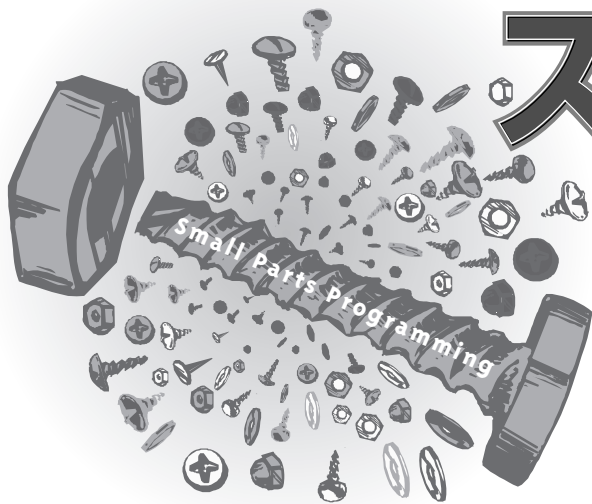


VB.NET

コンポーネント活用講座
～ 教養課程 ～

スモールパーツ プログラミング



第 9 回

小森 大輔 KOMORI, Daisuke

ListBox コントロール ～ Part 1 ～

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¥SMALLディレクトリに収録しています。

¥ITEMSADD
オブジェクトアイテムの追加、BeginUpdateメソッドの使用例

¥ITEMSADDRANGE
配列からアイテムの追加、Insertメソッドの使用例

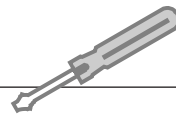
¥REMOVE
Clear、Remove、RemoveAtメソッドの使用例

¥RANDOM
Randomクラスの使用例

¥DATASOURCE
DataSourceプロパティの使用例

¥DATASET
DataSetを使用したデータの利用例

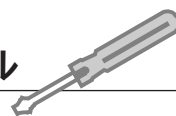
はじめに



本連載は、Visual Basic .NET (以下VB.NET) の標準コントロールにスポットを当て、とことん使い込んで、その機能や役割を探ってみます。

今回は、アプリケーションを使用するユーザーに、項目を選択してもらうようなUIで威力を発揮する、ListBox コントロールを主役に据えてお話しします。

List Box コントロール



List Box コントロールは、ユーザーが選択できる項目のリストを表示することができるコントロールで、図1のようなクラス階層になっています。

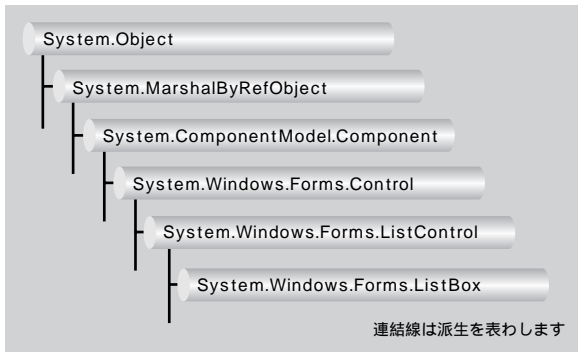
ある規則的な値などの場合、ユーザーにテキストボックスへ入力してもらうのではなく、List Box コ

ントロールへ表示した項目（以降“アイテム”と呼びます）から選択してもらうようなUIにすれば、ユーザーとしては操作が1クリックで済むことになります。

また、ユーザーにある決まった値をテキストボックスなどで入力してもらうと、入力された値の整合性検査が必要です。そのようなときにList Box コントロールを使用すれば、開発者側の意図した通りの値が得られることは言うまでもありません。このようなことから、このコントロールもかなり使用頻度の高いコントロールなのではないでしょうか？

「ユーザーにアイテムのリスト中からひとつあるいは複数のアイテムを選択してもらう」という、List Box コントロールのスタイルは、RadioButton コントロールやCheckBox コントロールと同様ですが、List Box コントロールは、次のような点で、RadioButton コントロールやCheckBox コントロールとは異なる

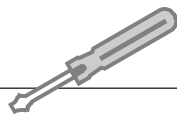
図1 : ListBox階層図



ります。

- ・リスト内のアイテムの追加や削除を容易に行なうことができる
- ・コントロール単体でスクロールの機能があり、多数のアイテムの表示を行なうことができる
- ・コントロール単体がアイテムリストのグループになっている

アイテムの追加



今回はListBox コントロールについての細かい説明を行なう前に、まずは基本的な使い方であるアイテムの追加と削除について説明しておきます。

まずはじめに「アイテムの追加」についてみてみましょう。

Listbox コントロールへのアイテムの追加は、リストボックス内のアイテムのコレクションを表わすObject Collection クラスがもつ各メソッドを使用します(表1)。ObjectCollection クラスは、Items プロパティで参照することができます。

◆ Add メソッド

Add メソッドは、リストボックスのリストにアイテムを追加します。

Add メソッドは、

表1 : ObjectCollection クラスがもつアイテムを追加するメソッド

メソッド	機能
Add	リストボックス内にアイテムを追加
AddRange	配列の要素をリストボックス内のアイテムとして追加
Insert	リストボックス内に、アイテムのインデックス位置を指定してアイテムを挿入

Listbox1.Items.Add("")

のように使用します。

Add メソッドは、表示するアイテムをパラメータとしますが、これは文字列だけではなく、Object 型の値を指定することができます。

この場合ListBox コントロールは、そのオブジェクトのToString メソッドを呼び出し、アイテムとして表示するテキストを取得します。

たとえばリスト1は、String 型配列をListBox コントロールへアイテム (Item オブジェクト) として追加し、その後そのアイテムを参照してString 型配列へ格納、配列の値を出力する例です。図2は、このサンプルを実行した画面と、出力された値です。

Add メソッドを使用してアイテムをリストに追加すると、アイテムが追加されるたびにリストボックスは再描画されます。Add メソッドで多数のアイテムをリストに追加する場合には、BeginUpdate メソッドとEndUpdate メソッドを使用して、すべてのアイテムがリストに追加されるまでリストボックスを再描画しないようにすればパフォーマンスが向上します。

リスト2は、リストボックスへ10,001個のアイテムを追加し、BeginUpdate メソッドを使用した場合と使用しなかった場合の、経過時間を出力するサンプルです。図3は、このサンプルの実行画面と出力結果ですが、これを見てわかるように、私の一世代前のマシンでは約3秒もの違いがありました。BeginUpdate は使っておいて損はないメソッドだと思います。

◆ AddRange メソッド

AddRange メソッドは、指定した配列の要素をアイテムのリストに追加します。リスト1のAdd メソッドを使