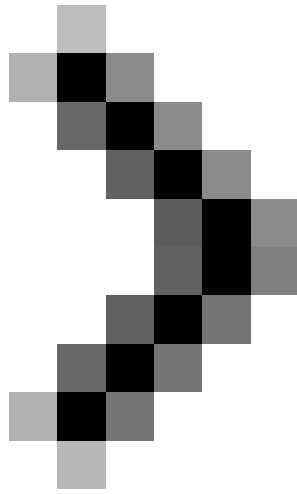


Managing Windows networks using scripts - Part 7: Troubleshooting errors

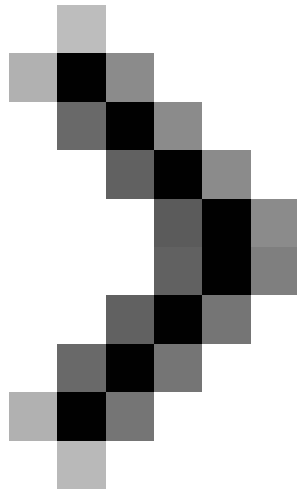
One thing we can do is say 'Please ignore the error'. That is what is said with this method. After all, any real-world administrator knows that IT is not a correct science



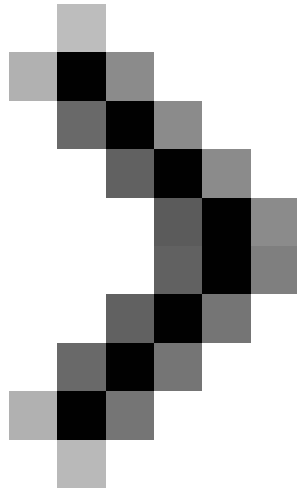
Part 1: Basic concepts



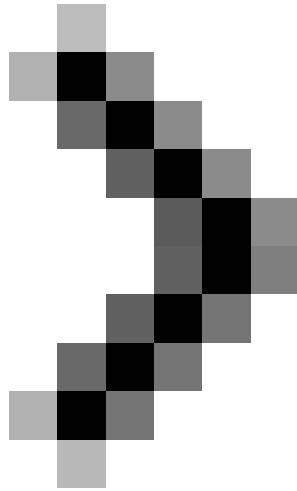
Part 2: Complete the script



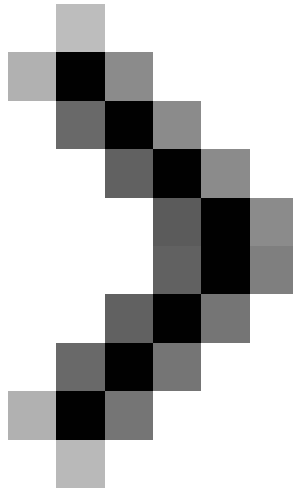
Part 3: Understanding WMI



Part 4: Use Win32_NetworkAdapterConfiguration



Part 5: Overcoming challenges



Part 6: The first steps for remote scripting

Mitch Tulloch

In the previous article, we used the *ChangeIPAddress.vbs* script that was developed earlier and used it to change the IP address on the remote computer. This is what we changed the script :

```
Option Explicit
Dim objWMIService
Dim objNetAdapter
Dim strComputer
Dim strAddress
Dim arrIPAddress
Dim arrSubnetMask
Dim colNetAdapters
Dim errEnableStatic
```

```
If WScript.Arguments.Count = 0 Then
Wscript.Echo "Usage: ChangeIPAddress.vbs new_IP_address"
WScript.Quit
End If
```

```
strComputer = "xp2"
strAddress = Wscript.Arguments.Item (0)
```

```
arrIPAddress = Array (strAddress)
arrSubnetMask = Array ("255.255.255.0")
Set objWMIService = GetObject ("winmgmts:" & strComputer & "rootcimv2")
Set colNetAdapters = objWMIService.ExecQuery ("Select * from Win32_NetworkAdapterConfiguration
where IPEnabled = TRUE")
For Each objNetAdapter in colNetAdapters
errEnableStatic = objNetAdapter.EnableStatic (arrIPAddress, arrSubnetMask)
next
```

Current:

```
strComputer = "xp2"
```

Tell us that the computer is integrated by a script called XP2. The original XP2 remote computer has an IP address of 172.16.11.43.

Now when we run this script by typing **ChangeIPAddress.vbs 172.16.11.65** from a management workstation called XP, what happens is shown below:

1. The script has been active. The address example of XP2 has changed from 172.16.11.43 to 172.16.11.65.
2. The script takes a lot of time to execute
3. The script returns the following error: *C: toolsChangeIPAddress.vbs (23, 6) SWbemObjectEx: The remote procedure call failed .*

How can we solve those results?

Easy solution

One thing we can do is say ' *Please ignore the error* '. That is what is said with this method. After all, any real-world administrator knows that IT is not a correct science in every detail and they often end with applying 'other solutions' to bouncing problems. birth when there are no right solutions for them.

So how can we ignore errors? Add the following line near the beginning of the header (header):

```
On Error Resume Next
```

In other words, our headers will look like this:

```
Option Explicit
On Error Resume Next
Dim objWMIService
.
```

Now we don't see errors, and our script works. However, it still takes a long time to execute, in fact it may take several minutes. So what's going on?

Troubleshooting error message

Error messages are sometimes confusing and this is one of the issues of concern. This is an error message:

SWbemObjectEx: The remote procedure call failed.

And this is the code that generated it:

```
errEnableStatic = objNetAdapter.EnableStatic (arrIPAddress, arrSubnetMask)
```

This line of code works (for example, the IP address is changed on the target computer) but then it gives an error message. Why is that? Let's start by finding out what SwbemObjectEx means. Quick search in MSDN may indicate (<http://msdn2.microsoft.com/en-us/library/aa393259.aspx>):

Extend the functionality of SWbemObject. This object adds the Refresh method for SWbemRefresher objects.

(Expand the functionality of SWbemObject. This object adds the Refresh method to SwbemRefresher objects)

So SwbemObjectEx basically just adds functionality to SWbemObject. So what is SwbemObject?

Contains và x? lý các m?t ??n v? trí WMI ??i t??ng ho?c instance.

(Including a WMI object or an event)

That means what? This page gives us much information but it is not appropriate. In all cases, though, SwbemObject (SWbemObjectEx) is all you manage or query in WMI. In our scenario, we are querying the Win32_NetworkAdapterConfiguration class and returning a collection of objects called colNetAdapter that represent network adapters on the computer. So SWbemObjectEx (or SWbemObject) mentioned in this error message simplifies the object representing the network adapter itself, such as objNetAdapter. So why objNetAdapter generates an error.

Anyway, this seems like a problem. According to one of the reputable experts, it is possible that a component in the hotfix for Windows XP is changed the way the return call is created and submitted when the statement that caused the error is executed. Usually if you call the EnableStatic method of an object that is instantiated with an example of the Win32_NetworkAdapterConfiguration class that is completely successful it will return 0, which means there is no error. If it returns 1, that means rebooting. Obviously with Windows XP there is no need to reboot when you change the IP address on the network adapter. If for some reason, a hotfix can change something in WMI or another component so that Windows does not restart before the new address is replaced on the target machine, this may cause a configuration error. The network adapter image on the computer falls into an unclear state until the computer is restarted. However, when the script is still running on the administrative workstation, when configuring the target computer's network adapter is in an unidentified state, the RPC connection between the two computers is very bad before this scenario ends. . Therefore, an error will appear here.

At least that is the best answer we have for this issue, and we continue to research it thoroughly. However, let's see if we can confirm for some reason that this problem is resolved, for example, that error only relates to the EnableStatic method of Win32_NetworkAdapterConfiguration. What if we try to write another script to do something other than the network adapter on the target computer instead of changing its IP address? For example, how to change the default port instead of the IP address on the target computer? If that is done, at least we can successfully connect from the administrative workstation to the remote computer and call the WMI method to change the network settings on it.

Change the default port

In this section, I recommend that you go back to a bit of part 4 of this series, where I showed you how to use MSDN to learn how to use the properties and methods of Win32_NetworkAdapterConfiguration. We think that by doing so you can write such a script yourself. Let's try it!

PAUSE

When you try to write your own script, sometimes it can work, sometimes not working. If it doesn't work, follow the steps below:

1. First go to the MSDN page of the Win32_NetworkAdapterConfiguration class.
2. Find on this page the method to perform to change the port on the network adapter. A quick check on the page will give you this:

SetGateways - Specifies a list of gates

(SetGateways - a port list for routing packages that has been predetermined for the subnet other than the subnet to which this adapter is connected)

That's exactly what we want, so click SetGateways to open the SetGateways Method page of Win32_NetworkAdapterConfiguration Class.

3. On this SetGateways Method page you will see this explanation:

The SetGateways WMI class method specifies a list of gateways for gói routing ?? m?t subnet là khác nhau t? subnet, các m?ng Adapter ???c k?t n?i ??n. Ch? ch? ?? này này khi thi?t b? Interface Network (NIC) là trong các Static IP mode.

(The SetGateways class method WMI specifies a port list for routing packets to another subnet for the subnet on which the network adapter is connected. This method works only when the Network Interface Card (NIC) is in Static IP mode)

So you have learned that the target computer must have a static address before you call this method. Reading more carefully you can find the syntax for calling this method as follows:

SetGateways (A, B)

Here A is the string variable of the IP address for the port, B is an integer value from 1 to 9999 specifying that parameter. You must now have enough information to write the script. The simplest way is to start with our original *ChangeIPAddress.vbs* script that is included in part 2 and change it a bit until we get a new script like this:

```
'=====
'NAME: ChangeGateway.vbs
',
'AUTHOR: Mitch Tulloch
'DATE: February 2007
',
'ARGUMENTS:
'first. gateway_address
'2. metric
'===== -
```

Option Explicit

```
Dim objWMIService
Dim objNetAdapter
Dim strComputer
Dim strGateway
Dim arrGateway
Dim intMetric
Dim arrMetric
Dim colNetAdapters
Dim errGateway
```

```
'L?u cho thi?u ??i s?
```

```
If WScript.Arguments.Count = 0 Then
Wscript.Echo "Usage: ChangeGateway.vbs gateway_address metric"
WScript.Quit
End If
```

```
strComputer = "xp2"
strGateway = Wscript.Arguments.Item (0)
arrGateway = Array (strGateway)
intMetric = Wscript.Arguments.Item (1)
arrMetric = Array (intMetric)
Set objWMIService = GetObject ("winmgmts:" & strComputer & "rootcimv2")
Set colNetAdapters = objWMIService.ExecQuery ("Select * from Win32_NetworkAdapterConfiguration
where IPEnabled = TRUE")
For Each objNetAdapter in colNetAdapters
errGateway = objNetAdapter.SetGateways (arrGateway, arrMetric)
next
```

```
'Display k?t qu? ho?c mã l?i
```

```
If errGateway = 0 Then
```

```
Wscript.Echo "Adapter's gateway has been successfully changed to" & strGateway
```

```
Else
```

```
Wscript.Echo "Changing the gateway's adapter was not successful. Error code" & errGateway
```

```
End If
```

Copy this script into Notepad (in Word Wrap mode has been dropped) and save it as *ChangeGateway.vbs* . Now check it out. Access the remote machine XP2, open a command **prompt** and enter **ipconfig / all** and the following result:

```
C:> ipconfig / all
```

```
Windows IP Configuration
```

```
Host Name.....: XP2
```

```
Primary Dns Suffix. . . . . : contoso.com
```

```
Node Type. . . . . : Unknown
```

```
IP Routing Enabled. . . . . : No
```

```
WINS Proxy Enabled.....: No
```

```
Ethernet adapter Local Area Connection:
```

```
Connection-specific DNS Suffix.:
```

```
Description . . . . . : Intel 21140-Based PCI Fast Ethernet
```

```
Adapter (Generic)
```

```
Physical Address. . . . . : 00-03-FF-21-88-8C
```

```
Dhcp Enabled. . . . . : No
```

```
IP Address. . . . . : 172.16.11.80
```

```
Subnet Mask. . . . . : 255.255.255.0
```

```
Default Gateway.....: 172.16.11.1
```

```
C:>
```

Now on the XP admin desktop, open a command window and run this new script as follows:

```
C: tools>ChangeGateway.vbs 172.16.11.2 5
```

```
Microsoft (R) Windows Script Host Version 5.6
```

```
Copyright (C) Microsoft Corporation 1996-2001.All rights reserved.
```

```
Máy ?nh adapter ?ã ???c thay ??i ?? 172.16.11.2
```

```
C: tools>
```

This scenario takes about 5 seconds to finish, no errors appear (Note that here we omitted On Error Resume Next from the given header of the script because we want to observe any errors if they appear)

Go back to the remote computer XP2 and run **ipconfig / all** again:

```
C:> ipconfig / all
```

```
Windows IP Configuration
```

```
Host Name.....: XP2
Primary Dns Suffix. . . . . : contoso.com
Node Type. . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled.....: No
```

```
Ethernet adapter Local Area Connection:
```

```
Connection-specific DNS Suffix.:
Description . . . . . : Intel 21140-Based PCI Fast Ethernet
Adapter (Generic)
Physical Address. . . . . : 00-03-FF-21-88-8C
Dhcp Enabled. . . . . : No
IP Address. . . . . : 172.16.11.80
Subnet Mask. . . . . : 255.255.255.0
Default Gateway.....: 172.16.11.2
```

```
C:>
```

It worked. No errors occurred - we ran a remote script against another computer and it changed the default port of the target computer.

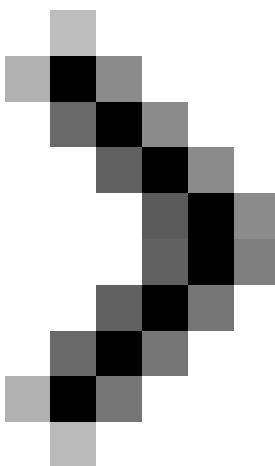
Can we check if the measurement parameter has been changed? The **ipconfig** command does not display this information, but we can use **netsh** to get this as follows:

```
C:> netsh interface ip show address
```

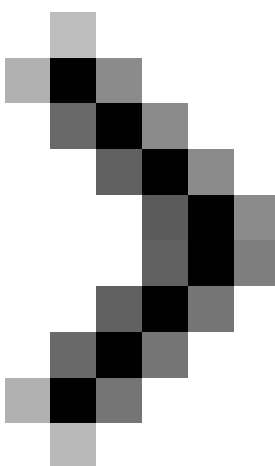
```
Configuration for interface "Local Area Connection"
DHCP enabled: No
IP Address: 172.16.11.80
SubnetMask: 255.255.255.0
Default Gateway: 172.16.11.2
GatewayMetric: 5
InterfaceMetric: 0
```

```
C:>
```

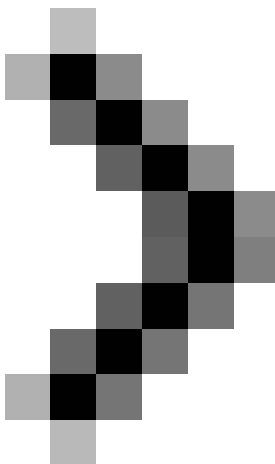
This is the result!



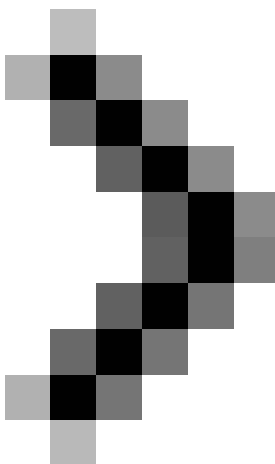
Part 8: Remote script error handling with Network Monitor 3.0



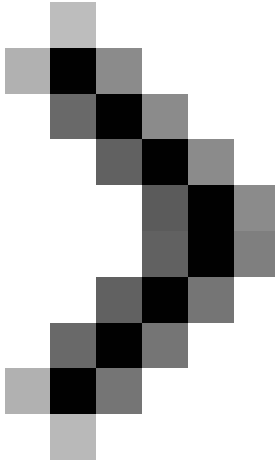
Part 9: Understanding remote control scenarios



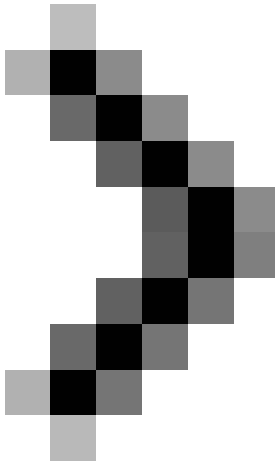
Part 10: Tricks of remote control scenarios



Part 11: Other script tricks



Part 12: Properties of the WMI class



Part 13: The script returns all values

You finished reading the article "**Managing Windows networks using scripts - Part 7: Troubleshooting errors**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
