

.NET Framework

なにを使うか、どう使えるのか アイデアノート

第 9 回

秋月巖ソリューション事務所
秋月 巖 AKIZUKI, Iwao
http://www.akizuki.co.jp

ADO.NET 自体を データベースエンジンとして使用する — その3 —

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¥NOTEディレクトリに収録しています。

¥帳票
今回解説した帳票作成プログラム

帳票作成プログラムの構成

外部データベースエンジンを使わずにADO.NETで開発した帳票作成プログラム (図1) を、先月号では簡単に紹介した。このプログラムは、表1のようなプロシージャによって構成されている。

以降、主要なプロシージャについて解説する。

フォームのLoadイベントプロシージャ [Form1_Load]

このプロシージャ (リスト1) では、まず以下の部分でデータベースファイルを読み込んでいる。

```
DataSet1.ReadXml("northwind1.xml", _  
    XmlReadMode.ReadSchema)  
DataSet2.ReadXml("northwind2.xml", _  
    XmlReadMode.ReadSchema)
```

northwind1.xmlには更新系のテーブル、northwind2.xmlにはマ

スター系のテーブルが格納されている。それぞれ別のDataSetオブジェクトのDataTableオブジェクトに読み込まれる。これらのDataTableオブジェクトは、以降インメモリのテーブルとして扱われる。

以下のコードは、帳票のマスター部のデータとしてテーブル連結クエリーを実行している。

```
Dim FieldListMaster() As String = {  
    {"受注", "受注コード"}, _  
    {"受注", "受注日"}, _  
    {"得意先", "得意先名"}, _  
    {"得意先", "得意先コード"}, _  
    {"得意先", "都道府県"}, _  
    {"得意先", "住所1"}}  
DataTableMaster = _  
    JoinTable(FieldListMaster, _  
        DataSet1.Tables("受注"), _  
        "INNER", _  
        DataSet2.Tables("得意先"), _  
        "得意先コード", "", _  
        "", _  
        "受注コード", _  
        "", _  
        "", _  
        DataSetJoinTable, _  
        "DataTableMaster")
```

このフォームのロード時に、すべ

表1：帳票作成プログラムのプロシージャ

| | | |
|---|---|---|
| Form1_Load | フォームのLoadイベントプロシージャ | <ul style="list-style-type: none"> データベースファイルの読み込み 帳票マスター部の連結テーブル作成 画面表示コントロールへのデータ連結 |
| btnNext_Click | [次へ] ボタンのClickイベントプロシージャ | <ul style="list-style-type: none"> 次の帳票を表示 |
| btnPrev_Click | [前へ] ボタンのClickイベントプロシージャ | <ul style="list-style-type: none"> 前の帳票を表示 |
| SetRow | サブプロシージャ。このプロシージャは新しい帳票が表示されるたびに呼び出される。つまり、上記の3つのプロシージャから呼び出される | <ul style="list-style-type: none"> 帳票詳細部の連結テーブルの作成 DataGridコントロールへの連結 合計の計算と表示 削除ログテーブルのクリア |
| btnUpdate_Click | [この帳票の変更をデータセットに反映] ボタンのClickイベントプロシージャ。「JoinTableWrite」汎用プロシージャを呼び出す。ファイルへの出力は行なわれない | <ul style="list-style-type: none"> 変更した内容をDataSetオブジェクトに反映 削除ログテーブルを参照して、元テーブルから削除する |
| btnWriteFile_Click DataTableDetail_ RowDeleting | [ファイルに保存] ボタンのClickイベントプロシージャ「DataTableDetail」DataSetオブジェクトのRowDeletingイベントプロシージャ。「JoinTableRowDeleting」汎用プロシージャを呼び出す | <ul style="list-style-type: none"> DataSetオブジェクトの内容をファイルに出力 帳票詳細部（連結テーブル）の削除された行に該当するオリジナルテーブルの行を、削除ログテーブルに移動 |
| Grid_CurrentCell Changed | DataGridコントロールのCurrentCellChangedイベントプロシージャ | <ul style="list-style-type: none"> 商品番号から商品名を検索 行を移動した場合は合計値の再計算 |

図1：帳票作成プログラム



ての注文データのマスター部を取得している。つまり [次へ] [前へ] ボタンをクリックしたときに、このクエリは再実行されない。アプリケーションの実行中、1度だけ実行されるクエリーである。異なるDataSetオブジェクトに属するDataTableオブジェクトを使用して、データ連結を行なっている。結果である連結テーブルは、以降「DataTableMaster」DataTableオブジェクトとして操作することができる。

以下は連結したテーブルを画面の表示コントロールに連結している。

```
lblCode.DataBindings.Add(
    New Binding("Text", DataTableMaster, "受注コード"))
lblDate.DataBindings.Add(
    New Binding("Text", DataTableMaster, "受注日"))
    ...
```

以下の2行では、詳細部のデータを生成して表示するための「SetRow」サブプロシージャを呼び出した後、DataGridコントロールのCurrentCellChangedイベントとイベントプロシージャを関連付けている。

```
SetRow()
AddHandler DataTableDetail.CurrentRowDeleting,
    AddressOf Grid_CurrentCellChanged
```

DataGridコントロール内のデータを生成/表示する「SetRow」プロシージャ

次のコードは、DataGridコントロールに表示する詳細部のデータをデータベースから取得するための連結テーブルクエリである (リスト2)。

```
Dim FieldListDetail() As String = {
    ("受注明細", "受注コード"),
    ("商品", "商品コード"),
    ("商品", "商品名"),
    ("受注明細", "単価"),
    ("受注明細", "数量"),
    ("受注明細", "割引")}
```

.NET FRAMEWORK IDEA NOTE