
	<p>1A06</p> <p>Windows Powershell Rc1 (aka Monad)</p> <p>Decus 2006</p> <p>Juergen.Krautner@hp.com HP-Services</p>
---	---

<p>Agenda </p>	
<ul style="list-style-type: none">- Background- Getting started (install and first steps)- Security- Concepts<ul style="list-style-type: none">• cmdlets, providers, drives ...- Development<ul style="list-style-type: none">• debugging, tracing ... - Demos / Samples- Quickref	
<p>Mai 2006</p>	<p>Microsoft Powershell 2</p>

Microsoft Powershell



- **Interactive Shell & Scripting language**
- Task-based & Command-line-oriented
 - comprehensive control and automation of Windows
- Unix like
- **Objectbased Pipelining**
- Man-page-style help with rich schema and searching
- **Customizable environment**
 - Namespaces and drives, startup files, aliases, preferences
- **Text and Object Utilities**
- Progress Reporting
- **.Net integration**
- Com (ActiveX) Support
- Text, COM, WMI, ADSI, ADO, XML, Commands
- Extensible
 - snapins, functions, cmdlets etc ..

Mai 2006

Microsoft Powershell

3

Installation / Getting Started



- Requires XP or W2003 (no W2000)
- Requires .NET 2.0 Runtime
- Straight forward Install of PowerShell_Setup-xxx.msi
 - x86 , x64 and IA64-Versions available
- Invocation
 - `c:\> Powershell (-help, -nologo, -noninteractive, noprofile ...)`

```
Windows (R) PowerShell
Copyright (C) 2005 Microsoft Corporation. All rights reserved
```
- Profiles
 - All Users: Documents and Settings\All Users\Documents\PSConfiguration\profile.ps1
 - Individual: \$HOME\My Documents\PSConfiguration\profile.ps1

Mai 2006

Microsoft Powershell

4

Getting Started (running a script)



```
D:\monad\samples> powershell
Windows(R) PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.
Loading System Profile ...
Loading Juergens Profile ...
Datum: 03.05.2006 13:55:54
PS D:\monad\samples> get-disksize.ps1
'get-disksize' is not recognized as a cmdlet, function, operable program, or script file.

PS D:\monad\samples> .\get-disksize
The file D:\monad\samples\Get-DiskSize.ps1 cannot be loaded. The file D:\monad\samples\Ge
t-DiskSize.ps1 is not digitally signed. The script will not execute on the system. Please
see "get-help about_signing" for more details..

PS D:\monad\samples> Set-ExecutionPolicy unrestricted
PS D:\monad\samples> .\get-disksize

Security Warning
While scripts from the Internet can be useful, this script can potentially harm your
computer. Only run scripts that you trust. Do you want to run
D:\monad\samples\Get-DiskSize.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R

Device ID      Type      Size (m)      Free Space(m)
-----
C:             HDD       17.002,59     3.918,84
D:             HDD       40.221,56     9.837,59
F:            CD         0,00         0,00
R:            CD         34,91         0,00
W:             4         3.396,09     2.332,99
```

Mai 2006

Microsoft Powershell

5

Security



- Execution policy:

- Restricted : Interactive shell only, no scripts, only signed config files
- Allsigned: requires digital signatures (verisign or selfsigned(makecert))
- Remotesigned local created scripts run. foreign scripts not
- Unrestricted local created scripts run. foreign scripts displays prompt

- Admins Friends


- **-Whatif**
 - del *[1-2]* -whatif
 - gps | where { \$_.handles -ge 500 } | stop-process -whatif
 - Get-mailbox b* | move-mailbox -targetdatabase VIP -whatif
 - **-Confirm**
 - stop-process -proc s* -confirm
- [Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help-Verbose

Mai 2006

Microsoft Powershell

6


Concepts



- Providers
 - present data from different data stores in a consistent manner to Cmdlets
 - default providers include: Alias, Certificate, Environment, Filesystem, Function, Registry, and Variable.
 - get-PSprovider
- Drives
 - a namespace created by mapping providers
 - contains numerous information axis
Item, ChildItem, Content, Property, ACL, etc.
 - provide a new navigation experience by browsing different data the same way

Mai 2006 Microsoft Powershell 7

Concepts



- Cmdlets
 - small executables (written in any .net language)
 - verb-noun syntax (get-help, clear-host)
 - can be aliased
- Pipelines
 - passes managed objects !
 - Cmdlet can use .NET reflection to look inside the objects being passed and see that the length attribute is an integer, and the name attribute is a string.
- demo

Mai 2006 Microsoft Powershell 8

Concepts



- Variables
 - any .Net Datatype (scalar, arrays, objects, **associative arrays**)
 - \$procs = get-process | where {\$_.handles -gt 500} | sort {\$_.handles}
 - (dir)[0].name
- Aliasing (mapping)
 - Set-Alias gps Get-Process
 - get-alias
- Positional Parameters
 - gps -ProcessName winlogon
 - gps winlogon
- wildcarding, parameter abbreviation
 - gps win*
 - gps -Proc winlogon
- formatting
 - format-list, format-wide, format-table, format-custom
- Tab-Completion in all Drives
 - Dir HKLM:\So<TAB>\Mi<TAB> => HLKM:\Software\Microsoft

Mai 2006

Microsoft Powershell

9

Help, I'm lost ...



- Man-page style help on language and commands
 - > Get-Help
 - > Get-Help <command> or <command> -?
 - > Get-Help * |Where {\$_.Synopsis -match "process"}
 - > Get-Help About_XXX (xxx = if,while,for... & others)
- Finding Commands
 - > Get-Command i* (exe,dll,cmdlets etc)
 - > Get-Command -Verb get
 - > Get-Command -Noun object
 - > Get-Command -Type {Alias | Function | Filter | Cmdlet | ExternalScript | Application | Script | All }

Mai 2006

Microsoft Powershell

10

Help on .Net



• Reflection

- Selfdocumenting Objects (Metadata that describes an object)
- Sample:

```
$dir=dir
```

```
$dir | get-member (-MemberType Property)
```

Name	MemberType	Definition
AppendText	Method	System.IO.StreamWriter
CopyTo	Method	System.IO.FileInfo CopyTo(String ..
Create	Method	System.IO.FileStream Create()
..		
CreationTime	Property	System.DateTime CreationTime {get;set;}
Exists	Property	System.Boolean Exists {get;}
Extension	Property	System.String Extension {get;}
FullName	Property	System.String FullName {get;}

```
([datetime] "10/1/2005") | gm
([datetime] "12/25/2006").AddDays(-35)
[dateTime]::now
```

Mai 2006

Microsoft Powershell

11

.Net example



```
C:\Program Files\MSH\msh.exe
MSH C:\> $news = [xml] (new-object Net.WebClient).DownloadString("http://news.google.de/news?ned=de&topic=m&output=rss")
MSH C:\> $news.rss.channel.item | format-table title,link

title                                     link
-----                                     -
Lichtblick für Geflügelhalter Stal...  http://news.google.de/news/url?sa=I&...
Aktion gegen Rezeptbetrüger Razzia ... http://news.google.de/news/url?sa=I&...
Schuld ist der lange Winter Frühblä... http://news.google.de/news/url?sa=I&...
Vierjährigen sollen Marathonläufe ... http://news.google.de/news/url?sa=I&...
Gezielte Aufklärung gegen zunehmend... http://news.google.de/news/url?sa=I&...
Hundt fordert fünf Euro pro Arzthes... http://news.google.de/news/url?sa=I&...
Harmloser Stich mit schweren Folgen ... http://news.google.de/news/url?sa=I&...
Kennedy kandidiert trotz Sucht für ... http://news.google.de/news/url?sa=I&...
Wie finde ich den richtigen Arzt? - ... http://news.google.de/news/url?sa=I&...
Weiterer Vogelgrippe-Impfstoff beste... http://news.google.de/news/url?sa=I&...
Nationalrat lehnt Einheitskrankenkas... http://news.google.de/news/url?sa=I&...
Weitere Gentests bei künstlicher Be... http://news.google.de/news/url?sa=I&...
Bluthochdruck schadet vor allem den ... http://news.google.de/news/url?sa=I&...
Vorsicht Zecken - Rundfunk Berlin-Br... http://news.google.de/news/url?sa=I&...
AOK warnt vor stärkerem Ärztemange... http://news.google.de/news/url?sa=I&...
Ärztestreik aktuell: - GesundheitPro... http://news.google.de/news/url?sa=I&...
Holzkreuze in den Schaufenstern - Ma... http://news.google.de/news/url?sa=I&...
Tarifverhandlungen für Uni-Ärzte v... http://news.google.de/news/url?sa=I&...
Neues Verfahren zur Zellreparatur - ... http://news.google.de/news/url?sa=I&...
USA: Spitalbehandlungen in XXL - 20n... http://news.google.de/news/url?sa=I&...

MSH C:\>
```

Mai 2006

Microsoft Powershell

12

Object (Shell) Utilities



- **Foreach {}**
 - gps | foreach {"\$((\$_.Name): \$((\$_.Handles))"
 - \$aa=@{"key1"=1;"key2"="juergen"}
 - foreach (\$k in \$h.Keys) {"{0} => {1}" -f \$k,\$h.\$k}
- **Group**
 - > gps | group Company
- **Measure**
 - > gps | Measure Handles -Sum -Ave -Min -Max
- **Select**
 - > gps | Select name,id -expand modules
- **Sort**
 - > gps | Sort handles |Select -first 10
- **Where**
 - > gps | where {(\$_.handles -ge 500)}

Language: Control Structures



- **If / else**
 - if (\$i % 2) {"Odd"} else {"Even"}
- **while statement**
 - while (\$i -lt 10) { \$i++; "I is \$i" }
- **do/while statement**
 - do { "I is \$i"; \$i-- } while (\$i -ge 0)
- **foreach statement**
 - foreach (\$i in get-childitem) { ... }
- **for statement**
 - for (\$i=0; \$i -lt 10; \$i++) { [math]::sqrt(\$i) }
- **switch statement !!!**

Switch statement



- The switch statement is a powerful pattern/action statement
- It can do simple switching like C# or complex processing like AWK

```
'switch' ['-regex' | '-wildcard' | '-exact' ] ['-casesensitive']
[-file' <propertyOrArrayReferenceRule> | '(' <pipeline> ')']
' { [ [
    'default' |
    <pattern> |
    <expression> |
    <statementBlockRule>
] <statementBlock> ]+
},
```

```
# Count all the .dll .pdb and other files in the current
# directory and all subdirectories
switch -wildcard (get-childitem
c:\windows\system32)
{
    "*.exe" {$exes++}
    "*.com" {$coms++}
    default {$others++}
}
"exes $exes coms $coms others $others"
```

Mai 2006

Microsoft Powershell

15

Functions



- Simple:

Arguments not declared or typed

```
function MyEcho {
    for ($i = 0; $i -lt $args.count; $i++)
        { "arg{0} = {1}" -f $i, $args[$i] }
}
```

```
function fh {
    $item = $args[0]
    "searching for content: $item"
    Get-Help * | Where {$_.Synopsis -match "$item" }
}
```

- Sophisticated:

Parameters can be: Named, Typed, Initialized

Processing can occ

Before any pipeline objects are

For each object in the pipeline

After all pipeline objects are processed

```
function Get-Total (
    [string]$property = $(throw "Property Required"),
    [string]$formatString = "Total {1} = {0}") {
    Begin { $total = 0 }
    Process { $total += $_.$property }
    End { $formatString -f $total, $property }
}
```

Beispiel: gps | Get-Total -Property Handles

Mai 2006

Microsoft Powershell

16

Language: Operators



- All the classic arithmetic, comparison, bitwise and logical operators
 - **plus**
 - -like / -notlike wildcard comparison
 - -match / -notmatch reg. Exp. comparison (\$matches !)
 - -is / -isnot type comparison
 - in 1...7 range operator
 - -replace / -ireplace replace operator
 - -contains
- (Case insensitive by default)

Examples:

```
$x -contains 4
"1c2cac9c" -replace "[1-7]c","Wc"
"abcdefg" -like "a*c*g"
"abcdefg" -match "^a.*c.*g$"
-
```

Mai 2006

Microsoft Powershell

17

Customizing



- Series of profile files from run at startup
 - C:\Documents and Settings\All Users\Documents\PSConfigurationProfile.ps1
 - <My Docs>\PSConfiguration\Profile.ps1
- XML based Config File
 - for loading Snapins
- Aliasing
 - Remap command names (only cmdlets, no parameters)
- Prompt
 - Function Prompt {


```
#Anything that returns a string
"$($pwd) >"
}
```
- Many preferences controllable via variables
 - > dir Variable:*preference*
 - > dir Variable:Maximum*
 - > dir Variable:Report*

Mai 2006

Microsoft Powershell

18

Debugging



- **Write-Debug (cmdlet)**
 - \$DebugPreference
 - Continue – Write message
 - Inquire – Write message and ask to continue
 - Stop – Write message and terminate
- **Set-PSDebug**
 - -trace
 - 0=NONE, 1=Statements, 2 = statements and assignments, traps, script calls
 - -Step
 - Single steps script execution
 - -Strict
 - Checks if variables has been set
 - -Off
- **write-progress**

```
for($i = 1; $i -lt 1000; $i++) {write-progress Updating progress->
  -perc ($i/10) -second (100-$i/10) }
```

Ubiquitous Parameters



- **-Debug**
 - Programmer-level information
- **-ErrorAction {SilentlyContinue | Stop | Continue | Inquire}**
 - What to do if an error occurs
- **-ErrorVariable <VariableName>**
 - Assign errors to a variable
- **-OutputBuffer <Number>**
 - Buffer objects in the pipeline
- **-OutputVariable <VariableName>**
 - Output assigned to a variable
- **-Verbose**
 - Additional information about the activities being performed

Tracing / Timing

- **Get-TraceSource**
 - Provides the list of what is available to trace
- **Set-TraceSource** (permanent)
- **Trace-Command**
 - Allows for inspection into what is going on under the covers
 - Trace-command parameterBinding,cmdlet -Option ALL {get-process c*} -PShost
- **measure-object**
 - Counters on objects
- **measure-command**
 - Timing of a command/scriptblock (ticks,msec, etc)
 - Measure-command { start-sleep 5 }

...

TotalSeconds : 5,0028816
TotalMilliseconds : 5002,8816


Mai 2006 Microsoft Powershell 21

Future Prospect

- Powershell 2.0
 - Programming in the large features
 - GUIs layered over PowerShell
 - Remoting
 - Functional coverage
 - AD Provider
- Next Generation of MMC-Tools is build on Powershell
- premier mechanism for scripting
- Exchange 2007 Mgmt Ops & MOM 2007 based on Powershell

Mai 2006 Microsoft Powershell 22


Sources



- Download & Documentation
<http://www.microsoft.com/technet/scriptcenter/hubs/msh.msp>
- Blogs: <http://blogs.msdn.com/powershell/>
- Pointer to Newsgroup, Team Blog etc...
<http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.msp>
- Wiki:
<http://channel9.msdn.com/wiki/default.aspx/Channel9.MSHWiki>
- Monad Script Center: <http://www.reskit.net/monad/>
- .Net Classes Libraries
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/cpref_start.asp

Mai 2006 Microsoft Powershell 23

Session Summary



- Powershell
 - **Interactive** shell
 - Command **Scripting**
 - Com scripting
 - **.NET** power from the commandline
- Its for free !!!
- Supports very simple to advanced scripting
- Have fun with it !

Mai 2006 Microsoft Powershell 24

Quick Ref

- **Arrays** (... are zero-based.)
 - "a","b","c" array of strings
 - 1,2,3 array of integers
 - @() empty array
 - @(2) or ,2 array of 1 element
 - 1,(2,3),4 array within array
 - \$a[5] sixth element of array*
 - \$a[2][3] fourth element of the third element of an array
- **Assignment Operators**
 - = Assigns a value to a variable
 - += Adds a value to a variable
 - = Subtracts a value from a variable
 - *= Multiplies a variable by a value
 - /= Divides a variable by a value
 - %= Performs a modulus on a variable


Mai 2006 Microsoft Powershell 27

Quick Ref

- **Associative Arrays (Hashtables)**
 - \$hash = @{} Create empty hashtable
 - \$hash.key1 = 1 Assign 1 to key "key1"
 - \$hash.key1 Returns value of key1
 - \$hash["key1"] Returns value of key1
 - \$hash.key1={cmd} Assign code block to key1
 - \$hash.key1(1,2) Run code block in key1 with parameters 1,2
- **Boolean Values**
 - \$true = 1 -eq 1 Assigns True to \$true
 - 1,2,3 -and \$true true
 - \$() -and \$true false
 - @() -and \$true false
 - @(1) -and \$true true
 - "" -and \$true False
 - "word" -and \$true True

Mai 2006 Microsoft Powershell 28


Quick Ref



- **Comparison Operators**
 - band, -bor bitwise and, bitwise or
 - match, -notmatch regex pattern matching
 - like, -notlike globbing pattern matching
 - eq, -ne Equal, Not equal
 - gt, -ge Greater than, greater or equal
 - lt, -le Less than, less or equal
 - is compare types (1 -is int)
- **Case Insensitive variants:**
 - imatch, -inotmatch, -ilike, -inotlike, -ieq, -ine, -igt, -ige, -ilt, -ile
- **Comments**
 - # This is a comment.
 - \$a = "#This is not a comment..."
 - \$a = "something" # ...but this is.

Mai 2006 Microsoft Powershell 29

Quick Ref



- **The "break" commands exits a loop.**
- while (1)


```
{
  $a = something
  if ($a -eq 1) break;
}
```
- **"continue" continues the next iteration of a loop without breaking out of it.**

```
while (1)
{
  $a = something
  if ($a -eq 1) (continue)
  # This line is not reached unless $a == 1
}
# This line is never reached.
```

Mai 2006 Microsoft Powershell 30

Quick Ref



- **Special escape sequences**

```
`0 (null)
`a (alert)
`b (backspace)
`f (form feed)
`n (new line)
`r (carriage return)
`t (tab)
`v (vertical quote)
use `` To specify a ` in a character literal
```

- **Execution Order**

MSH attempts to resolve commands in the following order:
aliases, functions, cmdlets, scripts, executables, normal files

- **Operator Precedence**

```
() {}, @
$, !, [, ], ., &, ++ --, Unary + -, * / %, Binary + -, Comparison
```

Mai 2006

Microsoft Powershell

31

Quick Ref



- **for (initializer; condition; iterator) {}**
for (\$i = 0; \$i -lt 5; \$i++) {write-object \$i}

- **foreach (identifier in pipeline or collection) {}**
\$i = 1,2,3
foreach (\$z in \$i) {write-object \$z}

- **if (condition) {...}**
elseif (condition) {...}
else {...}

- **functions (scripting)**
function MyFunction {
write-object \$args[0]
}

- **Filters (scripting)**
filter MyFilter {
\$_.name
}

Mai 2006

Microsoft Powershell

32

Quick Ref



- **Invoke Operator**

The & operator can be used to invoke the contents of an object. Example:

```
$a = "get-process"
&$a
$a = { get-process | pick-head 2 }
&$a
```

- **Methods can be called on objects**

```
$a.ToUpper()
$a = "This is a string"
$a.SubString(0,3)
$a.SubString(0,($a.length/2))
$a.Substring(($a.length/2), ($a.length/3))
```

- **Object Properties**

An object's properties can be referenced directly with the "." operator.

```
$a = get-date
$a.Date
$a.TimeOfDay.Hours
```

Mai 2006

Microsoft Powershell

33

Quick Ref



- \$\$ contains the last token of last line input into the shell
- \$? Contains that success/fail status of the last operation
- \$^ contains the first token of the last line input into the shell
- DebugPolicy The action to take when data is written via write-debug in a script or WriteDebug in a cmdlet or provider.
- ErrorPolicy The action to take when data is written via write-error in a script or WriteError in a cmdlet or provider.
- HistorySize Number of entries saved in the command history.
- MSHCommandPath The paths where .cmdlet and .cmdletprovider files may be found.
This is the msh equivalent of the CMD.EXE \$PATH.
- ReportErrorShowExceptionClass Set to true indicates that the class name of the exception(s) displayed will be shown. Default at internal startup is false.
- ReportErrorShowInnerException Set to true indicates that the chain of inner exceptions should be shown. Each exception message will be indented from the previous message.
- ReportErrorShowSource Set to true indicates that the assembly name from whence the exception originated will be displayed. Default at internal startup is true.
- ReportErrorShowStackTrace Set to true indicates that the stack trace of the exception will be emitted. Default at internal startup is false.
- ShouldProcessPolicy The action to take when ShouldProcess is used in a cmdlet.
- ShouldProcessReturnPolicy ShouldProcess will return this setting
- VerbosePolicy The action to take when data is written via write-verbose in a script or WriteVerbose in a cmdlet or provider.
- \$_ The current pipeline object, used in script blocks and where
- \$Args Used in creating functions that require parameters
- \$Error Objects which had an error occur while processing that object in a cmdlet.
- \$ErrorPolicy The action to take when data is written via write-error in a script or WriteError in a cmdlet or provider.
- \$foreach Reference to the enumerator in a foreach loop
- \$Home The users home directory; set to %HOMEDRIVE%\%HOMEPATH%
- \$Input Can aid in code blocks that are in the middle of a pipeline, (see code block)
- \$MshHome The install location of MSH
- \$MshHost Information about the current executing host
- \$OFS Output Field Separator
- \$StackTrace contains detailed stack trace information about the last error

Mai 2006

Microsoft Powershell

34

Quick Ref



- **Script Blocks**

Commands and expressions can be stored in a script block object and executed later.

```
$block = {get-process; $a=1}
&$block
```

- **Scripts**

Powershell commands can be stored in and executed from script files.

The file extension for scripts is ".ps1".

Parameters can be passed to a script and a script can return a value.

```
$sum = MyAdder.ps1 1 2 3
```

- **Strings and String Operators**

```
"this is a string, this $variable is expanded"
```

```
'this is a string, this $variable is not expanded'
```

- **String operators**

+ Concatenate two strings

* Repeat a string some number of times

-f Format a string

-replace replace elements in a string

Mai 2006

Microsoft Powershell

35

Quick Ref



- **Switch**

```
$a = 3
switch ($a)
{
  1 {"got one"}
  2 {"got two"}
  3 {"got three"}
}
```

```
$var = "word2"
switch -regex ($var)
{
  "word2" {"Multi-match Exact " + $_}
  "word.*" {"Multi-match Exact1 " + $_}
  default {"Multi-match Default " + $_; break}
  "w.*" {"Previous Break terminated the matching"}
}
```

```
$var = "word1","word2","word3"
switch -regex ($var)
{
  "word1" {"Multi-match Exact " + $_; continue}
  "word2" {"Multi-match Exact " + $_; continue}
  default {"Multi-match Default " + $_; continue}
}
```

Mai 2006

Microsoft Powershell

36

Quick Ref



- **Trap**

Execute a block of code in a terminating error condition. Example:

```
function handler1 { write-host "Hi, I'm a trap handler" }
function handler2 { write-host "Hi, I'm a trap handler2" }
trap [System.Management.Automation.ExecutionFailedException]
{ handler2 ; continue }
trap [System.Management.Automation.ExecutionBreakOnErrorException]
{ handler1 ; continue }
get-content thisisabadfilename -errorp notifystop
set-location thisisabadlocation
```

- **Unary Operators**

- ++ Increment a variable
- Decrement a variable
- + Indicate that a number is positive
- Indicate that a number is negative

- **[type]object cast object to type**

```
@$a = [int]"3"@
@$a + 3@
@6@
```

Quick Ref



- **Scope**

```
$a = 1
$global:a = 1
$local:a = 1
$env:path = "d:\windows"
```

Samples from the demos



```

. #-----
. # objects
. #-----
. "Good afternoon Decus"
. "Good afternoon Decus".gettype()
. "Good afternoon Decus".toupper()
. "Good afternoon Decusxxx".trim('x').toupper().split()
. "$(-+ * 40) `n1A06-DECUS VORTRAG n$(-+ * 40)"
. $a = [datetime] *1.1.2006*
. $a.gettype()
. $a | get-member
. $a = $(c:\boot.ini)
. 1 + *1*
. $arr = @("abc", "def", "ghi")
. $arr
. $arr.toupper()
. $arr[0].toupper()
. $arr | foreach { "$_.toupper() }
. "hello $arr"
. [string]::join(" ", $arr)
. "apfel birne kiwi orange" -replace "kiwi", "pflaume"
. "apfel birne kiwi orange" -match "Bohne|kiwi"
. $matches
. #-----
. # namespace providers
. #-----
. get-psprovider
. get-psdrive
. new-psdrive -name JK1 -Psprovider Filesystem -root d:\monad
. dir JK1:
. cd HKLM:\SYSTEM\CurrentControlSet\Services
. ls w*
. get-ltproperty .

```

Mai 2006

Microsoft Powershell

39