

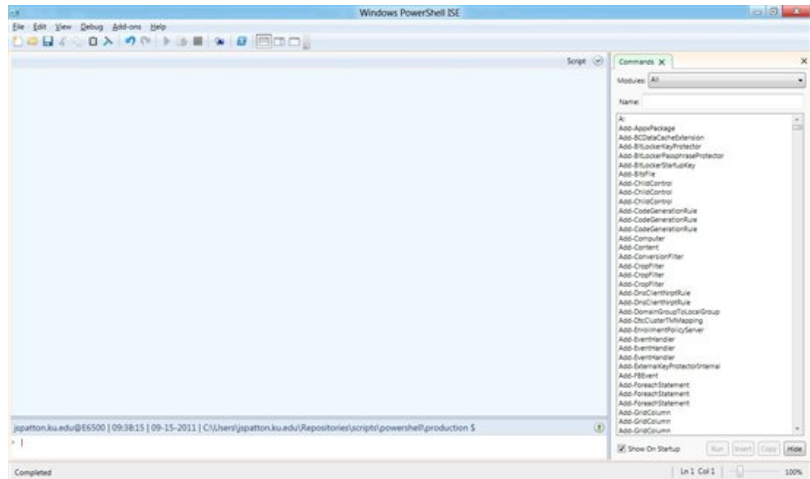
Windows PowerShell ISE v3.0 - New features

I'm sure by now everyone has had a chance to play with the Windows 8 Developer Preview, I know I'm really excited about it, but moreover I'm very excited about the PowerShell ISE. I have moved away from using Notepad and Notepad++ for developing in PowerShell, to PowerShell ISE exclusively. With this new version of PowerShell ISE it's much more of an IDE than it was before, so I wanted to show a few things off.

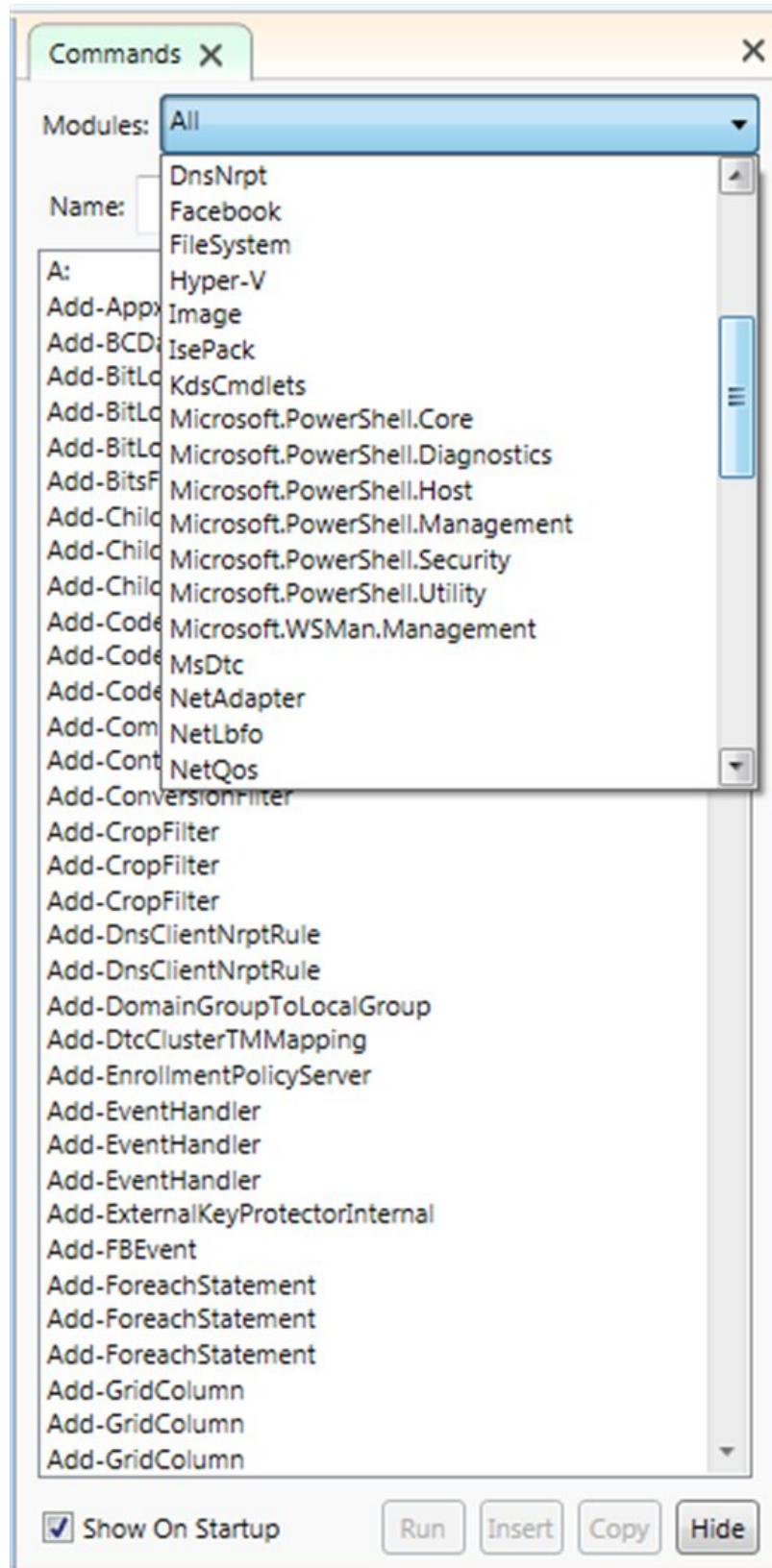
\$Host | Select-Object -Property Name, Version | ConvertTo-Html

Name	Version
Windows PowerShell ISE Host	3.0

Here you can see the whole application, and you will notice the command bar off to the right. This command bar has some nice features that will let you access commands from the various modules that are available on your machine very easily.

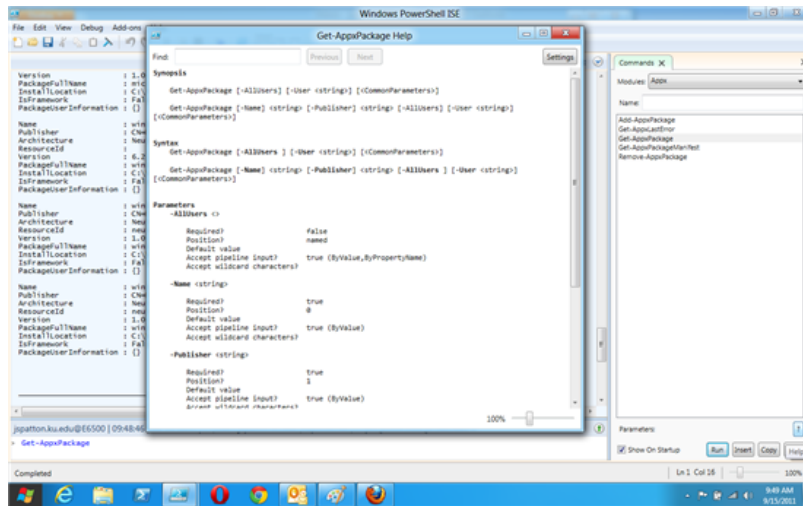


Here is a close-up of the command bar showing some of the modules that are available by default.

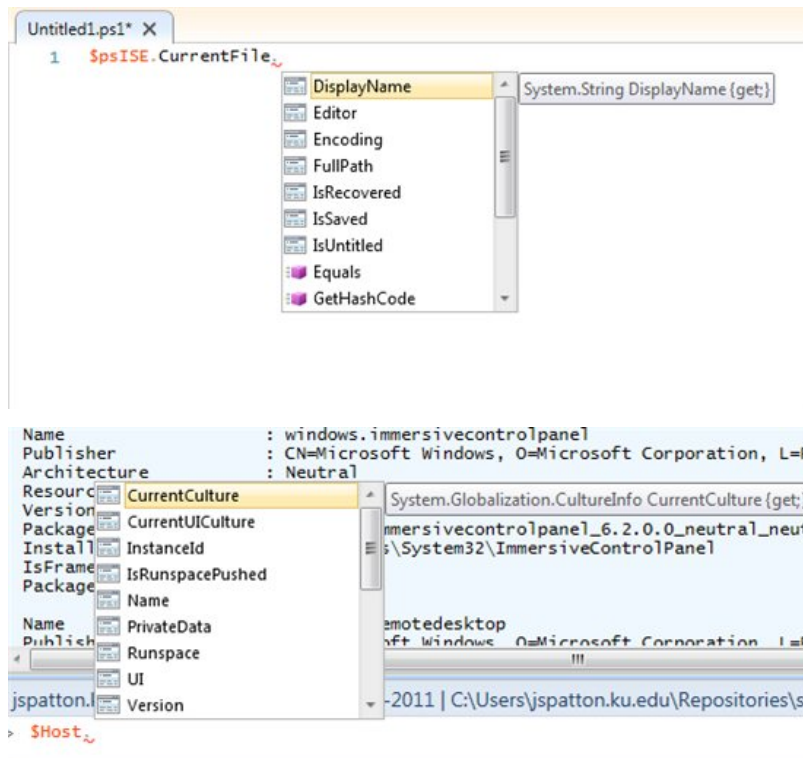


There are some buttons across the bottom Run, Insert, Copy and Hide. When you select a command from the list above and click Run the command is run and the output is displayed in the output pane. If you have a command selected and click Insert, that command is inserted into either the script pane or the command pane. Finally a selected command can be copied to the clipboard, or you can hide this bar altogether.

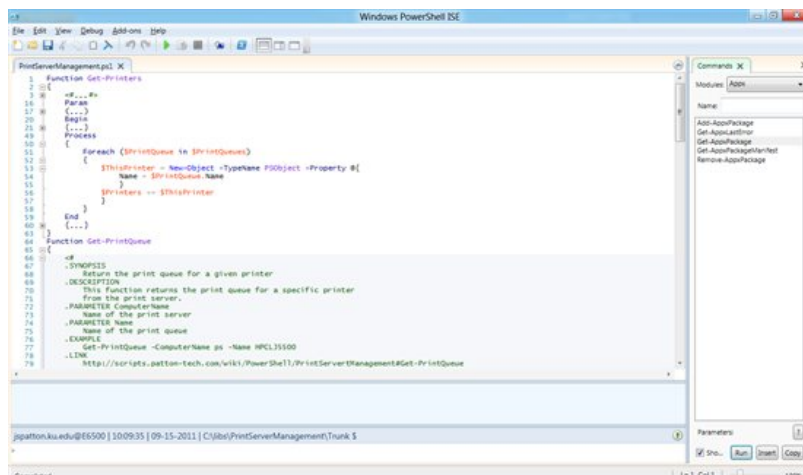
Once a command is selected an additional button appears [?] with the command selected clicking this displays the help in a separate window that is not modal, but sits on top of everything. The screenshot below shows you the command bar with all the buttons clicked.



Auto-complete for many things in PowerShell have been around for a while, but if you miss the Intellisense features from like Visual Studio, it's now available within ISE, both in the script pane and the command pane. I know this can be really helpful as I'm used to using get-member to get a lot of this information as I'm working on a script.



Outlines! I really missed this feature when I moved from Notepad++ to PowerShell ISE. But in the following screenshot you can see that you now have collapsible outlines in PowerShell ISE, as well as something like code highlighting. You can see I have one of my functions collapsed, and have left the Process section of the code expanded. In the function below you can see that the help section of the function is highlighted. You can't see my mouse from this screenshot but it's hanging out in the border near line 66 where the help section begins.



Now that Windows Server 8 Beta and Windows 8 Consumer Preview are available, HELP for ISE tells us the following:

Many new features and improvements have been added to Windows PowerShell ISE for Windows Server 8 Beta.

Intellisense, an auto-completion assistance feature that is similar to that found in Visual Studio®, is now part of Windows PowerShell ISE. Intellisense displays clickable menus of matching cmdlets, parameters, parameter values, files, or folders as you type.

Windows PowerShell ISE now supports add-on tools, which are Windows Presentation Foundation (WPF) controls that are added by using the object model. Add-on tools can be displayed in the console by using either a vertical or a horizontal pane. Multiple add-on tools in a pane are displayed as a tabbed control. A maximum of 20 user-defined add-on tools are allowed. You can also add or remove add-on tools that are produced by third parties.

PowerShell ISE now automatically saves your open scripts every two minutes. To change the automatic saving interval, run the following in the console pane:

`$psise.Options.AutoSaveMinuteInterval`. If Windows PowerShell ISE stops working, or if the operating system is restarted, after Windows PowerShell ISE is restarted, it recovers scripts that were open in the last session, even if the scripts were not saved.

The separate command and output panes that were available in the first release of Windows PowerShell ISE have been combined into a single console pane. The console pane is similar in function and appearance to a typical Windows PowerShell console, but includes the following enhancements,

- Syntax coloring for input text (not output text)
- Intellisense
- Brace matching
- Error indication
- Full Unicode support
- F1 context-sensitive help
- Ctrl+F1 context-sensitive Show-Command
- Complex script and right-to-left support
- Font support
- Zoom
- Line- and block-select modes
- Preservation of typed content at the command line when you press the Up arrow to view history in the console
- Command-line switches If you start Windows PowerShell ISE from the command line (`Powerhshell_ise.exe`) you can now add the following new command-line switches.
- `-NoProfile`: Starts Windows PowerShell ISE without running `$profile`
- `-Help`: Displays a help window
- `-mta`: Starts Windows PowerShell ISE in multi-threaded apartment mode. (The default is STA)

Other Windows PowerShell ISE editing features adapted from Visual Studio include:

- Brace matching Windows PowerShell ISE now includes brace matching and highlighting (for example, using the Go to Match command locates the closing brace, if you have an opening brace selected).
- Outline view The script pane now supports outlining, which allows collapsing or expanding sections of code by clicking plus or minus signs in the left margin. You can use either braces or `#region` and `#endregion` tags to mark the beginning or end of a collapsible section.
- Parse error display Parse errors are now indicated by using red underlines. When you hover over an indicated error, Tooltip text displays the problem that was found in the code.
- Zoom The zoom percentage of the console's content can be set by using the zoom slider (in the lower right corner of the Windows PowerShell ISE window), or by entering the command `$psise.options.Zoom` in the console pane.
- Rich text copy and paste Now, copying to the clipboard in Windows PowerShell ISE preserves the font, size, and color information of the original selection.
- Block selection You can select a block of text by holding down the ALT key while selecting text in the script pane with your mouse, or by pressing `Alt+Shift+Arrow`.
- New Help viewer window If you press F1 when your caret is in a cmdlet, or you have part of a cmdlet highlighted, the new Help viewer opens context-sensitive Help about the highlighted cmdlet. To display Windows PowerShell About help, type operators in the console pane, and then press F1. Before you use this feature, download the most current version of Windows PowerShell Help topics from the Microsoft website. The simplest method of doing this is to run the `Update-Help` cmdlet in the console pane.
- Show-Command Show-Command allows you to compose or run a cmdlet or function by filling out a GUI-based form. The form lets users work with Windows PowerShell in the graphical environment with which they are comfortable. Show-Command also enables advanced scripters to create a quick Windows PowerShell-based GUI

See Also

- [PowerShell Portal](#)
- [Wiki: Portal of TechNet Wiki Portals](#)