



.NETコントロールクラス 実践講座

コンポーネントと継承を上手にを使って楽々プログラミング

初音 玲 HATSUNE, Akira

はじめに

Visual Basicではコントロールと呼ばれる部品をフォームに貼り付けて画面を作成してゆく。コントロールは画面デザインだけではなく、そのデザインに関連したすべてのロジックを格納で

きるので、複雑な画面構成であっても対応するコントロールがあれば、実際に記述するコードは驚くほど少量です。また、Visual Basic 5.0以降では、COMコントロール（ActiveXコントロール）を独自に作成することができるので、画面の共通部分をCOMコントロール化することで画面デザインおよび

画面操作ロジックの共通化も可能になった。Visual Basic.NETでももちろんコントロールの作成は可能だが、作成できるコントロールは、COMベースの技術で動作するCOMコントロールではなく、.NET Frameworkベースの技術で動作するWindowsコントロールだ。もちろん、.NET Frameworkベースなので、Visual Basic.NETで作成したWindowsコントロールを、他の言語、たとえばC#から利用することも、その逆のことも何の問題もなく実現できるようになっている。

本稿で前提となるもの

OS	Windows 2000 Professional (SP3) 以降
開発環境	Visual Studio.NET .NET Framework 1.0.3705.288 (SP2) Visual Basic .NET Compiler 7.00.9466 Internet Explorer 6.0.2800.1106



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¥F01_03フォルダ以下に収録しています

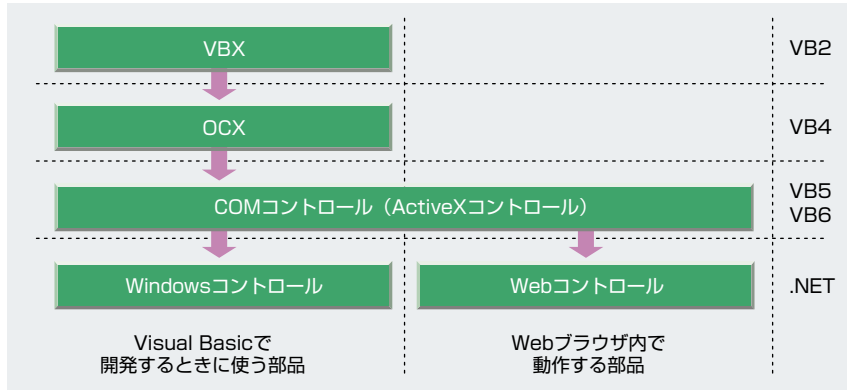
¥TEXTBOXTEST	: イベント記述サンプル
¥FOCUSTEXTBOX	: 標準コントロール継承
¥WINHSCRLABEL	: UserControl作成例
¥FORMSTANDARD	: Windowsフォーム継承

COMコントロールから Windowsコントロールへ

VBX

Visual Basicの特徴のひとつが、拡張子がVBXのファイルをプロジェクトに取り込むことによって簡単に言語仕様

図1：コントロールの沿革



の機能拡張が行なえることであった。また、VBXの作成方法が公開されたため、マイクロソフト社以外のベンダーによるコントロールの作成が可能となり、市販コントロール市場が形成され魅力的なコントロールがたくさん送りだされた。

この魅力的なコントロール製品がVisual Basicの普及に果たした役割は大きい(図1)。ただし、Visual Basic自体でVBXを作成することは不可能で、別途C言語などを使って作成する必要があった。また、VBXは16bit時代の技術であるので、Visual Basic.NETで使用することはできない。

OCX

マイクロソフト社がOLEベースの技術に移行してゆく中で、Visual Basic 4.0では(実際は日本未発売のVisual Basic 3.0から)、拡張子が「VBX」のコントロールから拡張子が「OCX」のカスタムコントロールへと切り替わった。

OLEはプロセス間通信の一形態で、VBXが採用していた技術よりも速度面や安定性の面で優位であり、OCXにシ

フトすることで市販コントロールを使ったときの速度や安定性も向上することになった。OCXについてもVisual Basicで作成することは不可能で、作成するためにはVisual C++などを使う必要があった。また、Visual Basic 4.0対応の製品はコンテナ(OCXを利用する側)がマルチタスクで動作したときの対応が行なわれていないため、Visual Basic.NETで使用することは事実上できない。

COMコントロール

Visual Basic 5.0では(拡張子は「OCX」のままではあったが)、カスタムコントロールはActiveXコントロールという呼び名に変わった。これは、マイクロソフト社がインターネットに対応する製品群をActiveXという名称で世に送り出したことに関連する。つまり、ActiveXコントロールとは、Javaアプレットに対抗するために開発された技術で、Visual Basicから使うだけではなく、Internet Explorer上で使うこともできるようになったコントロールのことなのだ。

ただし、インターネット対応として、セキュリティの強化や軽量化(MFCではなくATLをライブラリとして使うなど)は行なわれているが、その実態はOCXベースのカスタムコントロールの延長線上に位置している。ちなみにActiveXという名称はしだいに使われなくなったために、一般的にはOCXの仕組みであるCOM(Component Object Model)を意識して「COMコントロール」と呼ばれることも多く、本特集でもCOMコントロールと表記している。

COMコントロールを採用しているVisual Basic 5.0には、それを作成する機能も追加された。やっとVisual Basic自身でコントロールが作成できるようになったわけだ。もちろん問題がなかったわけではなく、Visual Basicで作成したCOMコントロールにはVisual Basicのランタイムルーチンが必要であったり、Visual Basicが内部的にATLではなくMFCを使用していたりしたためにVisual C++で作成したCOMコントロールに比べてサイズや速度面で使い勝手が悪く、最終的にはVisual Basicで作成したCOMコントロールはVisual Basicでのみ使うという形態に落ち着くことが多かったようだ。

COMコントロールは、COMで接続できるコンポーネントであるので、Visual Basic.NETでも図2のような形態をとればWindowsコントロールとして利用可能だ。ただし、Visual Basic 5.0 SP2以降対応の市販COMコントロール、またはVisual Basic 5.0 SP2以降で作成したものでないと、コンテナ(COMコントロールを利用する側)がマルチタスクで動作したときの動作保

