

ADO.NETの初歩からXMLデータベースまで一刀両断
AccessやExcelのデータも自由自在

ASP.NETで使う データベース バインディング入門

わかりにくいデータベースバインディングをじっくり解説

大澤 文孝

OSAWA, Fumitaka

Technology Tools

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

Level



Samples

はじめに

ASP.NETでは、各種コントロールとデータバインディングすることで、データベース操作するWebアプリケーションを容易に構築できます。

ASP.NETに用意されているコントロールのなかでも、DataGridコントロールは、表形式でユーザーにレコードを表示するだけでなく、編集させることも可能な、高機能なコントロールです。

DataGridコントロールは、多機能なので、使い方が難しい面もあります。

そこで本稿では、DataGridコントロールを使ったレコードの表示方法から、編集方法までを説明します。

DataGridコントロールを使ったデータバインディングの基本

DataGridコントロールには、配列(Collection)、ArrayList、HashTable、DataSet、DataTable、DataView、DataReader (SqlDataReader) などの各種オブジェクトをデータバインディング

できます。

Visual Studio .NETでデータベースアプリケーションをビジュアルに開発する場合には、非接続型データベースアクセスを使うのが主となるでしょう。

その場合、DataSet、DataTable、DataViewのいずれかのオブジェクトをデータバインディングすることになります。

どのオブジェクトを使ってもかまいませんが、別稿で説明したように、DataViewオブジェクトを使うと、条件によるフィルタリング操作ができて便利です。

そこで多くの場合は、DataViewオブジェクトをデータバインディングします。

DataViewオブジェクトをDataGridコントロールにデータバインディングするには、図1のように、DataGridコントロールを配置し、DataSourceプロパティにDataViewオブジェクトを設定します。

データバインディングのためのコード

図1のようにDataSourceプロパティにデータバインディングとなるオブジェクトを設定しただけでは、データを取

図1：DataGridコントロールの配置



得て表示するという処理は行なわれません。データバインディングは、DataBindメソッドを呼び出したときに実行されます。

もちろんDataBindメソッドを呼び出す前には、データバインディングの対象となるオブジェクト——DataSourceプロパティに設定したオブジェクト——に表示したいデータが格納されている必要があります。

そこで、

①SqlDataAdapterオブジェクトのFill

リスト1：データバインディングを実行する

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

    ' ページを初期化するユーザーコードをここに挿入します。
    SqlDataAdapter1.Fill(DataSet1)
    If Not Me.IsPostBack Then
        DataGrid1.DataBind()
    End If
End Sub
```

メソッドを呼び出してデータベースからレコードを取得する
②DataBindメソッドを呼び出してデータバインディングする

の2つの処理が必要です。

これらは、WebフォームのLoadイベントに記述すればよく、リスト1のようになります^[注1]。

実際、リスト1の処理を実装すると、図2のように、データバインディングした結果が表示されます。

DataGridコントロールのプロパティを設定する

DataGridコントロールの動作や外観は、プロパティ画面から設定できます。

列を定義する

DataGridコントロールのプロパティの[列] ページでは、表示する列の順序や書式、ヘッダーの文字を変更できます。

デフォルトでは、[実行時に自動的に列を作成する] にチェックが付けられています。この場合、データバインディングされた列が自動的に作成され、先の図2に示した結果となります。

しかし [実行時に自動的に列を作成する] のチェックをはずし、あらかじめ列を作成しておく、列のヘッダー名を変更したり、一部の列だけを見せるようにしたり、リンクとして構成したり、ボタンとして構成したりできるようになります。

[列] ページにおける列の定義例は、図3のようになります。いくつか列をカスタマイズすると、表示は、図4のようになります。

今回は説明しませんが、DataGridコントロールの [書式] ページで、いくつかの設定をすれば、さらに、各列の背景色やフォント、左寄せ、中央寄せ、

注1) リスト1では、IsPostBackプロパティを調べて、非ポストバック状態のときだけデータバインディングしてします。これは、本稿の最後で説明する編集処理において、ユーザーがテキストボックスに入力した値をデータバインディングによって上書きしてしまわないために必要です。なぜならLoadイベントとのちに説明するUpdateCommandイベントでは、Loadイベントのほうが先に発生するためです。