

初音 玲
HATSUNE, Akira

語られることのない VC++.NETの進化を解剖

マネージドとアンマネージドに架ける橋

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 - Visual C++ .NET

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¥F03ディレクトリに収録しています。

- ¥0311VC01
VC++.NETフォームサンプル
- ¥0311VC02
VC++.NETクラスサンプル
- ¥0311VC03
VC++.NET複数プロジェクトサンプル
- ¥0311VC04
VC++.NETライブラリサンプル
- ¥0311VBVC
VB.NETからVC++.NETライブラリを利用するサンプル

はじめに

Visual C++は、Visual Basicのようにフォームにボタンを配置してGUIを作成するような操作性を伴っていなかったが、Visual C++ .NET 2003（以下VC++.NET）では、この点が改善されており、.NETアプリケーションであれば、GUIの作成をVB.NETと同じようにMicrosoft Development Environment (MDE) で行なうことができる。

VC++.NET理解の キーポイント

VC++.NETを理解するうえでキーポイントになるのは、「マネージ」と「アンマネージ」というものの存在だ。

マネージとアンマネージ

VC++.NETは、VC++に.NET Framework上で動作する「マネージコード」を生成できるように、マネージ拡張という独自拡張を行なっている。マネー

ジ拡張を使った場合、コンパイルして得られる実行ファイルは、.NET Frameworkのアセンブリであるポータブル実行ファイル (PEファイル) となる。このPEファイルは、実行時に.NET FrameworkのJITコンパイラによって初めてOSネイティブの実行コードにコンパイルされる。

一方、アンマネージコードと呼ばれる、マネージ拡張された構文を使わないものは、ATLやMFCなどの従来どおりのランタイムルーチンを使ったり、Windows APIを直接使ったりしてプログラミングを行なう。そして、そのコンパイル結果は、OSネイティブの実行ファイルとなる (図1)。このアンマネージコードの生成機能は、従来のソースコードを使って、.NET以前の従来のWindowsアプリケーションを作成できるもので、マイクロソフト社の他の.NET言語にはない特徴だ。

MDEでの取り扱いの違い

今まで、VC++でのWindowsアプリケーション作成の敷居が高かった理由

のひとつに、「MFCやATL、Windows APIなどを使ってすべてコードでフォームの生成やコントロールの配置をしなくてはならなかった」ということがある。しかし、VC++.NETでは、Visual Basicのように、

MDE上でフォームにコントロールを貼り付けてプロパティを設定したあと、イベントに対応したコードを記述して作成する

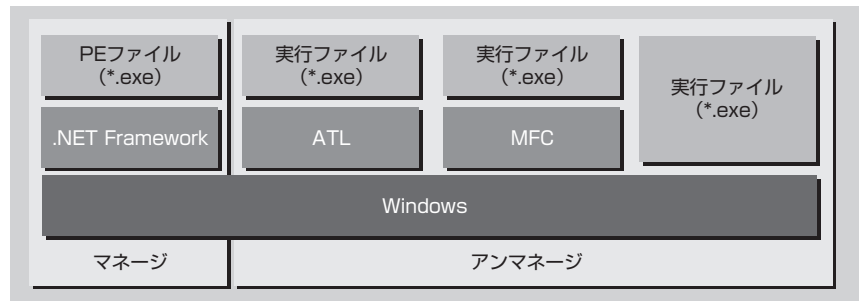
という操作性を取り入れ、GUIが格段に作成しやすくなっている。

しかし、この操作性の向上は、あくまでもマネージコード、つまり、.NET Framework上で動作するアセンブリを作成するときに限られ、アンマネージコードを作成するときには、従来のように、すべてコードで記述しなければならない。

フォームにコントロールを貼り付けた結果と.NET Frameworkを使ったコードとの相互変換が可能なのだから、同じように、MFCやATLを使った形式との相互変換も実現できそうなものだが、現在のバージョンではまったく対応していない。

なお、今回は、VC++.NETになって、実際にアプリケーションを作成する手順や、そのソースコードがどう変わったかをみてゆこうと思う。従来と変わらないアンマネージコードに関しては触れず、マネージコードに関する話題に限定して話を進めてゆこう。

図1：使用ライブラリの違い



Windows アプリケーションの作成

MDEの起動

Windowsの[スタート]メニューからメニューをたどり、Visual Studio .NET 2003を起動すると、「Microsoft Development Environment [デザイン]」画面が立ち上がる。

新しいプロジェクト

MDEのスタートページで、「新しいプロジェクト」ボタンをクリックし、「新しいプロジェクト」ダイアログから「.NET」フォルダの「Windowsアプリ

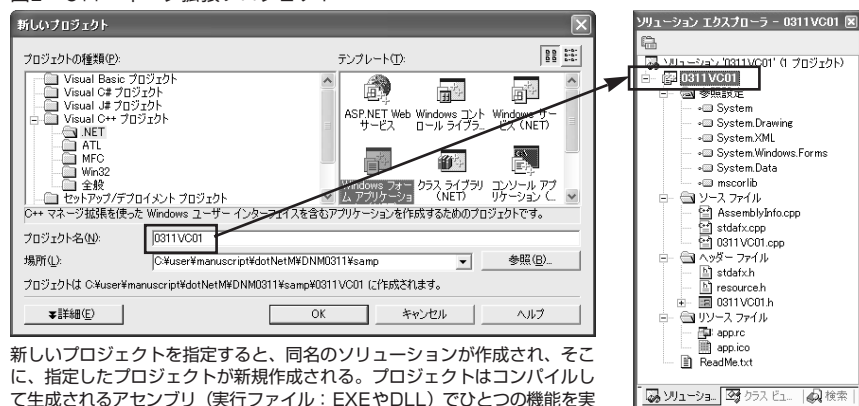
ケーション」を選択すると、自動的にフォームをひとつ持ったアセンブリ(実行ファイル)のプロジェクトが作成できる(図2)。

コードを記述する場所

MDEでフォームにコントロールを貼り付けた結果などは、すべてヘッダファイルのほうに記載するような流れになっている。なぜ、ソースファイルではないのか釈然としないが、それにあわせてサンプルはすべて、ヘッダファイルにコードを記述している。

なお、プロジェクトの初期状態では、フォームに関連したファイルは、「Form1.cpp」と「Form1.h」となっている。

図2：C++ マネージ拡張プロジェクト



新しいプロジェクトを指定すると、同名のソリューションが作成され、そこに、指定したプロジェクトが新規作成される。プロジェクトはコンパイルして生成されるアセンブリ(実行ファイル: EXEやDLL)でひとつの機能を実現する単位。通常1ソリューション=1プロジェクトの構成で事足りることが多い。