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

YOU ARE HERE: [HOME](#) / [POWERSHELL SCRIPTS](#) / HOW TO GET REMOTE SYSTEM INFORMATION – PART 3

HOW TO GET REMOTE SYSTEM INFORMATION – PART 3

14/03/2018 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 3

Scenario:

How to get remote system information – Part 3

I hope you like the previous two parts of this series. In this part, we will add some extra functionality in the script and a few more changes for clearer output. Until part 2 we are able to get the information from the system, which is show on the screen. We are able to perform some ping tests, using the Test-connection command from PowerShell. And also we are able to have remote desktop connection to the remote system. In part 3 we will add extra functionality such as NetStat, export the results in a text file and email the results to the required recipient.

NetStat functionality

Below is the function that we call to perform the NetStat test. If the administrator will not provide any computer name for remote system, we are using the Get-NetTCPConnection to find our information. As the command does not support to provide a computer name for the connection, then we have to use Invoke-Command to perform the change on the remote system.

```
$NetStat = {  
    $ComputerName = $txt_ComputerName.Text  
  
    if ($ComputerName -eq ""){  
        $LocalNetStat = Get-NetTCPConnection  
        $lbl_sysinfo.Text = "NetStat Information"  
        $lbl_sysinfo.Text += $LocalNetStat | FT | Out-String}  
    else{  
        $RemoteNetStat = Invoke-Command -ComputerName $ComputerN.
```

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

[adinsenter name="In Article"]

Export the results

Until part 2 we were not able able to export our results. Now with the below function we are able to export the results in a text file and you will be able to use them at a later stage to prove what was the status of the system at that time. The file that will be saved will include the system name, date and time and what was the information that we have extracted from the system. The test file will be saved under C:\Scripts. Please note that you need to create the folder before you run the script. If the file already exists, it will be replaced. So you need to make sure if you want to keep both file to save the old one in another place or rename it before exporting the results. During the process of exporting, we are checking if the \$ComputerName variable is empty. This happens when we get information from local system. To overcome this issue, we get the system name just before exporting the results to file so the text file that will be exported will have the actual system name.

```
$TextFile = {  
    $ExportOption.Close()  
  
    if ($ComputerName -eq ""){  
        $ComputerName = (Get-CimInstance -Class Win32_ComputerSy  
  
        $lbl_sysinfo.Text | Out-File C:\Scripts\$ComputerName.txt}
```

Email the results

As a second option to provide the results to someone, we are able to email the results to the recipients that we want. By email the results, you will be able to define the recipients that you would like to send the results to. Emailing the results, as you are able to see below is additional to the above method of exporting the result. The results that we see on the screen, are exported in a

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

that will be used to send the email. In order to enable the option to have multiple domains, we have to split the string based on a specific character. The character that we use is comma (.). After that, we get the credentials of the mailbox that will be used to send the email. This will fill also the "From" parameter for our command.

```
$Email = {  
    if ($ComputerName -eq ""){  
        $ComputerName = (Get-CimInstance -Class Win32_ComputerSy  
  
$lbl_sysinfo.Text | Out-File C:\Scripts\$ComputerName.txt  
  
$To = @(($txt_Recipients.Text) -split ',')  
$Attachement = "C:\Scripts\$ComputerName.txt"  
$Recipients.Close()  
  
$EmailCredentials = Get-Credential  
  
$From = $EmailCredentials.UserName  
  
$EmailParameters = @{  
    To = $To  
    Subject = "System Information - $ComputerName"  
    Body = "Please find attached the information that you ha'  
    Attachments = $Attachement  
    UseSsl = $True  
  
    Port = "587"  
    SmtpServer = "smtp.office365.com"  
    Credential = $EmailCredentials  
    From = $From}  
  
send-mailmessage @EmailParameters }
```

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

In order to achieve the above we have to create another form that we will be able first to select our export option. As you are able to see in the screenshot below, a form appears on the screen when we click the export button showing us the two options that we have. When we select the text file option the file will be exported and the form will close.

If the administrator will select to email the results then another form will appear to enter the recipients to receive the email. If there are multiple

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

Currently, the script uses Office 365 to send emails and port 587. In case you are using a different configuration then you will need to change the SMTP server and port to the correct one.

Do you like to include some other functions in the script?

If there are some functions that you might like to add, please let me know. I will check if I am able to add them in the script.

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

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.

[adinsserter name="In Article"]

Related Links:

- [How to get remote system information â€“ Part 1](#)
- [How to get remote system information â€“ Part 2](#)
- [How to get remote system information â€“ Part 4](#)
- [Get-CimInstance \(CimCmdlets\) | Microsoft Docs](#)
- [New-Object – Microsoft Docs](#)
- [Test-Connection – Microsoft Docs](#)
- [Out-File – Microsoft Docs](#)
- [Get-Credential – Microsoft Docs](#)
- [Send-MailMessage – Microsoft Docs](#)
- [Get-NetTCPConnection – Microsoft Docs](#)
- [Invoke-Command – Microsoft Docs](#)
- [Add-Type – Microsoft Docs](#)

[adinsserter name="In Article"]

Solution / Script:

```
<#
```

```
.SYNOPSIS
```

```
    Name: Get-SysInfo.ps1
```

```
    The purpose of this script is to retrieve information of remote
```

```
.DESCRIPTION
```

```
    This is a simple script with UI to retrieve information of remote
```

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

and operating system through a very simple and functioning of Remote Desktop and export the results in a text file or email

.RELATED LINKS

“[Home](#)

.NOTES

Version: 1.3

Updated: 13-03-2018 - Added ability to email results
- Added date and time for the results
- Updated Description

Updated: 02-03-2018 - Added ability to export results
- Added TCP Connection information
- Added Title for each information

Updated: 25-02-2018 - Added Fan Information
- Added Battery Information
- Added Portable Battery Information
- Added Network Settings Information
- Added ping connection test of remote
- Added Remote Desktop connection
- Warning for the use of Win32_Proc
- Added option for Win32Reg_AddReg
- Added visibility to Taskbar
- Added Help information

Release Date: 22-02-2018

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

Run the Get-SysInfo script to retrieve the information.

Get-SysInfo.ps1

#>

```
$System_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $System = Get-CimInstance -Class Win32_ComputerSystem -ComputerName $ComputerName  
    $lbl_sysinfo.Text = "System Information - $(Get-Date)"  
    $lbl_sysinfo.Text += $System | FL -Property Name,  
                                                                    Manufacturer,  
                                                                    Model,  
                                                                    PartOfDomain,  
                                                                    Domain,  
                                                                    Workgroup,  
                                                                    DNSHostName,  
                                                                    NumberOfProcessors,  
                                                                    NumberOfLogicalProcessors,  
                                                                    TotalPhysicalMemory,  
                                                                    CurrentTimeZone,  
                                                                    DaylightInEffect,  
                                                                    HypervisorPresent,  
                                                                    PrimaryOwnerName,  
                                                                    UserName | Out-Null  
}
```

```
$bios_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $Bios = Get-CimInstance -Class Win32_BIOS -ComputerName $ComputerName  
    $lbl_sysinfo.Text = "BIOS Information - $(Get-Date)"  
    $lbl_sysinfo.Text += $Bios | FL -Property Name,  
                                                                    SerialNumber,  
                                                                    Version,  
                                                                    BIOSVersion,  
}
```

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 = Get-CimInstance -Class Win32_Processor -ComputerName
$lbl_sysinfo.Text = "CPU Information - $(Get-Date)"
$lbl_sysinfo.Text += $CPU | FL -Property DeviceID,
                                Manufacturer,
                                Name,
                                Caption,
                                L2CacheSize,
                                L3CacheSize,
                                LoadPercentage,
                                CurrentClockSpeed

```

```

$RAM_info = {
    $ComputerName = $txt_ComputerName.Text
    $RAM = Get-CimInstance -Class Win32_PhysicalMemory -ComputerName
    $lbl_sysinfo.Text = "RAM Information - $(Get-Date)"
    $lbl_sysinfo.Text += $RAM | FL -Property Tag,
                                DeviceLocator,
                                Manufacturer,
                                PartNumber,
                                SerialNumber,
                                Capacity,
                                Speed | Out-String
}

```

```

$MB_info = {
    $ComputerName = $txt_ComputerName.Text
    $MB = Get-CimInstance -Class Win32_BaseBoard -ComputerName
    $lbl_sysinfo.Text = "MotherBoard Information - $(Get-Date)"
    $lbl_sysinfo.Text += $MB | FL -Property Manufacturer,
                                Model,
                                Version | Out-String
}

```

```

$PhysicalDrives_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

Firmware
Manufac
Model,
MediaTy
SerialN
Interfa
Partiti
Size,
TotalCy
TotalHe
TotalSe
TotalTr
TracksP
Sectors
Capabil
Capabil
Status

```
$LogicalDrives_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $LogicalDrives = Get-CimInstance -Class Win32_LogicalDisk -  
    $lbl_sysinfo.Text = "Logical Drives Information - $(Get-Dat  
    $lbl_sysinfo.Text += $LogicalDrives | FL -Property DeviceID  
                                Descript  
                                VolumeNa  
                                Provider  
                                Size,  
                                FreeSpac  
                                VolumeSe  
                                FileSyste  
                                Compress
```

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.Text += $GPU | FL -Property DeviceID,
Name,
VideoProcessor,
AdapterDACType,
AdapterRAM,
DriverDate,
DriverVersion,
VideoModeDescription,
CurrentBitsPerPixel,
CurrentHorizontalResolution,
CurrentVerticalResolution,
CurrentNumberOfColors,
CurrentRefreshRate,
MaxRefreshRate,
MinRefreshRate,
Status | Out-String

```

```

$Network_info = {
    $ComputerName = $txt_ComputerName.Text
    $Network = Get-CimInstance -Class Win32_NetworkAdapter -ComputerName $ComputerName
    $lbl_sysinfo.Text = "Network Devices Information - $(Get-Date -Format 'dd/MM/yyyy HH:mm:ss')
    $lbl_sysinfo.Text += $Network | FL -Property DeviceID,
Name,
Manufacturer,
ProductName,
ServiceName,
MACAddress,
AdapterType,
NetConnectionID,
NetEnabled,
Speed,
PhysicalAdaptation
}

```

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

```

$NetSettings = Get-CimInstance -Class Win32_NetworkAdapterConfiguration
$lbl_sysinfo.Text = "Network Configuration Information - $(Get-Date)"
$lbl_sysinfo.Text += $NetSettings | FL -Property Description,
DHCPEnabled,
DHCPLeaseObtained,
DNSDomain,
DNSDomainSuffix,
DHCPServer,
DNSHostName,
DNSServerSearchOrder,
DomainDNSResolver,
FullDNSRegistration,
IPEnabled,
IPAddress,
DefaultIPGateway,
IPSubnet,
MACAddress,
ServiceName

```

```

$Monitor_info = {
    $ComputerName = $txt_ComputerName.Text
    $Monitor = Get-CimInstance -Class Win32_DesktopMonitor -ComputerName $ComputerName
    $lbl_sysinfo.Text = "Monitors Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Monitor | FL -Property DeviceID,
Name,
MonitorManufacturer,
MonitorType,
PixelsPerXLogicalInch,
PixelPerYLogicalInch,
ScreenHeight,
ScreenWidth,
Status | Out-String
}

```

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.Text = "Operating System Information - $(Get-Di
$lbl_sysinfo.Text += $OS | FL -Property Name,
                                Manufacturer,
                                Caption,
                                Version,
                                MUILanguages,
                                BuildNumber,
                                BuildType,
                                InstallDate,
                                OSArchitecture,
                                PortableOperatingSy
                                Primary,
                                BootDevice,
                                LastBootUpTime,
                                LocalDateTime,
                                CurrentTimeZone,
                                RegisteredUser,
                                SerialNumber,
                                SystemDevice,
                                SystemDirectory,
                                SystemDrive,
                                WindowsDirectory,
                                EncryptionLevel,
                                FreePhysicalMemory,
                                FreeSpaceInPagingFi
                                FreeVirtualMemory,
                                SizeStoredInPagingF
                                TotalVirtualMemoryS
                                TotalVisibleMemoryS
                                Status | Out-String

```

```
$Keyboard_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

Caption,
NumberOfFunct

```
$Mouse_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $Mouse = Get-CimInstance -Class Win32_PointingDevice -ComputerName $ComputerName  
    $lbl_sysinfo.Text = "Pointing Device Information - $(Get-Date)"  
    $lbl_sysinfo.Text += $Mouse | FL -Property Description,  
                           Name,  
                           HardwareType,  
                           Manufacturer | Out-Null
```

```
$CDROM_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $CDROM = Get-CimInstance -Class Win32_CDROMDrive -ComputerName $ComputerName  
    $lbl_sysinfo.Text = "CD-ROM Drives Information - $(Get-Date)"  
    $lbl_sysinfo.Text += $CDROM | FL -Property Drive,  
                           Name,  
                           Caption,  
                           Description,  
                           Manufacturer,  
                           MediaType,  
                           MfrAssignedRevision,  
                           CapabilityDescription,  
                           MediaLoaded | Out-Null
```

```
$Sound_info = {  
    $ComputerName = $txt_ComputerName.Text  
    $Sound = Get-CimInstance -Class Win32_SoundDevice -ComputerName $ComputerName  
    $lbl_sysinfo.Text = "Sound Devices Information - $(Get-Date)"  
    $lbl_sysinfo.Text += $Sound | FL -Property DeviceID,  
                           Name,
```

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
$Printers = Get-CimInstance -Class Win32_Printer -ComputerName $ComputerName
$lbl_sysinfo.Text = "Printers Information - $(Get-Date)"
$lbl_sysinfo.Text += $Printers | FL -Property DeviceID,
                                                                Name,
                                                                HorizontalResolution,
                                                                VerticalResolution,
                                                                Default,
                                                                DriverName,
                                                                Direct,
                                                                Network,
                                                                Local,
                                                                Hidden,
                                                                KeepPrintedJobs,
                                                                PrintJobDataT,
                                                                PrintProcessors,
                                                                PortName,
                                                                Shared,
                                                                ServerName,
                                                                SpoolEnabled,
                                                                WorkOffline,
                                                                CapabilityDes,
                                                                Status | Out-Null

```

```

$Fan_info = {
    $ComputerName = $txt_ComputerName.Text
    $Fan = Get-CimInstance -Class Win32_Fan -ComputerName $ComputerName
    $lbl_sysinfo.Text = "Fans Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Fan | FL -Property Name,
                                                                Caption,
                                                                Description,
                                                                InstallDate,

```

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


```

$Battery_info = {
    $ComputerName = $txt_ComputerName.Text
    $Battery = Get-CimInstance -Class Win32_Battery -ComputerName $ComputerName
    $lbl_sysinfo.Text = "Battery Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Battery | FL -Property * | Out-String

$PortBattery_info = {
    $ComputerName = $txt_ComputerName.Text
    $PortBattery = Get-CimInstance -Class Win32_PortableBattery -ComputerName $ComputerName
    $lbl_sysinfo.Text = "Portable Battery Information - $(Get-Date)"
    $lbl_sysinfo.Text += $PortBattery | FL -Property * | Out-String

$Software_info = {
    $ComputerName = $txt_ComputerName.Text

$Product = {
    $Warning = [System.Windows.MessageBox]::Show('Are you sure you want to continue?')

    switch ($Warning){
        Yes {$SoftwareOption.Close()
            $Software = Get-CimInstance -Class Win32Reg_Product -ComputerName $ComputerName
            $lbl_sysinfo.Text = "Software Information - $(Get-Date)"
            $lbl_sysinfo.Text += $Software | FL -Property Name, Version, Description, Vendor, InstalledOn, InstalledBy, HelpLink, URLInfo, URLUpdate

        No {Break}
    }
}

```

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

```

$SoftwareOption.Close()
$Software = Get-CimInstance -Class Win32Reg_AddRemovePro
$lbl_sysinfo.Text = "Software Information - $(Get-Date)"
$lbl_sysinfo.Text += $Software | FL -Property DisplayName
                                                    InstallDate
                                                    Publisher,
                                                    Version | O

```

```

$SoftwareOption = New-Object system.Windows.Forms.Form
$SoftwareOption.Text = "Class Option"
$SoftwareOption.Size = New-Object System.Drawing.Size(500,1
$SoftwareOption.AutoSize = $False
$SoftwareOption.AutoScroll = $False
$SoftwareOption.MinimizeBox = $False
$SoftwareOption.MaximizeBox = $False
$SoftwareOption.WindowState = "Normal"
$SoftwareOption.SizeGripStyle = "Hide"
$SoftwareOption.ShowInTaskbar = $True
$SoftwareOption.Opacity = 1
$SoftwareOption.FormBorderStyle = "Fixed3D"
$SoftwareOption.StartPosition = "CenterScreen"

```

```

$lbl_SoftwareOption = New-Object System.Windows.Forms.Label
$lbl_SoftwareOption.Location = New-Object System.Drawing.Po
$lbl_SoftwareOption.Size = New-Object System.Drawing.Size(50
$lbl_SoftwareOption.Text = "Please select the class that you
$lbl_SoftwareOption.Font = $Font
$SoftwareOption.Controls.Add($lbl_SoftwareOption)

```

```

$btn_Product = New-Object System.Windows.Forms.Button
$btn_Product.Location = New-Object System.Drawing.Point(10,
$btn_Product.Size = New-Object System.Drawing.Size(230,25)

```

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

```

$btn_AddRemove = New-Object System.Windows.Forms.Button
$btn_AddRemove.Location = New-Object System.Drawing.Point(2
$btn_AddRemove.Size = New-Object System.Drawing.Size(230,25
$btn_AddRemove.Text = "Win32_AddRemovePrograms"
$btn_AddRemove.Font = $Font
$btn_AddRemove.Add_Click($AddRemove)
$SoftwareOption.Controls.Add($btn_AddRemove)

$SoftwareOption.ShowDialog()
}

```

```

$Process_info = {
    $ComputerName = $txt_ComputerName.Text
    $Process = Get-CimInstance -Class Win32_Process -ComputerName
    $lbl_sysinfo.Text = "Processes Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Process | FL -Property ProcessName,
                                                                    Path,
                                                                    CreationDate |

```

```

$Services_info = {
    $ComputerName = $txt_ComputerName.Text
    $Services = Get-CimInstance -Class Win32_Service -ComputerName
    $lbl_sysinfo.Text = "Services Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Services | FL -Property Name,
                                                                    DisplayName,
                                                                    Description,
                                                                    StartMode,
                                                                    Started,
                                                                    State,
                                                                    PathName | Out-Null

```

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.ForeColor = "Red"
$lbl_sysinfo.Text = "Please provide a computer name to test"
else {
    $Ping_Test = Test-Connection $ComputerName
    $lbl_sysinfo.Text = "Ping Test Information - $(Get-Date)"
    $lbl_sysinfo.Text += $Ping_Test | Out-String}}

$RDP_Connection = {
    $ComputerName = $txt_ComputerName.Text
    mstsc /v:$ComputerName}

$Export = {
    $ComputerName = $txt_ComputerName.Text

    $TextFile = {
        $ExportOption.Close()

        if ($ComputerName -eq ""){
            $ComputerName = (Get-CimInstance -Class Win32_Compute

        $lbl_sysinfo.Text | Out-File C:\Scripts\$ComputerName.txt

    $Email = {
        if ($ComputerName -eq ""){
            $ComputerName = (Get-CimInstance -Class Win32_ComputerSy

        $lbl_sysinfo.Text | Out-File C:\Scripts\$ComputerName.txt

        $To = @(($txt_Recipients.Text) -split ',')
        $Attachement = "C:\Scripts\$ComputerName.txt"
        $Recipients.Close()

```

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

```
$EmailParameters = @{  
    To = $To  
    Subject = "System Information - $ComputerName"  
    Body = "Please find attached the information that you ha'  
    Attachments = $Attachement  
    UseSsl = $True  
    Port = "587"  
    SmtServer = "smtp.office365.com"  
    Credential = $EmailCredentials  
    From = $From}
```

```
send-mailmessage @EmailParameters}
```

```
$RecipientsDetails = {  
$ExportOption.Close()
```

```
$Recipients = New-Object system.Windows.Forms.Form  
$Recipients.Text = "Recipients"  
$Recipients.Size = New-Object System.Drawing.Size(500,500)  
$Recipients.AutoSize = $False  
$Recipients.AutoScroll = $False  
$Recipients.MinimizeBox = $False  
$Recipients.MaximizeBox = $False  
$Recipients.WindowState = "Normal"  
$Recipients.SizeGripStyle = "Hide"  
$Recipients.ShowInTaskbar = $True  
$Recipients.Opacity = 1  
$Recipients.FormBorderStyle = "Fixed3D"  
$Recipients.StartPosition = "CenterScreen"
```

```
$RecipientsInfo = @"  
Please enter the recipient.
```

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_Recipients = New-Object System.Windows.Forms.Label
$lbl_Recipients.Location = New-Object System.Drawing.Point(0,100)
$lbl_Recipients.Size = New-Object System.Drawing.Size(500,100)
$lbl_Recipients.Text = $RecipientsInfo
$lbl_Recipients.Font = $Font
$Recipients.Controls.Add($lbl_Recipients)

$txt_Recipients = New-Object System.Windows.Forms.TextBox
$txt_Recipients.Location = New-Object System.Drawing.Point(10,100)
$txt_Recipients.Size = New-Object System.Drawing.Size(460,100)
$txt_Recipients.Font = $Font
$Recipients.Controls.Add($txt_Recipients)

$btn_Recipients = New-Object System.Windows.Forms.Button
$btn_Recipients.Location = New-Object System.Drawing.Point(180,100)
$btn_Recipients.Size = New-Object System.Drawing.Size(125,25)
$btn_Recipients.Text = "OK"
$btn_Recipients.Font = $Font
$btn_Recipients.Add_Click($Email)
$Recipients.Controls.Add($btn_Recipients)

$Recipients.ShowDialog()

$ExportOption = New-Object system.Windows.Forms.Form
$ExportOption.Text = "Export Method"
$ExportOption.Size = New-Object System.Drawing.Size(500,130)
$ExportOption.AutoSize = $False
$ExportOption.AutoScroll = $False
$ExportOption.MinimizeBox = $False
$ExportOption.MaximizeBox = $False
$ExportOption.WindowState = "Normal"
$ExportOption.SizeGripStyle = "Hide"

```

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_ExportOption = New-Object System.Windows.Forms.Label
$lbl_ExportOption.Location = New-Object System.Drawing.Point(20, 50)
$lbl_ExportOption.Size = New-Object System.Drawing.Size(500, 25)
$lbl_ExportOption.Text = "Please select how you want to export"
$lbl_ExportOption.Font = $Font
$ExportOption.Controls.Add($lbl_ExportOption)
```

```
$btn_TextFile = New-Object System.Windows.Forms.Button
$btn_TextFile.Location = New-Object System.Drawing.Point(10, 50)
$btn_TextFile.Size = New-Object System.Drawing.Size(230, 25)
$btn_TextFile.Text = "Text File"
$btn_TextFile.Font = $Font
$btn_TextFile.Add_Click($TextFile)
$ExportOption.Controls.Add($btn_TextFile)
```

```
$btn_Email = New-Object System.Windows.Forms.Button
$btn_Email.Location = New-Object System.Drawing.Point(250, 50)
$btn_Email.Size = New-Object System.Drawing.Size(230, 25)
$btn_Email.Text = "Email"
$btn_Email.Font = $Font
$btn_Email.Add_Click($RecipientsDetails)
$ExportOption.Controls.Add($btn_Email)
```

```
$ExportOption.ShowDialog() }
```

```
$NetStat = {
$ComputerName = $txt_ComputerName.Text
```

```
if ($ComputerName -eq ""){
$LocalNetStat = Get-NetTCPConnection
$lbl_sysinfo.Text = "NetStat Information - $(Get-Date)"
```

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.Text += $RemoteNetStat | FT | Out-String }}
```

```
Add-Type -AssemblyName System.Windows.Forms
```

```
$Font = New-Object System.Drawing.Font("Consolas",12,[System.D
```

```
$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"
```

```
$MainForm.ShowInTaskbar = $True
```

```
$MainForm.Font = $Font
```

```
$lbl_ComputerName = New-Object System.Windows.Forms.Label
```

```
$lbl_ComputerName.Location = New-Object System.Drawing.Point(0
```

```
$lbl_ComputerName.Size = New-Object System.Drawing.Size(150,20
```

```
$lbl_ComputerName.Font = $Font
```

```
$lbl_ComputerName.Text = "Computer Name"
```

```
$MainForm.Controls.Add($lbl_ComputerName)
```

```
$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.Font = $Font
```

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.Location = New-Object System.Drawing.Point(1
$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.Font = $Font
$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.Font = $Font
$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.Font = $Font
$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)
```

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_MB = New-Object System.Windows.Forms.Button
$btn_MB.Location = New-Object System.Drawing.Point(5,150)
$btn_MB.Size = New-Object System.Drawing.Size(145,25)
$btn_MB.Font = $Font
$btn_MB.Text = "Motherboard"
$btn_MB.Add_Click($mb_info)
$MainForm.Controls.Add($btn_MB)
```

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

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_Network.Font = $Font
$btn_Network.Text = "Network"
$btn_Network.Add_Click($Network_info)
$MainForm.Controls.Add($btn_Network)
```

```
$btn_NetSettings = New-Object System.Windows.Forms.Button
$btn_NetSettings.Location = New-Object System.Drawing.Point(5,
$btn_NetSettings.Size = New-Object System.Drawing.Size(145,25)
$btn_NetSettings.Font = $Font
$btn_NetSettings.Text = "Net Settings"
$btn_NetSettings.Add_Click($NetSettings_info)
$MainForm.Controls.Add($btn_NetSettings)
```

```
$btn_Monitors = New-Object System.Windows.Forms.Button
$btn_Monitors.Location = New-Object System.Drawing.Point(5,300
$btn_Monitors.Size = New-Object System.Drawing.Size(145,25)
$btn_Monitors.Font = $Font
$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(5,325)
$btn_OS.Size = New-Object System.Drawing.Size(145,25)
$btn_OS.Font = $Font
$btn_OS.Text = "OS"
$btn_OS.Add_Click($OS_info)
$MainForm.Controls.Add($btn_OS)
```

```
$btn_Keyboard = New-Object System.Windows.Forms.Button
$btn_Keyboard.Location = New-Object System.Drawing.Point(5,350
$btn_Keyboard.Size = New-Object System.Drawing.Size(145,25)
```

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

```
$btn_Mouse = New-Object System.Windows.Forms.Button
$btn_Mouse.Location = New-Object System.Drawing.Point(5,375)
$btn_Mouse.Size = New-Object System.Drawing.Size(145,25)
$btn_Mouse.Font = $Font
$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(5,400)
$btn_CDROM.Size = New-Object System.Drawing.Size(145,25)
$btn_CDROM.Font = $Font
$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(5,425)
$btn_Sound.Size = New-Object System.Drawing.Size(145,25)
$btn_Sound.Font = $Font
$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(5,450)
$btn_Printers.Size = New-Object System.Drawing.Size(145,25)
$btn_Printers.Font = $Font
$btn_Printers.Text = "Printers"
$btn_Printers.Add_Click($Printers_info)
$MainForm.Controls.Add($btn_Printers)
```

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_Fan.Font = $Font
$btn_Fan.Text = "Fan"
$btn_Fan.Add_Click($Fan_info)
$MainForm.Controls.Add($btn_Fan)
```

```
$btn_Battery = New-Object System.Windows.Forms.Button
$btn_Battery.Location = New-Object System.Drawing.Point(5,500)
$btn_Battery.Size = New-Object System.Drawing.Size(145,25)
$btn_Battery.Font = $Font
$btn_Battery.Text = "Battery"
$btn_Battery.Add_Click($Battery_info)
$MainForm.Controls.Add($btn_Battery)
```

```
$btn_PortBattery = New-Object System.Windows.Forms.Button
$btn_PortBattery.Location = New-Object System.Drawing.Point(5,
$btn_PortBattery.Size = New-Object System.Drawing.Size(145,25)
$btn_PortBattery.Font = $Font
$btn_PortBattery.Text = "Port Battery"
$btn_PortBattery.Add_Click($PortBattery_info)
$MainForm.Controls.Add($btn_PortBattery)
```

```
$btn_Software = New-Object System.Windows.Forms.Button
$btn_Software.Location = New-Object System.Drawing.Point(5,550)
$btn_Software.Size = New-Object System.Drawing.Size(145,25)
$btn_Software.Font = $Font
$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(5,575)
$btn_Process.Size = New-Object System.Drawing.Size(145,25)
```

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

```
$btn_Services = New-Object System.Windows.Forms.Button
$btn_Services.Location = New-Object System.Drawing.Point(5,600)
$btn_Services.Size = New-Object System.Drawing.Size(145,25)
$btn_Services.Font = $Font
$btn_Services.Text = "Services"
$btn_Services.Add_Click($Services_info)
$MainForm.Controls.Add($btn_Services)
```

```
$btn_Ping = New-Object System.Windows.Forms.Button
$btn_Ping.Location = New-Object System.Drawing.Point(5,625)
$btn_Ping.Size = New-Object System.Drawing.Size(145,25)
$btn_Ping.Font = $Font
$btn_Ping.Text = "Ping Test"
$btn_Ping.Add_Click($Ping_Test_info)
$MainForm.Controls.Add($btn_Ping)
```

```
$btn_NetStat = New-Object System.Windows.Forms.Button
$btn_NetStat.Location = New-Object System.Drawing.Point(5,650)
$btn_NetStat.Size = New-Object System.Drawing.Size(145,25)
$btn_NetStat.Font = $Font
$btn_NetStat.Text = "NetStat"
$btn_NetStat.Add_Click($NetStat)
$MainForm.Controls.Add($btn_NetStat)
```

```
$btn_RDP = New-Object System.Windows.Forms.Button
$btn_RDP.Location = New-Object System.Drawing.Point(5,675)
$btn_RDP.Size = New-Object System.Drawing.Size(145,25)
$btn_RDP.Font = $Font
$btn_RDP.Text = "RDP"
$btn_RDP.Add_Click($RDP_Connection)
$MainForm.Controls.Add($btn_RDP)
```

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

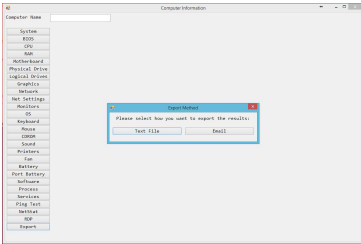
```
$btn_Export.Font = $Font
$btn_Export.Text = "Export"
$btn_Export.Add_Click($Export)
$MainForm.Controls.Add($btn_Export)

$MainForm.ShowDialog()
```

[adinsserter name="Matched-Content"]

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

Summary



Article Name

How to get remote system information - Part 3

Description

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


Author

Stephanos

Publisher Name

Stephanos Constantinou Blog

Publisher Logo



FILED UNDER: POWERSHELL SCRIPTS

WIN32_PHYSICALMEMORY, WIN32_POINTINGDEVICE, WIN32_PORTABLEBATTERY, WIN32_PRINTER,
WIN32_PROCESS, WIN32_PROCESSOR, WIN32_PRODUCT, WIN32_SERVICE, WIN32_SOUNDDEVICE,
WIN32_VIDEOCONTROLLER, WIN32REG_ADDREMOVEPROGRAMS

Comments

[Habib Salim](#) says

[09/05/2018 at 19:43](#)

Hello Stephanos

This is really nice. i just played with it a bit

Please consider adding a part 4.

a script to poll a list of machines (belonging to an AD group)

and tabulating the results.

I'll try my hand at it and post back results if i am successful

[Reply](#)

[Stephanos](#) says

[14/05/2018 at 11:22](#)

Dear Habib,

Thank you for your comments.

I will go though the script again and check what additional features I can add.

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

Trackbacks

How to get remote system information - Part 3 - How to Code .NET says:

14/03/2018 at 19:08

[...] on March 13, 2018by admin submitted by /u/SConstantinou [link] [comments]

No comments [...]

How to get remote system information - Part 1 - Stephanos Constantinou says:

03/04/2018 at 10:23

[...] How to get remote system information – Part 3 – Stephanos Constantinou says:

14/03/2018 at 17:25 [...]

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

says:

14/04/2018 at 13:01

[...] How to get remote system information – Part 3 [...]

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

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

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)

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