

# .NET Framework

# ネームスペースの旅

すべての根源よりすべてのクラスに向けて

Episode

3

## System.Windows.Forms ネームスペース編

グレーシティ株式会社  
アドバイザリースタッフ  
矢沢 久雄 YAZAWA, Hisao



### はじめに



これまで堅苦しいネームスペースの説明が続いてしまいましたので、今回は、ビジュアルで楽しいWindowsアプリケーションを作成するためのクラスを提供する「System.Windows.Forms」ネームスペースで遊んでみましょう。

Visual Studio.NETで作成できるプログラムの種類は、インターネット関連のXML WebサービスやWebアプリケーションだけではなくありません。従来どおりのWindowsアプリケーションも作成できます。ただし、その作成方法は、MF Cを使ったVisual C++ 6.0や、ActiveXコントロールを使ったVisual Basic 6.0とは大きく変わっています。Visual Basic

.NET (以下VB.NET) やVisual C# (以下C#) では、.NET Frameworkが提供するまったく新しいクラスライブラリを使うのです。



### System.Windows.Forms ネームスペースの概要



Windowsアプリケーションを作成するために使われるフォームやコントロールなどは、System.Windows.Formsネームスペースのクラスとして提供されています。このネームスペースには、膨大な数のクラス、構造体、列挙型があります。とてもではありませんが、短い誌面の中ですべてを紹介することはできません。表1に主なクラスだけを示しますので、全体的な雰囲気をつかんでください。フォーム、コントロール、ダイアログボックス、メニューなどが独立したクラスとして提供されていることに注目してください。

すべてのWindowsアプリケーションのフォームに共通する機能は、Formクラスに汎化 (共通点をまとめること)

#### 本稿で前提となるもの

OS Windows XP (SP2)  
開発環境 Visual Studio.NET  
.NET Framework 1.0.3705.288 (SP2)

初級

中級

上級

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¥NAMESPACEフォルダ以下に収録しています。

¥BMINWINAPP : 最小限のWindowsアプリケーション (VB.NET版)  
¥CSMINWINAPP : 最小限のWindowsアプリケーション (C#版)  
¥VBFORM : 透明でスクロール可能なフォーム (VB.NET版)  
¥CSFORM : 透明でスクロール可能なフォーム (C#版)  
¥VBADDCCTRL : フォームにコントロールを貼り付ける (VB.NET版)  
¥CSADDCCTRL : フォームにコントロールを貼り付ける (C#版)  
¥VBEVENT : コントロールのイベント処理 (VB.NET版)  
¥CSEVENT : コントロールのイベント処理 (C#版)



表1：System.Windows.Forms ネームスペースの主なクラス

クラス名	機能
Application	Windowsアプリケーションを起動および終了する
Form	フォームを提供する
Button、TextBox、Label、CheckBox、ToolBar、StatusBar	各種のコントロールを提供する
Menu、MainMenu、MenuItem、ContextMenu	メインメニューおよびポップアップメニューを提供する
FontDialog、FileDialog、ColorDialog、PrintPreviewDialog	各種のダイアログボックスを提供する
MessageBox	メッセージボックスを提供する
Clipboard、Timer、Cursor、Help、ToolTip	クリップボード、タイマー、カーソル、ヘルプ、ツールチップを提供する
Control	Formクラスや各種のコントロールのベースクラス
ScrollableControl	Formクラスや各種のコントロールのベースクラス
ContainerControl	Formクラスや各種のコントロールのベースクラス

されています。したがって、Windowsアプリケーションは、Formクラスを継承したクラスとして容易に作成できます。リスト1は、最小限のWindowsアプリケーションです。実行するとフォームが表示されます(図1)。システムメニュー、最大化ボタン、最小化ボタン、終了ボタンをもったウインドウを描画する機能、および境界線をドラッグしてサイズ変更する機能などは、す

図1：リスト1の実行結果



リスト1 最小限のWindowsアプリケーション

**VB.NET**

```
Imports System
Imports System.Windows.Forms

Public Class Class1
    Inherits Form
    Public Shared Sub Main(ByVal args As String())
        Application.Run(New Class1())
    End Sub
End Class
```

**C#**

```
using System;
using System.Windows.Forms;

public class Class1 : Form
{
    public static void Main(string[] args)
    {
        Application.Run(new Class1());
    }
}
```

べてFormクラスが提供してくれます。

リスト1の内容は、実にシンプルです。Formクラスを継承したClass1クラスがあります。Class1クラスのメンバは、静的メンバであるMainメソッドだけです。このメソッドの中で、ApplicationクラスのRunメソッドを呼び出し、その引数としてClass1クラスのインスタンスを与えています。これによって、Formクラスから継承した多くのメンバをもつClass1クラスのコピーがメモリにロードされ、Windowsアプリケーションとして実行されるのです。



## Applicationクラスの機能と使い方

Windowsアプリケーションを起動してくれるApplicationクラスの機能と使い方を説明しましょう。このクラスには、多くのメンバがありますが、その中でもとくに重要なものを厳選して表2に示します。すべて静的メンバ(インスタンスを作成せず「クラス名.メンバ名」という構文で使うメンバ)です。

アプリケーションの企業名、製品名、バージョンは、プログラムの属性(コンパイル後のEXEファイルやDLLファイルに付加される任意の情報)として設定できます。Visual Studio.NETを使ってプロジェクトを作成すると、VB