

# 例題でわかる データ処理

Webアプリケーションに欠かせない処理法を徹底伝授

第4回

## データの追加と検証

PROJECT KySS ● <http://www.PROJECTKySS.NET/>

### Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
  - XML
  - Access 2003

### 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¥WEB¥dotNetMagazineディレクトリに収録しています。

#### ¥DATA

サンプルで利用しているデータ

#### ¥SQLADDDATA

SQL Server/XMLデータを追加するサンプル

### ユーザーを意識した処理

これまでの3回で、SQL ServerデータとXMLデータの互換、データの更新/削除について見てきました。この辺で、Webアプリケーション開発には不可欠の、ユーザーを意識した処理を練習していきましょう。今回は、ユーザーが入力したデータをもとに、SQL ServerデータおよびXMLデータを追加してみます。

例題  
その  
1

### 入力されたデータをSQL Serverに追加する

今回はデータを追加していくため、フォーム上には、表示用のコントロールだけでなく、入力ボックスを設ける必要があります。

今回も、商品データを登録したデータベース (図1) を使います [註1]。

ただし、今回初めて本稿を読まれる方は、Access 2003で「Microsoft SQL Serverデータベースウィザード」を使い、付録CD-ROMに収録しているMDBデータ (productData.mdb) からproductData.adpを作成して、「productDataSQL」というデータベース名で登録しておいてください [註2]。

今回のサンプルでは2つの処理を行いません。ひとつは入力されたデータをSQL Serverデータとして登録し、DataGridコントロールの最後尾にバインドしていく処理です。もうひとつは、追加されたDataGridコントロール内のデータを、XML

注1) データの追加を練習するため、既存のデータの件数は少なくしています。

注2) Access 2003のMDBデータをSQL Serverにストアする手順については、第1回 (2004年12月号) を参照してください。付録CD-ROMには、MDBファイル (productData.mdb) を再収録しています。今回初めて本連載を読まれる方は、このファイルをもとに、「productDataSQL」というデータベース名のSQL Serverデータベースを作成してください。

図1：今回使用するSQL Serverデータベース

品番	品名	単価	仕入先
DPC-MMT11	ポータブルCDプレイヤー	9980	A社
SAD-2222	ステレオCDシステム	8800	B社
NV-3333	ハイファイビデオ	9800	C社
EC4444	ハイパワークリー	15800	D社
BW-5555	食器洗い乾燥機	39800	E社

に変換してTextBoxコントロールに表示する処理です。

## ●画面のレイアウト

まずはSQL Serverデータを追加する処理から見ていきましょう。

Visual Studio .NET 2003 (以下VS.NET) を起動し、ASP.NET Webアプリケーションを作成してください (この例ではプロジェクト名はsqlAddData、プログラムはWebForm1.aspx.vb)。そして、データ入力用のボックスと、データを表示させるコントロールを、レイアウトしていきます (図2)。

入力ボックスは、HTMLコントロールのTableコントロールを使って作ります。右クリックで表示されるショートカットメニューで列や行を増減し、1列目には項目名を直接入力し、2列目にはTextBoxコントロールをドラッグ&ドロップします。

今回は、入力されたデータをチェックするため、RequiredFieldValidatorコントロールも使ってみましょう。

詳細は後述しますが、2列目にRequiredFieldValidatorコントロールをドラッグ&ドロップし、ErrorMessageプロパティに“※”を入力しておいてください。

また、XMLデータを表示させるため、ButtonとTextBoxコントロールもレイアウトします。TextBoxコントロールのTextModeプロパティには、複数行の入が可能ないように“MultiLine”を指定しておきます。

DataGridコントロールもレイアウトし、AutoGenerateColumnsプロパティの値に“False”を指定してから、プロパティビルダを使って、各列の定義を行なってください<sup>[注3]</sup>。[OK] ボタンをクリックすると、図3のように「DataGrid1プロパティ」で設定した内容が、DataGridコントロールに反映されます。「DataGrid1プロパティ」の「データフィールド」に指定した項目名は、プログラムの中で、DataColumnオブジェクトを生成する際に、同じ項目名を指定する必要があります。

ここまでの作業ができれば、プログラミングに移る前に、SQL Serverデータベースに接続しましょう。データベース名は「productDataSQL」として登録してください<sup>[注4]</sup>。

注3) プロパティビルダの設定方法については、これまで何度も詳説しましたので、前回までの記事を参照してください。

注4) 各々の環境に合わせて設定を行ってから、プログラミングへと進みましょう。手順については、前回、前々回に詳説しましたので、参照してください。

図2：入力ボックスと、データを表示させるDataGridを使用

