# Windows PowerShell

## Introduction to PowerShell

### What is PowerShell?

* Task Automation Framework
* Command-Line Shell **and** Scripting Language for Windows
* Based on the .NET Framework
* Inspired by UNIX shells and scripting languages such as Perl and Python



### Terminology: Shell- vs. Scripting-Languages

#### (Command-Line) Shell

“A shell is a piece of software that lets you access the functionality provided by the operating system.”  
− Bruce Pyette: PowerShell in Action

Common Features:

* Read-Evaluate-Print loop (REP)
* Aliases/Shortcuts for long command names
* Wildcard matching, so you don’t have to type the full names of everything
* Call external programs
* Command History

UNIX Shells

* Bourne-Again Shell (bash)
* Korn Shell (ksh)

Windows Shells

* Command.com
* Cmd.exe



* PowerShell
* Cygwin



Others

* Chrome Console



* Firebug Command Line



#### Scripting Languages

* Scripting languages typically provide more sophisticated features for debugging your scripts and they provide mechanisms for developing larger scripts by letting you break a script into components or modules
* Scripting Language syntax is oriented more towards writing an application than toward interactively issuing commands

### Why PowerShell?

* “When comparing the command-line manageability of a Windows system to a UNIX system, Windows was found to be limited”
* Windows command line is an inferior technology which cannot compete with the known UNIX Shells
* Windows is known for its Management UIs but lacked an automation framework 🡪 “Point and click does not scale”
  + Not across different people
  + Not over time
  + Not across different server environments
  + Not across multiple servers
  + And it is hard to control and to document

### Version History

* Designed from scratch in 2005
* Version 1.0 was released in 2006 for Windows XP, Windows Server 2003 and Windows Vista. Only optional for Windows Server 2008.
* Version 2.0 is the current version and was released in 2009 for Windows 7, Winder Server 2008 R2

### The Future of PowerShell

* PowerShell Version 3.0 is currently a CTP and will be released with Windows Server 8
* The management of Windows Server 8 (Core) will be entirely based on PowerShell
* For the release of Windows Server 8 the number of built-in PowerShell commands (cmdlets) will be increased from 200 to 2300

Links

* Windows Server 8 Developer Preview and PowerShell v3 First Look - <http://www.mikepfeiffer.net/2011/09/windows-server-8-developer-preview-and-powershell-v3-first-look/>
* Microsoft verabschiedet sich vom GUI - <http://www.golem.de/1109/86435.html>
* Powershell dominiert die Server-Verwaltung ab Windows Server 8 - <http://www.nt4admins.de/themen/verwaltungs-tools/artikel/powershell-dominiert-die-server-verwaltung-ab-windows-server-8.html>

### What is special about Windows PowerShell?

* PowerShell is a new class of object-based shell language ­− most other shell-languages are string-based
  + PowerShell preserves the structure of the Windows data types by using the .NET object model 🡪 No tedious string parsing required
* Command-Line Shell AND Scripting Language in one product – PowerShells goal is to be both a good scripting language and a good interactive shell
* PowerShell leverages the full functionality of the .NET framework
  + You can use all the objects and functions that you know and love from the .NET Framework

## Getting Started with PowerShell

### Online Resources about Windows PowerShell

* A list of PowerShell Cmdlets − <http://technet.microsoft.com/en-us/library/hh848794.aspx>
* Scott Hanselman’s PowerShell Posts − <http://www.hanselman.com/blog/archives.aspx#PowerShell>
* PowerGUI: An alternative to the PowerShell ISE − [http://powergui.org](http://powergui.org/)
* Web Administration Cmdlets for Windows PowerShell − <http://technet.microsoft.com/en-us/library/ee790599(WS.10).aspx>
* Microsoft Technet Script Resources − <http://gallery.technet.microsoft.com/scriptcenter/>
* Official PowerShell Homepage − <http://technet.microsoft.com/en-us/scriptcenter/dd742419.aspx>
* My Personal PowerShell Bookmark Collection − <http://delicious.com/andyk7/powershell>

### Installation & Startup

#### PowerShell Command-Line

If you are running a recent version of Windows you will have PowerShell already installed:

Start > Accessories > Windows PowerShell



#### Windows PowerShell Integrated Scripting Environment (ISE)

If don’t have the PowerShell debugger installed you can add it via the Windows Features dialog:

Server Management > Features > Add Feature > “Windows Integrated Scripting Environment (ISE)”





#### Installation Directory

The PowerShell executables are located in the folder:

%windir%\System32\WindowsPowerShell\v1.0



### Configuring Mercurial

Start PowerShell

> notepad $Home\Mercurial.ini

Enter the following text and then save the changes:

[ui]  
username = Your Name <Your.Name@Domain.Com>  
verbose = True

[Extensions]

purge =



### Fetching the workshop material from my Mercurial repository

In order to get a copy of the PowerShell workshop material please clone my mercurial repository and create a branch that is named after you.

Start PowerShell

> mkdir $Home\Desktop\dev | Set-Location  
> hg clone <http://andyk.dyndns-server.com:8000> PowerShell-Workshop  
> Set-Location PowerShell-Workshop  
> hg branch “YourName”  
> hg push --new-branch

## Core Elements of PowerShell

## Lessons

### Extending PowerShell with Clipboard Access (Lesson: Profile-Extension-Clipboard)

Links

* The Windows PowerShell Profile − <http://technet.microsoft.com/en-us/library/ee692764.aspx>
* Copy and Paste with Clipboard from PowerShell − <http://brianreiter.org/2010/09/03/copy-and-paste-with-clipboard-from-powershell/>