

赤尾の “現場で勝てる” Visual Basic

第11回

6.0

自分で作るActiveXコントロール

赤尾 猫太 AKAO, Nekota

その1

「コントロールの作成とテスト」「コントロールのイベント」
「プロジェクトグループ」



はじめに

Visual Basic（以下VB）を起動すると図1のように、さまざまな形式のプログラムを作成できるようになります。今回からは、ActiveXコントロールを作成します。

ActiveXコントロールもWordやExcelなどのVBA（Visual Basic for Applications）も、基本的には今までに本連載で学習してきたこと的应用で作成できます。それぞれの形式の目的を理解して、徐々に作成できるものの幅を広げてゆくとよいでしょう。

まずは復習をしつつ、ActiveXコントロールの作成目的などについて考えてみましょう。



ActiveXコントロール

本連載では、今までにサブルーチンを作成しました。サブルーチンとは、“複数箇所ですらう処理を塊にして他の部分でも使えるようにするもの”でしたね。しかし、サブルーチンは、ソースコードレベルでしか使用できません。もっとも、自分のプログラムであればとくに問題はありませんし、ソースコードレベルであるからこそ、サブルーチンを流用して機能の追加ができるとも言えます。

しかし、他の人に“プログラムを提供する”のではなく、“プログラム作成のための機能を提供する”には不向きです。たとえば特殊な機能の場合、機能を他の人に

本稿で前提となるもの

OS Windows 98以上

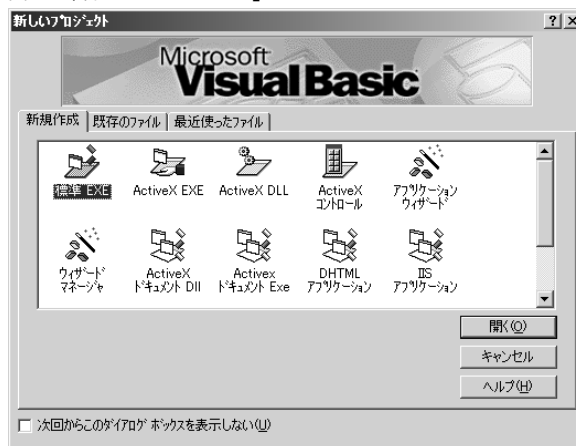
開発環境 Visual Basic 5.0/6.0



この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥AKAOフォルダ以下に収録しています。

- ¥CONTROL1 : 今回作成した自作コントロール
- ・GROUP1.VBG : 「年月日」と「時分秒」を表示する自作コントロール
- ・GROUP2.VBG : コモンダイアログのような自作コントロール

図1: 「新しいプロジェクト」ダイアログ



提供するのによくても、その作り方（ソースコード）は提供したくないという場合もあります。利用目的はそれぞれでしょうが、「ソースコードレベル以外でも使用できるサブルーチンのようなもの」の作り方を覚えておくとよいことはおわかりですね。

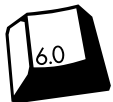
では、「ソースコードレベル以外でも使えるサブルーチンのようなもの」とは、どんなものなのでしょうか。本連載では以前（本誌2002年12月号）、コモンダイアログコントロールを使用しました。コモンダイアログコントロールのように、ソースコードレベルではなく別ファイルとして存在する機能の塊がActiveXコントロールだと思ってください。

今回から2回にわたって、独自のコントロールを作成してみます。

■ コントロールに必要な機能

まずは、コントロールの機能として、どんなものがあったか、どんなものが必要なのかを考えてみましょう。別ファイルではありませんが、VBのツールボックスにはテキストボックスなどのよく使うコントロールが用意されていますね。そして、コントロールには「イベント」「プロパティ」「メソッド」などの機能が存在します。これらの機能を実現しなければなりません。

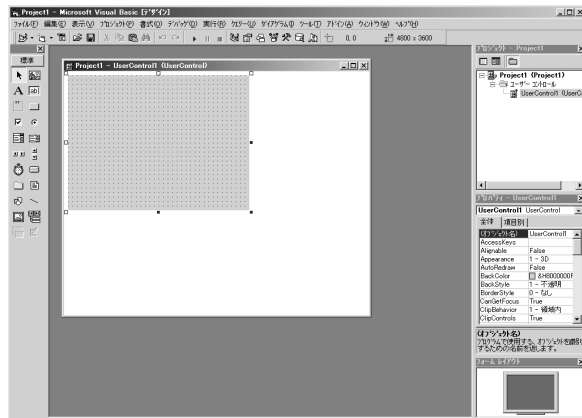
しかし、コントロールは、これらを使用して呼び出しているプログラムに機能を提供するだけなので、「イベント」「プロパティ」「メソッド」のすべてが必要とは限りません。必要なものを用意するだけでよいのです。



6.0 コントロール作成の基本

さて、つまらない話はこれぐらいにして、早速作り始めてみましょう。まずは、VBを起動して、新規のプロジェクトとして [ActiveXコントロール] を選択してください。すると、図2のような画面になるはずです。ひと目で違いはわかりますね。このタイトルバーのないフォームのようなものの上に、コントロールとして必要となる機能を作成してゆくのです。

図2：「ActiveXコントロール」プロジェクトのデザイン画面



■ コントロールと標準EXEのイベント

では、練習として、ごく簡単なコントロールを作成してみましょう。まずは、図3のようにラベルコントロールを2つとタイマーコントロールをひとつ貼り付けます。

続いて、コードエディタに切り替えてリスト1のコードを記述します。

最初にUserControl_Initializeという見慣れないイベントが出てきました。これは、標準EXEのフォームにおけるForm_Initializeと同じような意味をもつイベントで、このコントロールの初期化処理を行なうためのものです。

今まで扱ってきた標準EXEのフォームでは、フォームがロードされる過程でInitializeイベント、Loadイベントが順に発生し、プログラムが開始されるときにResizeイベントが発生します。逆に、フォームがアンロードされるときには、QueryUnloadイベント、Unloadイベント、Terminateイベントが順に発生します。今まではInitializeイベントやTerminateイベントは使わずに、LoadイベントやUnloadイベントなどでプログラムを作成してきましたが、厳密にはInitializeイベントにはロード（Load）前に行ないたい処理を実装し、Terminateイベントにはア

図3：作成するコントロールのデザイン

