



Object BLVD オブジェクトの散歩道

例題でわかる！ .NET Framework

第4回 「やさしいリモートオブジェクト」

吉田 弘一郎 YOSHIDA, Koichiro

「.NETリモータリング」

本連載わずか4回目にして、突如リモートオブジェクトが登場しました。ネットワークのモダンな利用法です。これまでの3回が非常に基本的かつ初歩的な話題に終始していたのに、いきなり「.NETリモータリング」ですから、本誌編集部の方々も面食らったのではないかと思います。実際、「.NETリモータリング」は従来の「DCOM」を置き換えるものですから、かなり高度なプログラミングに属するはず。しかも、

DCOMといえばCORBAの対抗馬でしたね。そのCORBAは、Javaでエンブラする際にもはやされている例の複雑怪奇な代物。そんなネットワーク技術を凌駕する「.NETリモータリング」が、簡単であるとは思えません。少々簡単になったところで、決して本連載第4回にふさわしいレベルになるはずありません。そもそも、DCOMとかCORBAの類は、いまだに「最新の技術」であるかのような扱いではありませんか。しかも、CORBAの場合には、複雑怪奇に加えて金もかかるという二重苦の世界。何がどうなっているのでしょうか。

小さなサンプルで .NETリモータリング

インターネットに加速された分散プログラミングには話題は事欠きません。いわゆるエンタープライズうんぬん、eBusinessうんぬんの世界です。しかしDCOM、CORBA、RMI……、などという類の雑学は、本連載の読者のみなさんには決して求められていませんのでご安心ください。みなさんを初心者扱いするのではなく、DCOMやCORBAがすでに時代遅れの代物になっているからです。「.NETリモータリング」の登場で、分散処理うんぬんの世界は長足の進歩を遂げ、従来の技術は一夜にして時代遅れの産物と化したのです。論より証拠、今月はみなさんに「.NETのリモータリング」していただきます。まず今月の例題のリストを見てください。いつもより短く簡単であることがわかります。今月、これだけの例題をこなせば、みなさんにご自分で結構見栄えのする「.NETのリモータリング」プログラミングができるようになるのです。

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)

Internet Explorer 6.0.2800.1106

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¥SAMPOフォルダ以下に収録しています。

SAMPO04.SLN：今回紹介したサンプルのソリューションファイル

例題1：ローカルにCalc (CS01.cs)

今月はCalcという電卓用のクラスを例に用います。ただし、話を簡単にするために、この電卓には足し算の演算機能しかありません。Calcの中身は次のように非常に簡単なものです。

```
class Calc
{
    public double Add(double x,
                      double y)
    {
        return x+y;
    }
}
```

ここで、このクラスが非常に大規模なものになっても、話の筋はまったく変わりませぬので、今月はこれで我慢してください。

この電卓クラスの使い方は次の通りです。

```
Calc calc = new Calc();
double z = calc.add(1.23, 2.34);
```

Calc型オブジェクトcalcを生成した

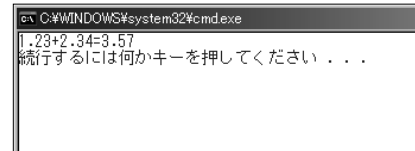
ら、これに対してメソッドaddを用いればよろしい。それでは、このCalcクラスとその使い手（クライアント）をまとめて例題1とします。リスト1をご覧ください。何の問題もないと思います。

実行結果は図1のとおり。

先月まではプリント時の書式を指定してきました。しかし、今月はリモートに焦点を当てるため、書式は気にしないことにします。

さて、この例題では、電卓をローカルに生成しています。ここで、この電卓が、実は単なる電卓ではなく、大規模なデータベースにアクセスしたり、あるいは特殊なハードウェアにアクセスするために、特定のサーバー上でしか動作しないというような状況を考えてください。しかし、その電卓を用いるクライアントプログラムは、あちこちのWindows機で走る必要があるとすれば、サーバーとクライアントの間には必然的にネットワークが介在し、それを支えるカラクリが「.NETリモート」なのです。それでは、段階を

図1：例題1の実行結果



追ってリモートにすることにしてしましましょう。

例題2：Calcのライブラリ化 (CS02calc.cs)

Calcライブラリ

「.NETリモート」への第一歩は、クラスライブラリの作成です。あるいは、クラスライブラリがあれば、それは容易にリモート化できると理解してもよろしい。先の例題からCalcクラスの部分だけを抜き出してクラスライブラリを作ります。Visual Studio .NETで新規プロジェクトを作成する際にテンプレートで「クラスライブラリ」を選択して枠組みを作ればよろしい。

ここで要注意なのは、クラスをPublicにすることです（リスト2）。

リスト1：例題1（ローカルにCalcクラスを用いるコンソールアプリケーション）

```
using System;

namespace CS01calc
{
    /// <summary>
    /// Calc：足し算専用電卓クラス。ここではローカルに
    /// しか用いないので、Publicにしなくてもよい
    /// </summary>
    class Calc
    {
        /// <summary>
        /// 実数の足し算、x+yを返す
        /// </summary>
        public double Add(double x, double y)
        {
            return x + y;
        }
    }
}

/// <summary>
/// Calcを用いるクラス、すなわちクライアント
/// </summary>
class Class1
{
    /// <summary>
    /// アプリケーションの主関数
    /// </summary>
    [STAThread]
    static void Main(string[] args)
    {
        // Calcオブジェクトを生成
        Calc calc=new Calc();
        double x=1.23;
        double y=2.34;
        Console.WriteLine("{0}+{1}={2}", x, y, calc.Add(x,y));
    }
}
```