Otey

"PowerShell 2.0 adds several important features to the PowerShell language and development experience that make it easier to use."

New Features in Windows PowerShell 2.0

Remoting, script debugging, and other features fill PowerShell gaps

If you've wanted to take advantage of PowerShell scripting but found the learning curve a bit too steep, PowerShell 2.0 might be just the ticket. Expected to be released in the second half of 2009, PowerShell 2.0 adds several important features to the PowerShell language and development experience that make it easier to use and fill in some of the gaps that were present in PowerShell 1.0. Here are my top ten favorite new features.

10 **New cmdlets**—PowerShell 2.0 includes 24 new cmdlets. Some of the cmdlets work with debugging, Windows Management Instrumentation (WMI), and background jobs. Out-GridView is one of my favorites; it displays the results of other commands in an interactive table so that you can sort and search the data.

9 **New operators**—PowerShell 2.0 provides three useful new operators. The @ operator (pronounced splat) passes a collection of parameters; -split breaks a string into an array; and -join concatenates multiple strings, adding separators.

8 **New built-in variables**—PowerShell 2.0 includes four new built-in variables. The $commandLineParameters variable accesses command line parameters. $PSVersionTable reports the current PowerShell version. $Culture and $UICulture report the current culture (i.e., the language setting) and UI culture information.

7 **Try-Catch-Finally**—Following in the footsteps of the other .NET languages, PowerShell 2.0 adds the standard Try-Catch-Finally structure to the language. You use the Try block to safely execute one or more statements. If an error occurs, the code in the Catch block will be executed. An optional Finally block contains code that is run after the Try-Catch portion completes.

6 **PowerShell Hosting APIs**—Microsoft has included PowerShell scripting support in all its recent server products, including Exchange Server 2007, SQL Server 2008, and Windows Server 2008, as part of its Common Engineering Criteria. The new PowerShell Hosting APIs promise to extend PowerShell functionality to other products because they simplify hosting PowerShell in applications.

5 **Script debugging**—Another important improvement in PowerShell 2.0 is its enhanced debugging capabilities. PowerShell 2.0 has a cmdlet-based debugger that lets you set breakpoints and step through your scripts using the PowerShell console window without needing any graphical development tools. To find out more about PowerShell 2.0's script debugging, run the command

```
get-help about_debugger
```

4 **Background jobs**—PowerShell 1.0 doesn't have the ability to run a background process, which makes it tough to replace Windows Shell scripts that make use of the Start command. PowerShell 2.0's new Start-PSJob cmdlet asynchronously runs background jobs on local or remote systems. For more information, you can run

```
get-help about_psjob
```

3 **ScriptCmdlets**—In PowerShell 1.0, you need to program in .NET to create new cmdlets. This requirement means that typically only developers can make new cmdlets. Powershell 2.0 lets administrators create ScriptCmdlets using PowerShell itself. For more information about creating ScriptCmdlets, run

```
get-help about_scriptcmdletparameters
get-help about_scriptcmdletmethods
```

2 **Remoting**—One of the most important changes in PowerShell 2.0 is support for running scripts on remote systems. PowerShell Remoting lets you run scripts on remote networked systems. This new remoting support requires that PowerShell 2.0 be installed on both the local and remote systems. For more information, run

```
get-help about_remoting
```

1 **Integrated Scripting Environment**—My favorite PowerShell 2.0 feature is the new Integrated Scripting Environment. ISE is a multitabbed graphical PowerShell development platform that features color-coded syntax. It also includes debugging capabilities that let you set breakpoints and step through your PowerShell scripts. If you've had trouble getting started with PowerShell, ISE will definitely kick start your PowerShell scripting.

InstantDoc ID 101734

MICHAEL OTEY (motey@windowsitpro.com) is technical director for *Windows IT Pro* and *SQL Server Magazine* and author of *Microsoft SQL Server 2008 New Features* (Osborne/McGraw-Hill).