

Windows プログラミング

第 14 回

Windows PowerShell & IronPython

こだか かおる
KODAKA, Kaoru

Monad Shellから PowerShellへ

今回は、Monad Shellについて紹介しました。Monad Shellという名称は開発時点のコードネームでしたが、ようやく正式名称が決まったようです。その名も「Windows PowerShell」。

<http://www.microsoft.com/windowsserver/2003/technologies/management/powershell/default.mspx>

そこで本稿でも今後は正式名称で統一したいと思います。名前は変わりましたが、含まれる機能は変わりありません。前回の記事では、今までのコマンドプロンプト、また、UNIXのシェル

機能に比べても、Windows PowerShellがかなり高機能であることが理解してもらえたと思います。その登場が非常に楽しみなテクノロジーのひとつといえるでしょう。Windows PowerShellは、Exchange Server 2007とSystem Center Operations Manager 2007に搭載されるということですが、単体での配布や、Windows/Microsoft Updateでの提供も強く期待したいと思います。

さて、人間というのは不思議なもので、その時点では納得していても、のちのち不満が出てきたりするものです。前回、Windows PowerShellの記事を書いたのはみたものの、読み返してみると少々言葉が足りなかったと後悔してし

まいました。ということで、今回はまず前回書き足りなかったWindows PowerShellの補足説明をしたいと思えます。また、それだけでは何なので、Windows PowerShellとの関係が目されるIronPythonについても取り上げることにします。

もう一度、ダウンロードと インストール

Windows PowerShellという正式名称の決定に伴い、RC1のダウンロードも始まりました(図1)。RC1はベータのときと同様に、Microsoftのダウンロードセンターからダウンロードできます。ドキュメンテーションパック(Windows PowerShell RC1 Documentation Pack)も提供されていますのでお忘れなく。

以前のベータバージョンがインストールされていると、インストール時に怒られることがあるので、あらかじめアンインストールしておきましょう。また、インストールが終了したら、リリースノートも欠かさずに確認しましょう。機

レベル >>> Level

1 2 3 4 5

言語 >>> Language

- C#
- PowerShell
- Python

ツール >>> Tool

- Visual Studio 2005 Professional
- Windows PowerShell RC1
- Windows PowerShell RC1 Documentation Pack
- IronPython

*) 本記事では英語環境(Windows XP英語版)で検証しています。そのため、掲載画面の一部が日本語版OSの表示と異なります。

能的な変更はさほど多くないものの、名称が変わったことによる違いがけっこうあります。この変更は「MSH」→「PS」という形で行なわれ、スクリプトファイルの拡張子なども変わっている（「.msh」が「.ps1」に）ため注意が必要です。

なお、前回と同じように、初回実行時に認証するかどうかの確認が求められます。ここで拒否してしまうとWindows PowerShellを動かすことができないので、忘れずに許可しておきましょう。

スクリプトファイルの関数を実行する

前回の記事では、まとまった処理をひとつのファイルに保存してから実行するという方法を紹介しました。一般的な手法として、ある程度大きな処理では、このようにスクリプトファイルの形で利用することが多いでしょう。ひとつの処理がひとつの機能で完結するのであれば、それでもよいのですが、「再利用」ということに思いをめぐらせるのが開発者というものです。

つまり、一度作った処理は二度と作りたくありませんし、コピー&ペーストするのも面倒です。特定の機能を別のファイルに切り出しておいて、コンポーネントのように扱えると便利でしょう。Windows PowerShellでは、.NET Frameworkのコンポーネントを使うこともできますし、それ以外にもいくつかの方法が用意されています。まずは、もっとも簡単な方法から見ていくことにします。

図1：Download Center - Windows PowerShell RC1

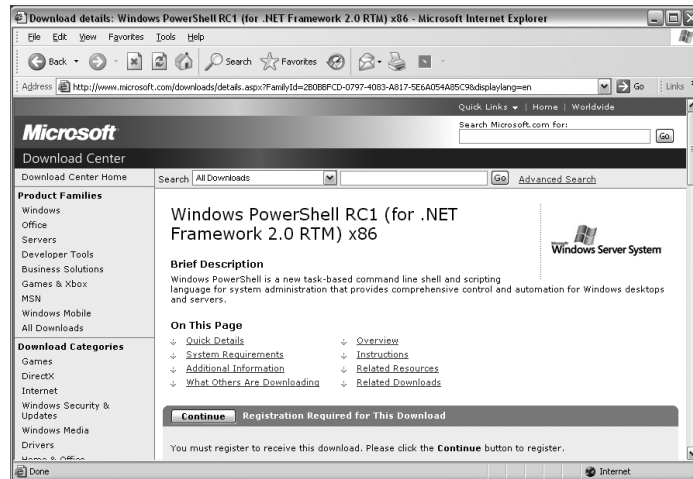


図2：Set-ExecutionPolicy のヘルプ

```

Windows PowerShell
PS C:\sample> get-help set-executionpolicy

NAME
    Set-ExecutionPolicy

SYNOPSIS
    Set the execution policy for the current shell.

DETAILED DESCRIPTION
    The cmdlet writes the execution policy into the local preference setting in the registry. It will warn you if the local setting is overridden by group policy. The registry setting remains updated regardless. NOTE: This cmdlet can't be used to modify group policy setting.

    Restricted
    - Default execution policy.
    - Does not run scripts.
    - Interactive only.

    AllSigned
    - Runs scripts.
    - All must be signed by a publisher that you trust.

    RemoteSigned
  
```

最初に紹介するのは、特定の機能を関数として切り出し、それを再利用する方法です。これは多くのプログラミング言語でソースライブラリとして提供される方法で、中に含まれているスクリプトコードを利用者が見ることができます。「作成者＝利用者」という環境であれば、この方法で何の問題もありません。必要なときに読み込み、その関数を実行するだけです。

ファイルを実行するときは、前回も紹介したようにExecution Policyを変

更しなくてはなりません。ベータ3では、レジストリを編集する必要がありましたが、Windows PowerShell RC1では、専用のCmdletが用意されています。

Get-ExecutionPolicy
Set-ExecutionPolicy

がそれです。これらのCmdletを利用することで、現在の設定を確認したり、新しく設定を変更することができます(図2)。