
[HOME](#) [BLOGS](#) [ABOUT](#) [CONTACT](#)

YOU ARE HERE: [HOME](#) / [POWERSHELL SCRIPTS](#) / HOW TO GET REMOTE SYSTEM INFORMATION – PART 1

HOW TO GET REMOTE SYSTEM INFORMATION – PART 1

21/02/2018 by [STEPHANOS](#) – [5 COMMENTS](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

How to get remote system information – Part 1

Scenario:

How to get remote system information with PowerShell – Part 1.

This time we will go through a new script that allows us to get system information and not only from remote systems. This post is the first part from a series of posts that will follow. The script can run on PCs, laptops, physical and virtual servers. Read the post to find out what is needed so you will be able to run the script smoothly.

The script will help you to understand and work with retrieving information from remote systems. We will go through the information later on what we can retrieve with this version and how it works. To help you understand what we are doing and what is needed we will go through some background information. Please note that this is not the only way to do it but at least one way to do it. This is a simple way to do it. It can be done with various ways using also a combination of other languages.

Graphical User Interface (GUI)

In this script we are working with Windows Forms. Windows Forms is a library that you can use to provide GUI on your scripts. It is included in Microsoft .Net Framework by default. As PowerShell is built above Microsoft .Net Framework you can use PowerShell scripting language to create GUI on your scripts and be able to retrieve information, manipulate information and present it on GUI. Windows Forms and controls are static. Controls are placed based on an initial point in the form which is the first top left point. A more professional and flexible way is to use WPF Forms. We will discuss about WPF Forms on another post. To be able to create UI with PowerShell you need first to load the assembly. The command to load the assembly is:

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

After the assembly is created we will need to create our form and the controls we want. First of all we create our form by using the below:

```
$MainForm = New-Object system.Windows.Forms.Form
$MainForm.Text = "Computer Information"
$MainForm.Size = New-Object System.Drawing.Size(1200,800)
$MainForm.ShowDialog()
```

The first command creates the form object, the second command shows the title of the form and the third command is the size of the form in pixels. The last command, will display the form. We have configure more properties based on what we want to achieve. In the script you will see that more properties are configured.

Labels

To create a label inside the form you need to use the below commands.

```
$lbl_ComputerName = New-Object System.Windows.Forms.Label
$lbl_ComputerName.Location = New-Object System.Drawing.Point(0,5)
$lbl_ComputerName.Size = New-Object System.Drawing.Size(150,20)
$lbl_ComputerName.Text = "Computer Name"
$MainForm.Controls.Add($lbl_ComputerName)
```

As for all our UI object we need to create the label object first by using the first command. By using the second command we are providing the location of the label inside the form. For the example above we says that we want the label to start from 0 pixels on x axis and 5 pixels on y axis. The third command is the size of the label. The example above says that that we want to have the label 150 pixels on x axis and 20 pixels on y axis. The next command is the text that we want to show in the label. The last command is the command that we use for all of our objects to present the any control, textbox and label in the form. Please note the the last command needs to be placed before displaying the form.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```
$txt_ComputerName = New-Object System.Windows.Forms.TextBox
$txt_ComputerName.Location = New-Object System.Drawing.Point(150,5)
$txt_ComputerName.Size = New-Object System.Drawing.Size(200,20)
$MainForm.Controls.Add($txt_ComputerName)
```

Buttons

Buttons are created and displayed the same way as labels and text boxes. The difference on buttons is that we want them to do something when we click them.

```
$btn_System = New-Object System.Windows.Forms.Button
$btn_System.Location = New-Object System.Drawing.Point(5,50)
$btn_System.Size = New-Object System.Drawing.Size(145,25)
$btn_System.Text = "System"
$btn_System.Add_Click($System_info)
$MainForm.Controls.Add($btn_System)
```

The above commands will create, size, locate, and display the button in the form. The 5th command is the command that we use to add the action that we want to be done when we click the button. Note here that `$System_info` is a function that we call when the button is clicked.

CIM Standard

Microsoft was using WMI to retrieve information from systems. WMI was using CIM standard to provide this information. With `Get-WMIObject` you were able to retrieve the information that you wanted. WMI is built on the WBEM and CIM standards from the Distributed Management Task Force (DMTF). In PowerShell 3.0, Microsoft introduced CIM commands to directly access this information. In this script we are using CIM commands to retrieve the information. As you will see below in the script, functions that are called from the buttons, are using `Get-CIMInstance` to get the information.

Information that we get from the systems

By using this script, in its initial version, you are able to get information about the systems.

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

- Processor
- RAM Memory
- Motherboard
- Physical Drives
- Logical Drives
- Video Controller
- Network Adapters
- Monitors
- Operating System
- Keyboard
- Pointing Devices
- CD/DVD-drives
- Sound Devices
- Printers
- Software installed (Use this at your own risk. Win32_Product class is not stable by design and might was issues on remote system.)
- Processes
- Services

For each of the above we get some information as it is provided by the system. If any of the fields are empty or any action does not provide any result, then it means that the system that you are trying to get the information does not provide it. With this script you will be able to get all this information remotely without the need to login with RDP on the system. If you do not provide any computer name, then the script will provide the information of the local system.

Requirements

First of all as we are using CIM commands you will need to have at least PowerShell 3.0 installed on the computer/ server that you will run the script. Secondly the user that will run the script need to have remote management permission on the remote system to be able to connect and get the

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

connect to the remote system. If you are not running WINRM on them then you will not be able to get any information. Note that script will work if you run it locally as CIM uses DCOM to connect to the local system and if no computer name will be specified the script will provide information of the local system.

You can download the script [here](#) or copy it from below.

Hope you like it. If you have any questions or anything else please let me know in the comments below.

Stay tuned for the next part of this series.

Related Links:

- [How to get remote system information – Part 2](#)
- [How to get remote system information – Part 3](#)
- [How to get remote system information – Part 4](#)
- [PowerShell script to get computer information](#)
- [Get-CimInstance \(CimCmdlets\) | Microsoft Docs](#)
- [Out-String – Microsoft Docs](#)
- [Add-Type – Microsoft Docs](#)
- [New-Object – Microsoft Docs](#)

Solution / Script:

```
$System_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $System = Get-CimInstance -Class Win32_ComputerSystem -ComputerName :  
    $lbl_sysinfo.Text = $System | FL -Property Name,Manufacturer,Model,Pro  
    DNSHostName,NumberOfProce  
    TotalPhysicalMemory,Curren  
    HypervisorPresent,PrimaryV  
}  
  
$bios_info = {
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```

$CPU_info = {
    $ComputerName = $txt_ComputerName.Text
    $CPU = Get-CimInstance -Class Win32_Processor -ComputerName $ComputerName
    $lbl_sysinfo.Text = $CPU | FL -Property DeviceID,Manufacturer,Name,CPUID,
                                                                    LoadPercentage,CurrentClockSpeed
}

$RAM_info = {
    $ComputerName = $txt_ComputerName.Text
    $RAM = Get-CimInstance -Class Win32_PhysicalMemory -ComputerName $ComputerName
    $lbl_sysinfo.Text = $RAM | FL -Property Tag,DeviceLocator,Manufacturer,
                                                                    Capacity,Speed | Out-String
}

$MB_info = {
    $ComputerName = $txt_ComputerName.Text
    $MB = Get-CimInstance -Class Win32_BaseBoard -ComputerName $ComputerName
    $lbl_sysinfo.Text = $MB | FL -Property Manufacturer,Model,Version | Out-String
}

$PhysicalDrives_info = {
    $ComputerName = $txt_ComputerName.Text
    $PhysicalDrives = Get-CimInstance -Class Win32_DiskDrive -ComputerName $ComputerName
    $lbl_sysinfo.Text = $PhysicalDrives | FL -Property DeviceID,FirmwareRevision,
                                                                    SerialNumber,InterfaceType,
                                                                    TotalHeads,TotalSectors,
                                                                    SectorsPerTrack,Cylinders
}

$LogicalDrives_info = {
    $ComputerName = $txt_ComputerName.Text
    $LogicalDrives = Get-CimInstance -Class Win32_LogicalDisk -ComputerName $ComputerName
    $lbl_sysinfo.Text = $LogicalDrives | FL -Property DeviceID,Description,
                                                                    VolumeSerialNumber
}

$GPU_info = {
    $ComputerName = $txt_ComputerName.Text
    $GPU = Get-CimInstance -Class Win32_VideoController -ComputerName $ComputerName
    $lbl_sysinfo.Text = $GPU | FL -Property DeviceID,Name,VideoProcessor,
                                                                    DriverVersion,VideoModeDescription,
                                                                    CurrentVerticalResolution,Cur

```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```

$ComputerName = $txt_ComputerName.Text
$Network = Get-CimInstance -Class Win32_NetworkAdapter -ComputerName $ComputerName
$lbl_sysinfo.Text = $Network | FL -Property DeviceID,Name,Manufacturer,NetConnectionID,NetEnabled
}

$Monitor_info = {
    $ComputerName = $txt_ComputerName.Text
    $Monitor = Get-CimInstance -Class Win32_DesktopMonitor -ComputerName $ComputerName
    $lbl_sysinfo.Text = $Monitor | FL -Property DeviceID,Name,MonitorManufacturer,PixelPerYLogicalInch,ScreenSize
}

$OS_info = {
    $ComputerName = $txt_ComputerName.Text
    $OS = Get-CimInstance -Class Win32_OperatingSystem -ComputerName $ComputerName
    $lbl_sysinfo.Text = $OS | FL -Property Name,Manufacturer,Caption,Version,IsX64,IsX86,IsPortableOperatingSystem,PrimaryPartition,RegisteredUser,SerialNumber,SupportURL,FreePhysicalMemory,FreeSpaceInRootDirectory,TotalVisibleMemorySize,Status
}

$Keyboard_info = {
    $ComputerName = $txt_ComputerName.Text
    $Keyboard = Get-CimInstance -Class Win32_Keyboard -ComputerName $ComputerName
    $lbl_sysinfo.Text = $Keyboard | FL -Property Description,Caption,NumLock
}

$Mouse_info = {
    $ComputerName = $txt_ComputerName.Text
    $Mouse = Get-CimInstance -Class Win32_PointingDevice -ComputerName $ComputerName
    $lbl_sysinfo.Text = $Mouse | FL -Property Description,Name,HardwareType
}

$CDROM_info = {
    $ComputerName = $txt_ComputerName.Text
    $CDROM = Get-CimInstance -Class Win32_CDROMDrive -ComputerName $ComputerName
    $lbl_sysinfo.Text = $CDROM | FL -Property Drive,Name,Caption,Description,CapabilityDescriptions,Media
}

$Sound_info = {

```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok


```

$Printers_info = {
    $ComputerName = $txt_ComputerName.Text
    $Printers = Get-CimInstance -Class Win32_Printer -ComputerName $Comp
    $lbl_sysinfo.Text = $Printers | FL -Property DeviceID,Name,Horizontal
                                Hidden,KeepPrintedJobs,
                                WorkOffline,CapabilityD
}

$Software_info = {
    $ComputerName = $txt_ComputerName.Text

    $Software = Get-CimInstance -Class Win32_Product -ComputerName $Comp
    $lbl_sysinfo.Text = $Software | FL -Property Name,Version,Description
}

$Process_info = {
    $ComputerName = $txt_ComputerName.Text
    $Process = Get-CimInstance -Class Win32_Process -ComputerName $Comp
    $lbl_sysinfo.Text = $Process | FL -Property ProcessName,Path,Creation
}

$Services_info = {
    $ComputerName = $txt_ComputerName.Text
    $Services = Get-CimInstance -Class Win32_Service -ComputerName $Comp
    $lbl_sysinfo.Text = $Services | FL -Property Name,DisplayName,Descri
}

Add-Type -AssemblyName System.Windows.Forms
$MainForm = New-Object system.Windows.Forms.Form
$MainForm.Text = "Computer Information"
$MainForm.Size = New-Object System.Drawing.Size(1200,800)
$MainForm.AutoScroll = $True
$MainForm.MinimizeBox = $True
$MainForm.MaximizeBox = $True
$MainForm.WindowState = "Normal"
$MainForm.SizeGripStyle = "Hide"
$MainForm.ShowInTaskbar = $True
$MainForm.Opacity = 1
$MainForm.StartPosition = "CenterScreen"
$Font = New-Object System.Drawing.Font("Consolas",12,[System.Drawing.Font
$MainForm.Font = $Font
$lbl_ComputerName = New-Object System.Windows.Forms.Label
$lbl_ComputerName.Location = New-Object System.Drawing.Point(0,5)

```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```

$lbl_sysinfo = New-Object System.Windows.Forms.Label
$lbl_sysinfo.Location = New-Object System.Drawing.Point(155,50)
$lbl_sysinfo.Size = New-Object System.Drawing.Size(500,500)
$lbl_sysinfo.AutoSize = $True
$lbl_sysinfo.Text = ""
$MainForm.Controls.Add($lbl_sysinfo)

$txt_ComputerName = New-Object System.Windows.Forms.TextBox
$txt_ComputerName.Location = New-Object System.Drawing.Point(150,5)
$txt_ComputerName.Size = New-Object System.Drawing.Size(200,20)
$txt_ComputerName.Font = $Font

$MainForm.Controls.Add($txt_ComputerName)

$btn_System = New-Object System.Windows.Forms.Button
$btn_System.Location = New-Object System.Drawing.Point(5,50)
$btn_System.Size = New-Object System.Drawing.Size(145,25)
$btn_System.Text = "System"
$btn_System.Add_Click($System_info)
$MainForm.Controls.Add($btn_System)

$btn_BIOS = New-Object System.Windows.Forms.Button
$btn_BIOS.Location = New-Object System.Drawing.Point(5,75)
$btn_BIOS.Size = New-Object System.Drawing.Size(145,25)
$btn_BIOS.Text = "BIOS"
$btn_BIOS.Add_Click($bios_info)
$MainForm.Controls.Add($btn_BIOS)

$btn_CPU = New-Object System.Windows.Forms.Button
$btn_CPU.Location = New-Object System.Drawing.Point(5,100)
$btn_CPU.Size = New-Object System.Drawing.Size(145,25)
$btn_CPU.Text = "CPU"
$btn_CPU.Add_Click($cpu_info)
$MainForm.Controls.Add($btn_CPU)

$btn_RAM = New-Object System.Windows.Forms.Button
$btn_RAM.Location = New-Object System.Drawing.Point(5,125)
$btn_RAM.Size = New-Object System.Drawing.Size(145,25)
$btn_RAM.Text = "RAM"
$btn_RAM.Add_Click($ram_info)
$MainForm.Controls.Add($btn_RAM)

$btn_MB = New-Object System.Windows.Forms.Button
$btn_MB.Location = New-Object System.Drawing.Point(5,150)

```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```
$btn_PhysicalDrives = New-Object System.Windows.Forms.Button
$btn_PhysicalDrives.Location = New-Object System.Drawing.Point(5,175)
$btn_PhysicalDrives.Size = New-Object System.Drawing.Size(145,25)
$btn_PhysicalDrives.Text = "Physical Drives"
$btn_PhysicalDrives.Add_Click($PhysicalDrives_info)
$MainForm.Controls.Add($btn_PhysicalDrives)

$btn_LogicalDrives = New-Object System.Windows.Forms.Button
$btn_LogicalDrives.Location = New-Object System.Drawing.Point(5,200)
$btn_LogicalDrives.Size = New-Object System.Drawing.Size(145,25)

$btn_LogicalDrives.Text = "Logical Drives"
$btn_LogicalDrives.Add_Click($LogicalDrives_info)
$MainForm.Controls.Add($btn_LogicalDrives)

$btn_Graphics = New-Object System.Windows.Forms.Button
$btn_Graphics.Location = New-Object System.Drawing.Point(5,225)
$btn_Graphics.Size = New-Object System.Drawing.Size(145,25)
$btn_Graphics.Text = "Graphics"
$btn_Graphics.Add_Click($GPU_info)
$MainForm.Controls.Add($btn_Graphics)

$btn_Network = New-Object System.Windows.Forms.Button
$btn_Network.Location = New-Object System.Drawing.Point(5,250)
$btn_Network.Size = New-Object System.Drawing.Size(145,25)
$btn_Network.Text = "Network"
$btn_Network.Add_Click($Network_info)
$MainForm.Controls.Add($btn_Network)

$btn_Monitors = New-Object System.Windows.Forms.Button
$btn_Monitors.Location = New-Object System.Drawing.Point(5,275)
$btn_Monitors.Size = New-Object System.Drawing.Size(145,25)
$btn_Monitors.Text = "Monitors"
$btn_Monitors.Add_Click($Monitor_info)
$MainForm.Controls.Add($btn_Monitors)

$btn_OS = New-Object System.Windows.Forms.Button
$btn_OS.Location = New-Object System.Drawing.Point(5,300)
$btn_OS.Size = New-Object System.Drawing.Size(145,25)
$btn_OS.Text = "OS"
$btn_OS.Add_Click($OS_info)
$MainForm.Controls.Add($btn_OS)
```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```

$btn_Keyboard = New-Object System.Windows.Forms.Button
$MainForm.Controls.Add($btn_Keyboard)

$btn_Mouse = New-Object System.Windows.Forms.Button
$btn_Mouse.Location = New-Object System.Drawing.Point(5,350)
$btn_Mouse.Size = New-Object System.Drawing.Size(145,25)
$btn_Mouse.Text = "Mouse"
$btn_Mouse.Add_Click($Mouse_info)
$MainForm.Controls.Add($btn_Mouse)

$btn_CDROM = New-Object System.Windows.Forms.Button

$btn_CDROM.Location = New-Object System.Drawing.Point(5,375)
$btn_CDROM.Size = New-Object System.Drawing.Size(145,25)
$btn_CDROM.Text = "CDROM"
$btn_CDROM.Add_Click($CDROM_info)
$MainForm.Controls.Add($btn_CDROM)

$btn_Sound = New-Object System.Windows.Forms.Button
$btn_Sound.Location = New-Object System.Drawing.Point(5,400)
$btn_Sound.Size = New-Object System.Drawing.Size(145,25)
$btn_Sound.Text = "Sound"
$btn_Sound.Add_Click($Sound_info)
$MainForm.Controls.Add($btn_Sound)

$btn_Printers = New-Object System.Windows.Forms.Button
$btn_Printers.Location = New-Object System.Drawing.Point(5,425)
$btn_Printers.Size = New-Object System.Drawing.Size(145,25)
$btn_Printers.Text = "Printers"
$btn_Printers.Add_Click($Printers_info)
$MainForm.Controls.Add($btn_Printers)

$btn_Software = New-Object System.Windows.Forms.Button
$btn_Software.Location = New-Object System.Drawing.Point(5,450)
$btn_Software.Size = New-Object System.Drawing.Size(145,25)
$btn_Software.Text = "Software"
$btn_Software.Add_Click($Software_info)
$MainForm.Controls.Add($btn_Software)

$btn_Process = New-Object System.Windows.Forms.Button
$btn_Process.Location = New-Object System.Drawing.Point(5,475)
$btn_Process.Size = New-Object System.Drawing.Size(145,25)
$btn_Process.Text = "Process"
$btn_Process.Add_Click($Process_info)

```

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

```

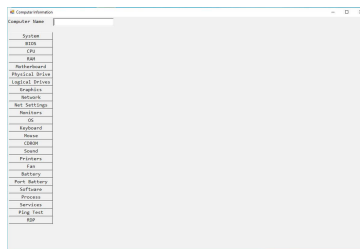
$btn_Services.Size = New-Object System.Drawing.Size(150,25)
$btn_Services.Text = "Services"
$btn_Services.Add_Click($Services_info)
$MainForm.Controls.Add($btn_Services)

$MainForm.ShowDialog()

```

[f Share](#)
[t Tweet](#)
[g+ Share](#)
[in Share](#)
[p Pin](#)

Summary



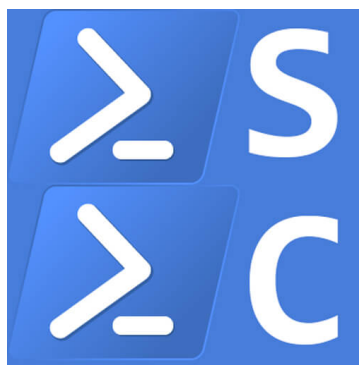
Article Name How to get remote system information - Part 1

Description How to get remote system information with PowerShell. Getting remote computer or server information has never been easier with this PowerShell script and its GUI. Stephanos Constantinou Blog - PowerShell Scripting

Author Stephanos

Publisher Name Stephanos Constantinou Blog

Publisher Logo



FILED UNDER: POWERSHELL SCRIPTS

TAGGED WITH: ADD-TYPE, GET-CIMINSTANCE, NEW-OBJECT, OUT-STRING, WIN32_BASEBOARD, WIN32_BIOS, WIN32_CDROMDRIVE, WIN32_COMPUTERSYSTEM, WIN32_DESKTOPMONITOR, WIN32_DISKDRIVE, WIN32_KEYBOARD, WIN32_LOGICALDISK, WIN32_NETWORKADAPTER, WIN32_OPERATINGSYSTEM, WIN32_PHYSICALMEMORY, WIN32_POINTINGDEVICE, WIN32_PRINTER, WIN32_PROCESS, WIN32_PROCESSOR,

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

Comments

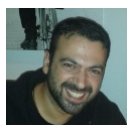


[Daniel](#) says

[23/09/2018 at 11:27](#)

Pretty neat!

[Reply](#)



[Stephanos](#) says

[24/09/2018 at 23:20](#)

Thank you Daniel for your comment.

[Reply](#)

Trackbacks

How to get remote system information. - How to Code .NET says:

[21/02/2018 at 19:00](#)

[...] on February 20, 2018by admin submitted by /u/SConstantinou [link]

[comments] No comments [...]

How to get remote system information – Part 2 - Stephanos Constantinou says:

[26/02/2018 at 16:22](#)

[...] How to get remote system information – Part 1 [...]

How to get remote system information - Part 3 - Stephanos Constantinou says:

[14/02/2018 at 17:25](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

Leave a Reply

Your email address will not be published.

Comment

Name

Email

Website

POST COMMENT

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok

Recent Posts

ICS Cube Product
Review 26/04/2019
PowerShell Module
SysInfo v1.2.0
15/03/2019
PowerShell Module
SysInfo v1.1.2
13/11/2018
PowerShell Module
SysInfo 24/10/2018
Get-VoltageProbe
24/10/2018
Get-VideoController
24/10/2018
Get-USBController
24/10/2018
Get-TrackPoint
24/10/2018
Get-TrackBall
24/10/2018
Get-TouchScreen
24/10/2018

Categories

Modules Cmdlets (57)
PowerShell Modules (5)
PowerShell Scripts (38)
PowerShell Tutorials
(35)
Software Reviews (2)

Archives

April 2019 (1)
March 2019 (1)
November 2018 (1)
October 2018 (56)
September 2018 (13)
August 2018 (9)
July 2018 (6)
June 2018 (8)
May 2018 (7)
April 2018 (9)
March 2018 (4)
February 2018 (6)
January 2018 (12)
December 2017 (4)

Blogroll

Planet PowerShell
Reddit – PowerShell
PowerShell Magazine
PowerShell.org
PowerShell Team Blog
Hey, Scripting Guy! Blog
Mike F Robbins
PowerShell Explained
with Kevin Marquette
Mike Kanakos –
Network Admin
The Lonely
Administrator
AskME4Tech

© 2020 · Stephanos Constantinou Blog

[HOME](#) [BLOGS](#) [ABOUT](#) [CONTACT](#)

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

Ok