



Object BLVD オブジェクトの散歩道

例題でわかる! .NET Framework

最終回 「Print」

吉田 弘一郎 YOSHIDA, Koichiro

Technology Tools

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

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¥SAMPOディレクトリに収録しています。

・SAMPO12.SLN
今回紹介したサンプルのソリューションファイル



MS Blaster騒ぎ

MSBlasterワームは、結構な騒ぎになりました。私も、これを機会と心得、10台ほどのWindows機にサービスパックをインストールし、パッチを当て、Anti Virusなどもすべて更新し、ファイアウォールの設定を再確認して、結局数日潰してしまいました。おかげで被害は何もなかったのですが、私の周囲には悲鳴を上げている初級ユーザーが何人もいます。Norton AntiVirusさえあれば絶対に大丈夫と信じ、DSLモデムとラウターの区別もつかないような方々です。

コンピュータも、ここまでネットワークが絡むと「無知ゆえの事故」が無視できなくなり、自動車を運転するのと似たような感覚が必要であるように思えます。Webサーバーに関しては、「MicrosoftのIISはオートマチックの自動車のようなもので誰でも即動かせる。Apacheはマニュアルの自動車で、動かすまでが大変だ」と言われていますが、OSそのものにも当てはまりそうです。オートマの自動車でも、最低限の運転

技術は必要なのです。

それでは、私のセキュリティ対策はといえば、セキュリティ門外漢の典型的で地味なものなのですが、あえて紹介させていただきます。

対策 0 DSLの場合にはラウターを追加。

対策 1 Norton AntiVirusの定義ファイルをこまめに更新。

対策 2 Microsoft Baseline Security Analyzer^[注1]で頻繁にチェック。これは非常に便利なツールなのですが、日本での知名度は低いようです。日本語版Windowsでも大丈夫。とにかく、このツールで診断することからすべては始まるように思えます。

対策 3 Microsoft Windows Update。これもBaseline Security Analyzerの診断結果に従い、処置を施します。

対策 4 ファイアウォールソフトウェア。私の周囲はZoneAlarm^[注2]が主流(BlackIce派も少々)です。アメリカでは

注1) <http://www.microsoft.com/technet/security/tools/Tools/MBSAhome.asp>

注2) ZoneLab社=<http://www.zonelabs.com/>

非常に知名度が高いのに、なぜ日本では普及しないのか不思議。有志による日本語化パッチのようなものもあるのかという話もあるのにな。

一般論としては、これで十分という感じ。さらに凝るならば、次のような項目があるでしょう。

対策 5 Ad-Awareなどによるスパイウェアの監視/除去。

対策 6 Registry Clean Expertなどによるレジストリの監視/管理。

対策 7 Essential NetToolsなどによるネットワークの監視/管理。

そして、セキュリティ関係のニュースに注意し、迅速に対応してゆくしかありません。メジャーなワームやウイルスの場合、それ専用のパッチや防止/除去ツールが即利用可能になりますから、楽といえば楽な話です。

例題1： 簡単なWindowsアプリ

本連載は、今回が12回目ゆえ、めでたく完了。「Visual Basic .NETで気にな

リスト1：例題1 (VB01.vb)

```
Imports System.Drawing

Public Class Form1
    Inherits System.Windows.Forms.Form

    ' フォームデザイナーで生成された部分は省略

    ' 描画ルーチン
    Protected Overrides Sub OnPaint(_
        ByVal e As System.Windows.Forms.PaintEventArgs)
        Dim g As Graphics = e.Graphics
        Dim i, j As Integer
```

```
        Dim x, y As Single
        For i = 1 To 15
            For j = 1 To 10
                x = 10 * i
                y = 10 * j
                g.DrawRectangle(Pens.Black, _
                    New Rectangle(x, y, 8.0, 8.0))
            Next
        Next
    End Sub

End Class
```

なること」を毎月取り上げてきましたが、最後は「印刷」に関して少々調べてみることにしましょう。要するに、プリンタへの出力です。ここで、アプリケーションは、本連載にしては珍しくWindowsアプリケーション。つまり、GUI付きのアプリケーションです。ただし、その内容たるや図1にあるように、単純極まりないもの。ソースコードを見ても、「OnPaint」を付け加えただけで、まだ印刷機能はありません（リスト1）。

さて「フォームへの描画は実際にはGraphicsオブジェクトへの操作である」とは、みなさんもよくご存知の通り。ここで、GraphicsオブジェクトはBorlandで言うところの“Canvas”であり、GDI+の世界です。要は「描画用各種機能付き画布」です。この例題では、このGraphicsオブジェクトに、とりあえず図1のような絵を描きました。

例題2：印刷機能

先の例題に印刷機能を付けます（リスト2）。図2のように、印刷メニューから印刷できるようにするわけです。そ

のためには、このMainMenuコンポーネントのほかに、PrintDialogコンポーネントとPrintDocumentコンポーネントが必要です。

ここでは、PrintDialogコンポーネントをPrintDialog1、PrintDocumentコンポーネントをPrintDocument1と名付けました。プロパティウィンドウでPrintDialog1のDocumentプロパティを「PrintDocument1」に設定するのを忘れ

図1：例題1のWindowsアプリケーション

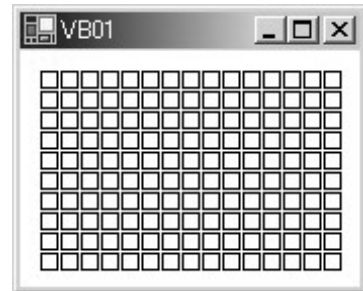


図2：例題2のWindowsアプリ

