

レコードセット(ADODB.Recordset)の使い方

作成日: 2018/02/05

作成者: 西村

更新履歴

更新日	更新概要	作業者
2018/02/05	・ 新規作成	西村
	・	
	・	
	・	
	・	

目次

更新履歴	1
はじめに.....	3
レコードセット?	3
準備	5
SQL でデータを取得する	6
データのループ処理	7
列の値を取得する	7
対象行を変更する (MoveFirst, MoveNext, MovePrevious, MoveLast).....	8
グリッドへレコードセットを設定する (TDBGrid, DataSource)	9
データ行を追加する (AddNew).....	9
レコードセットをコピーする	10
レコードセットに列定義を行う (Fields.Append).....	11
レコードセットをソートする (Sort).....	12
レコードセットを絞込する (Filter)	12

はじめに

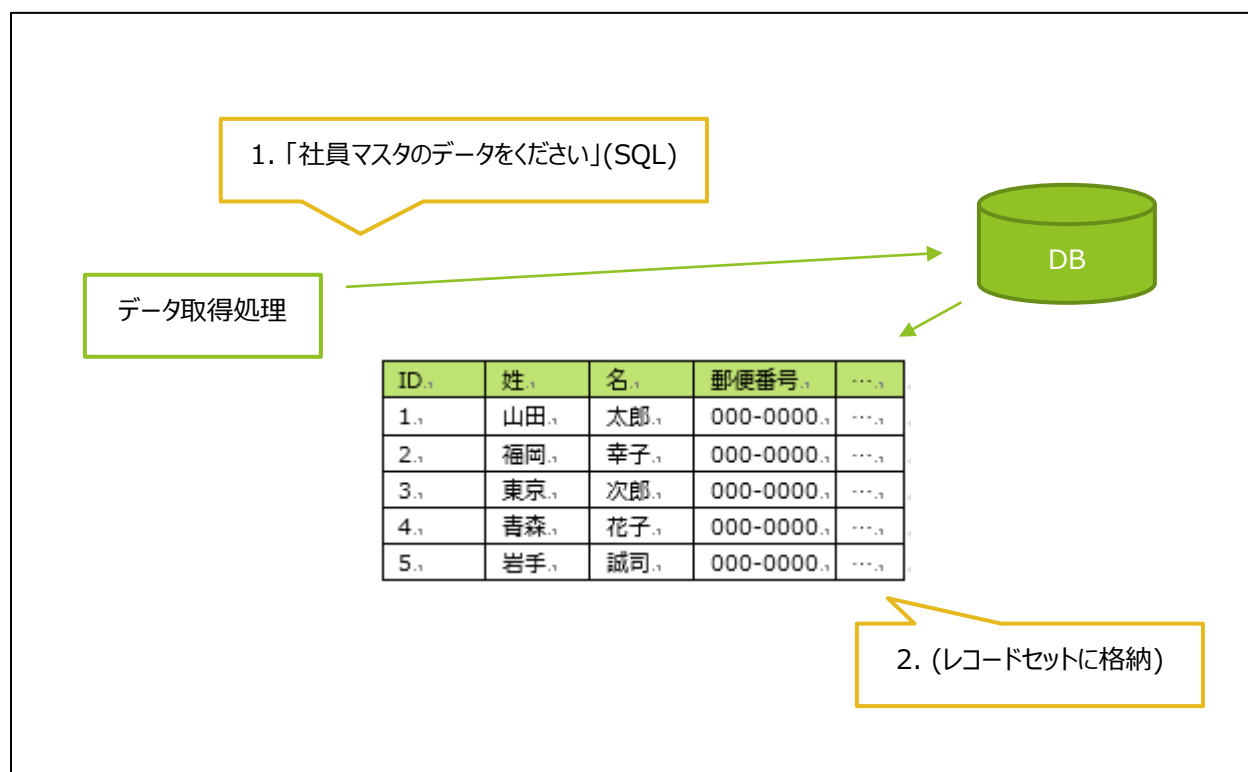
この資料では、VB6 で使用するレコードセット(ADODB.Recordset)の使い方について簡単に記載します。

レコードセット？

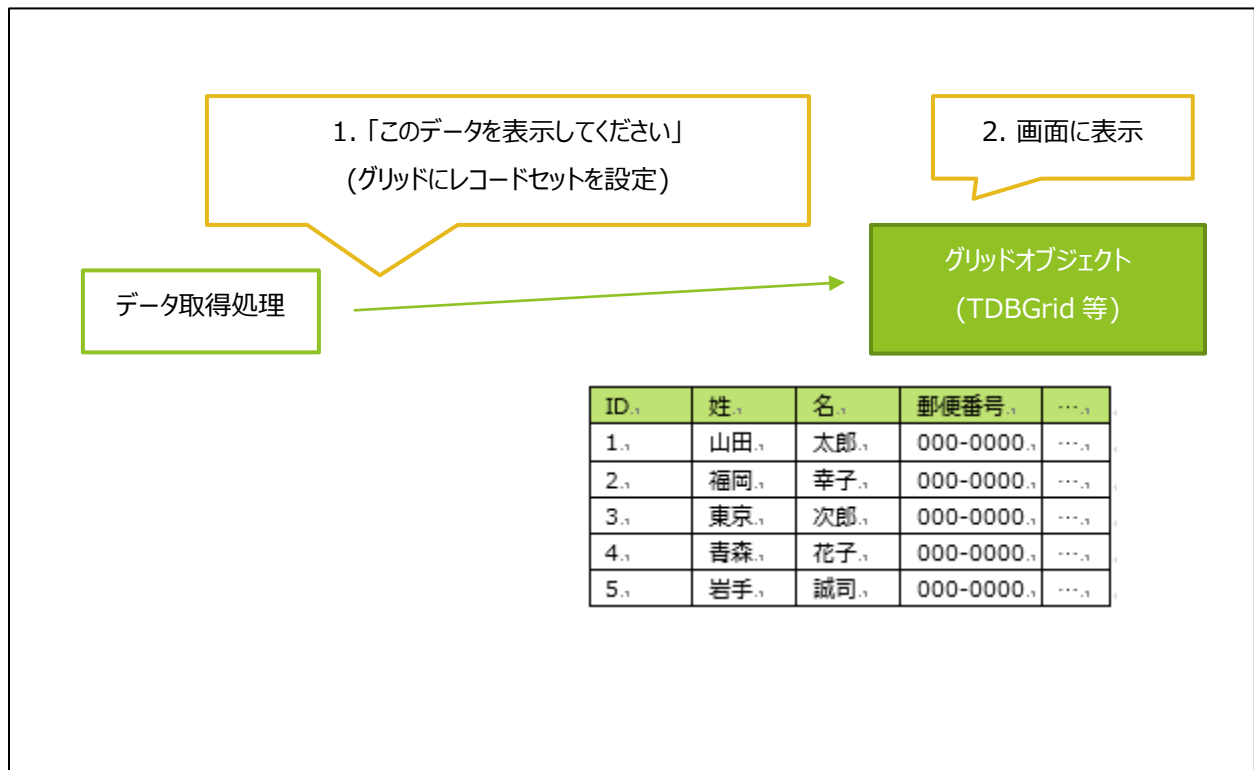
VB6 のレコードセットは、データベースから SQL でデータを取得するときに使用したり、データをグリッドに表示するときに使用したりする、表のようなデータ構造をもつオブジェクトです。

(C#や VB.NET でいうところの DataTable (System.Data.DataTable) などに近いものです)

例: データの取得イメージ



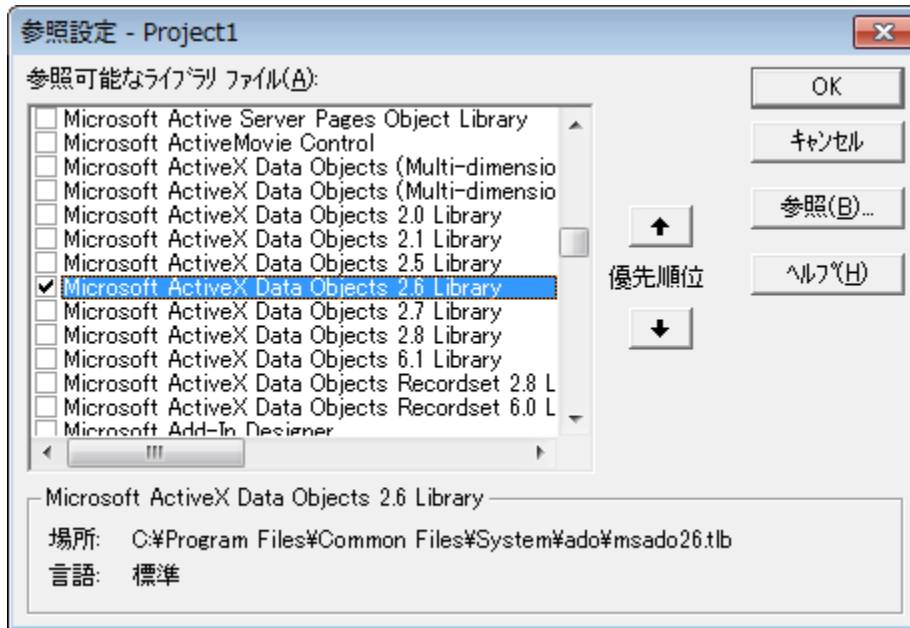
例: グリッドへの設定イメージ



準備

※ この項目は、既存システムの改造の作業をする場合は不要です。

レコードセットを使用するには、VB6 のメニューバー「プロジェクト」>「参照設定」を選択し、「Microsoft ActiveX Data Object x.x Library」にチェックしたのち、「OK」してください。



(2.6 を選んだ時の例)

SQL でデータを取得する

「ユーザーマスタテーブルからデータを取得して姓のみ出力したい」という場合、単純な例では下記ようになります。

- ※ この例では、DB は SQL Server を想定しています。
- ※ 既存システムの改造作業では、レコードセットの取得を行う処理が共通関数になっている場合がありますので、その場合は共通関数を使用してください。

```
Private Sub Command1_Click() ' ボタンを押したときの処理を想定
    Dim adoConn As New ADODB.Connection ' 接続オブジェクト
    Dim adoRs As New ADODB.Recordset ' レコードセット

    Dim connectionString As String ' 接続文字列 (必要なものに変更してください)
    connectionString = "Provider=SQLOLEDB.1;Initial Catalog=testdb;Data
Source=.%SQLXPTTEST;User ID=sa;Password=testpassword"

    Dim sSQL As String ' SQL
    sSQL = "SELECT * FROM ユーザーマスタ" ' ユーザーマスタテーブルから全データを取得する SQL

    adoConn.Open connectionString ' 接続オープン

    Set adoRs.ActiveConnection = adoConn ' レコードセットの接続元を指定
    adoRs.CursorLocation = adUseClient ' クライアントカーソルを設定
    adoRs.Open sSQL, adoConn, adOpenStatic, adLockReadOnly ' 読み取り専用で SQL を取得

    ' 取得したレコードのループ
    Do While Not adoRs.EOF ' 最終行までループ
        Debug.Print adoRs("姓") ' 列の値を取得して出力 (姓)
        adoRs.MoveNext ' 次の行へ移動
    Loop

    adoRs.Close ' レコードセットの終了 (これ以上使わない場合)
    adoConn.Close ' 接続オブジェクトの終了 (これ以上使わない場合)

    Set adoRs = Nothing ' 参照の解除 (これ以上使わない場合)
    Set adoConn = Nothing ' 参照の解除 (これ以上使わない場合)
End Sub
```

データのループ処理

データのループ処理の基本構造は下記のとおりです。

```
' 取得したレコードのループ
Do While Not adoRs.EOF ' 最終行までループ
    ' --- この部分で取得処理などを行う ---
    adoRs.MoveNext ' 次の行へ移動
Loop
```

最終行(EOF = End of File)になるまで次の行に移動(MoveNext)しながら処理を行う形です。

列の値を取得する

列の値の取得は、下記のいずれかの書き方で行うことができます。

※ Debug.Print はイミディエイトウィンドウと呼ばれるデバッグ用のウィンドウに値を出力するメソッドです。

```
Debug.Print adoRs("姓") ' 列の値を取得して出力 (姓)
Debug.Print adoRs!姓
Debug.Print adoRs![姓]
```

これは「現在の対象行の列の値を取得する」という意味なので、MoveNext をするたびに対象行が変わっていき、取得できる値もそれに応じて変わっていきます。

対象行を変更する (MoveFirst, MoveNext, MovePrevious, MoveLast)

対象行を変更するには、「MoveNext」「MoveFirst」といったメソッドを使用します。

関数	意味
MoveFirst	最初の行へ移動 (対象行を最初の行にする)
MoveNext	次の行へ移動 (対象行を次の行にする)
MovePrevious	前の行へ移動 (対象行を前の行にする)
MoveLast	最終行へ移動 (対象行を最後の行にする)

上記のような関数で対象行を移動すると、それに応じて Bookmark という項目の値も変わっていきます。Bookmark は、「現在の対象行が何番目の行か」を示す値です。

```
' 取得したレコードのループ
Do While Not adoRs.EOF ' 最終行までループ
    Debug.Print adoRs.Bookmark, adoRs("姓") ' 行番号と列の値を取得して出力
    adoRs.MoveNext ' 次の行へ移動
Loop
```

上記の例では、1 から順に番号が出力されます。

```
adoRs.Bookmark = 2
```

上記の例では、対象行を 2 番目の行にします。

グリッドへレコードセットを設定する (TDBGrid, DataSource)

グリッドの設定は、TDBGrid の場合は下記ようになります。

```
TDBGrid1.DataSource = adoRs 'レコードセットをグリッドに設定
```

- ※ TDBGrid は ComponentOne 社のサードパーティ製コンポーネントです。メニューの「コンポーネント」 > 「Component One True DBGrid Pro 7.0 (OLEDB)」を有効にしている必要があります。
- ※ 上記の「TDBGrid1」は、フォームデザイナー上に配置した TDBGrid の部品名の例です。
- ※ グリッドに設定するレコードセットは、開いた状態 (Open して Close していない状態) のままにしてください。閉じているとデータが表示されません。

データ行を追加する (AddNew)

データ行の追加は、「AddNew」というメソッドを使用します。

```
adoRs.AddNew 'レコードセットに行を追加
```

ただ、読み取り専用のレコードセットでは行を追加できません。また、読み取り専用でなくてもそのまま行を追加すると DB のテーブルに行が追加されてしまうため、「グリッドの表示を調整したい」程度の場合は、後述の「レコードセットのコピー」をしてから行の追加を行う必要があります。

レコードセットをコピーする

前述の「データの取得」では読み取り専用でデータを取得しましたが、この場合「レコードセット内に行を追加したい」と思っても行を追加することができません。行を追加する方法のひとつとして、「読み取り専用でない、DBとも紐付いていないレコードセットを作成する」、という方法があります。

```
Dim adoRsClone As New ADODB.Recordset ' 値をコピーするレコードセット
adoRsClone.Fields.Append "姓", adVarChar, 40, adFldIsNullable ' 列の設定
adoRsClone.Fields.Append "名", adVarChar, 40, adFldIsNullable ' 列の設定
adoRsClone.Open ' レコードセットを SQL なしで開く

' 取得したレコードのループ
Do While Not adoRs.EOF ' 最終行までループ
    adoRsClone.AddNew ' コピー先のレコードセットに行を追加

    adoRsClone("姓") = adoRs("姓") ' 値をコピー
    adoRsClone("名") = adoRs("名") ' 値をコピー

    adoRsClone.Update ' 値を保持
    adoRs.MoveNext ' 次の行へ移動
Loop

adoRsClone.AddNew ' 行を追加
adoRsClone("姓") = "山田" ' 最終行の値を設定
adoRsClone("名") = "太郎" ' 最終行の値を設定
adoRsClone.Update ' 値を保持

adoRsClone.MoveFirst ' 対象行を最初に戻してループ・出力
Do While Not adoRsClone.EOF
    Debug.Print adoRsClone.Bookmark, adoRsClone("姓"), adoRsClone("名") ' 列の値を出力
    adoRsClone.MoveNext ' 次の行へ移動
Loop
```

※ レコードセットには「Clone」というメソッドがありますが、Cloneして複製したのもDBと紐付いているため、新しいレコードセットを作って値をコピーしていくほうがよいと思います。

レコードセットに列定義を行う (Fields.Append)

前述の「レコードセットのコピー」のサンプルのように、レコードセットに列を定義するにはレコードセットの「Fields」というオブジェクトが持っている「Append」というメソッドで列が定義できます。

```
Dim adoRsClone As New ADODB.Recordset ' 値をコピーするレコードセット
adoRsClone.Fields.Append "姓", adVarChar, 40, adFldIsNullable ' 列の設定
adoRsClone.Fields.Append "名", adVarChar, 40, adFldIsNullable ' 列の設定
```

Fields.Append メソッドの引数

引数	意味
列名	adoRs("名") などで参照するための列名を指定します。
型名	列の値の型名を指定します。 例 adInteger → 整数 adNumeric → 数値(固定の精度を持つ小数点数) adVarChar → 文字列
サイズ	型のサイズを指定します。 DB の列が varchar(40) の場合、型名 = adVarChar, サイズ = 40 になります。
属性	追加の情報を指定します。 特に adNullable (NULL を許可) を指定することが多いです。

型名の一覧は <https://msdn.microsoft.com/ja-jp/library/cc389790.aspx> を参照してください。

レコードセットをソートする (Sort)

SQLを開いた後に、レコードセット上でデータのソートを行うことができます。

```
adoRs.Sort = "姓,名"
```

※ SQLのORDER BYに近いものです。

解除するときは空文字を設定します。

```
adoRs.Sort = ""
```

レコードセットを絞込する (Filter)

SQLを開いた後に、レコードセット上でデータの絞込を行うことができます。

```
adoRs.Filter = "姓='山田' AND 名='太郎'"
```

※ SQLのWHEREに近いものです。

解除するときは空文字を設定します。

```
adoRs.Filter = ""
```

レコードセット内に特定の行があるかを調べる際に、Filter をセット→行の値を取得→Filter を解除、という処理を行うことが時々あります。