DataSet内で正常に更新された行数	
Overloads Public Function Update(ByVal dataSet As DataSet, _ ByVal srcTable As String) As Integer		
パラメータ	dataSet	データソースの更新に使用するDataSet
	srcTable	テーブルマップに使用するソーステーブルの名前
戻り値	DataSet内で正常に更新された行数	

続いて、「Save1」ボタンのイベントハンドラを作成します。

「Update」メソッドは「Fill」メソッドと同じように、多くのオーバーロードがあります（表1）。データセット、テーブルなどいろいろなパラメータを使い分けることができることを覚えておいてください。

ここでは、「Fill1」ボタンの場合と同じように、表1の一番下のオーバーロードを使います（リスト1）。

さっそく実行してみましょう。先ほどと同じように、「ふりがな」テキスト

ボックスで「イシカワ アヤコ」の“コ”を消去してください。[>] ボタンでひとつ移動してから [<] ボタンで元に戻ると、前掲の図2と同じく、確かにデータを変更したことがわかります。

ここで「Save1」ボタンをクリックします。すると、「Fill1」ボタンをクリックしたときと同じように、「データソースとの接続」が開いたことを表わすメッセージボックスが出ます（図3）。[OK] ボタンをクリックすると、今度は接続を閉じたことを表わすメッセージ

図3：「Save1」ボタンをクリック後にメッセージが現われた

