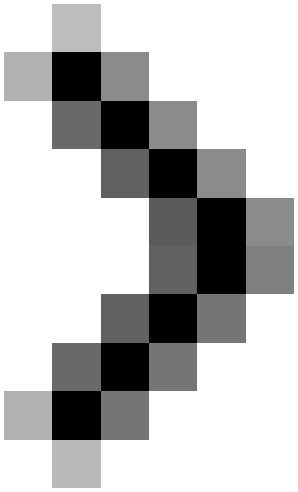
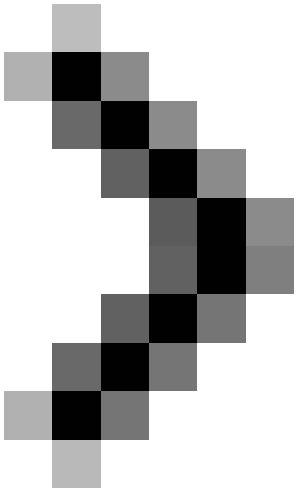
Managing Windows networks using Script - Part 9: Understanding remote scripting

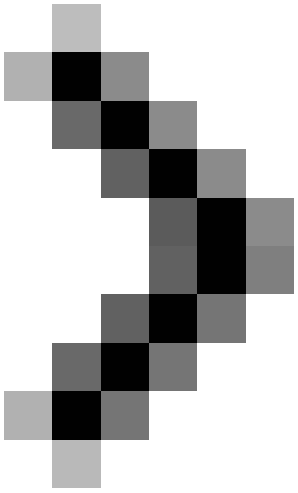
Now you need to go back and review the techniques of remote control script in detail before we go further in this regard. One good way for you is to jump in and try everything, but this way sometimes brings you to the wall. To avoid encountering this wall we have to find it



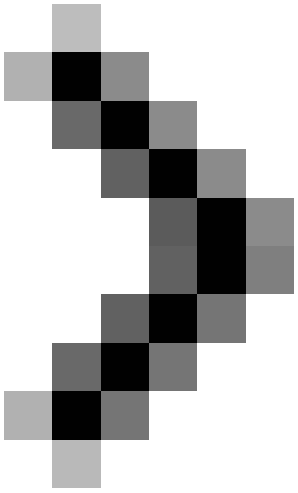
Part 1: Basic concepts



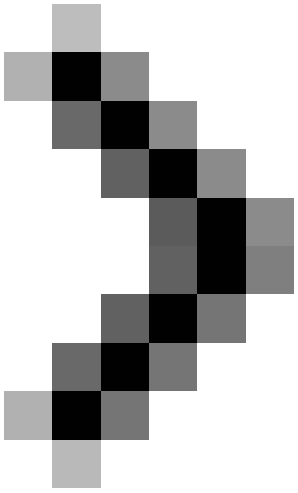
Part 2: Complete the script



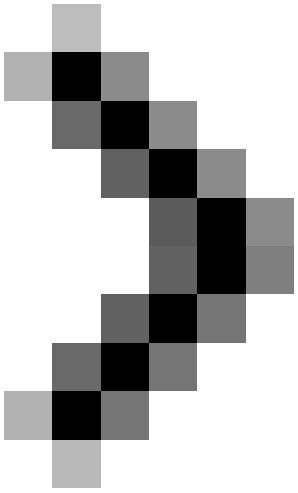
Part 3: Understanding WMI



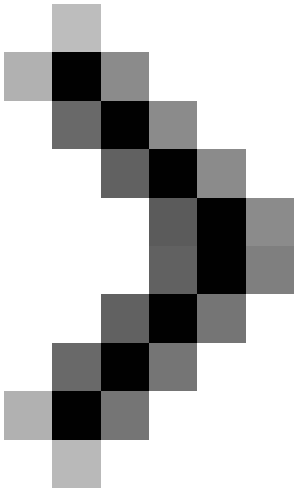
Part 4: Use Win32_NetworkAdapterConfiguration



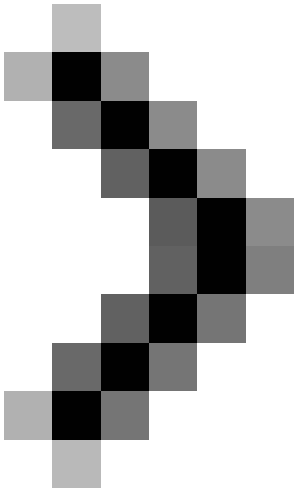
Part 5: Overcoming challenges



Part 6: The first steps for remote scripting



Part 7: Troubleshooting errors



Part 8: Remote script error handling with Network Monitor 3.0

Mitch Tulloch

Now you need to go back and review the techniques of remote control script in detail before we go further in this regard. One good way for you is to jump in and try everything, but this way sometimes brings you to the wall. To avoid encountering this wall we must learn about their foundations.

Two types of remote control scenarios

There are two types of remote scripting. The first is when we run a script on computer A and the target computer is B to perform some action on it. In our test using the ChangeIPAddress.vbs script, we changed the line:

```
strComputer = "."
```

into:

```
strComputer = "xp2"
```

If we use the first line above and run the script on computer A, it will change the IP address of this computer. If we use the second line above and run the script on computer A, we can change the IP address of computer B.

The second type of remote control script and it works the same way. I am an administrator, logged in to

computer A and have a script I want to use to perform some work on computer B. However, instead of trying to run the script on computer A and the target As computer B, I want to run the script directly on computer B. That's why I brought the script from computer A to the target computer B and then ran it here. How can I do that? If there is an Active Directory environment then I can try and run the script as the login script on the remote computer. Let's see how to do it in the next article, but now notice that there are two types of remote scripting.

1. Run the script on the local computer and target the remote computer.
2. Run the script directly on the remote computer.

Describe the difference between the two ways to describe the remote script:

1. The first type involves connecting to the remote computer and then running the script.
2. The good type involves deploying the script for the remote computer, then running the script.

Learn about connecting remote control scenarios

Now let's focus on the first type of remote control scenario. What does it mean to run a script on your local computer to connect to a remote computer and run it back? It has 3 meanings:

1. Network connections
2. User identification
3. Allow appropriate

1. Network connection

For the script to do something on the remote computer, it must first establish a network connection with the remote computer. What problems can prevent your network connection?

First, it may be a name problem, if your script cannot resolve the computer's hostname or FQDN into its IP address, the script may be corrupted.

Second, it could be a firewall problem. We have seen in the previous article that in order to get our WMI script to run for a remote computer, we must open the Remote Administration exception for the Remote Administration in the Windows firewall on the computer. remote Now if you open Windows Firewall applet from Control Panel and select the Exceptions tab, you will not see the Remote Administration labeled checkbox that you can choose to open this exception. The reason for this is that this Control Panel applet is meant primarily for home users to use to configure their firewalls. In an enterprise environment where Active Directory is used, the way to manage the preferred Windows Firewall is to use Group Policy. We saw in the previous article that setting up Group Policy we need to configure is as follows:

Computer Configuration Administrative TemplatesNetworkNetworkConnectionsWindows FirewallDomain ProfileWindows Firewall: Dho allows remote administration exceptions to return.

When you target this policy for a remote computer, it will open two TCP ports on that computer: ports 445 and 135.

• TCP port 445 is the port for traffic going into the Server Message Block (SMB), if this port is locked on the remote computer's firewall, you can not only connect to it with WMI, but you can also not connect connect to it

with standard MMC administration tools like Computer Management. When the port is locked and you are trying to run scripts on the remote computer, there may be some confusing errors like 'System error 53 has occurred. The network path was not found '- System error 53 appears. Network path not found .

• TCP port 135 is the port for traffic into Distributed COM (DCOM). More specifically, port 135 is the listening port for DCOM Service Control Manager (SCM), which provides RPC services for instantiating COM objects.

Its length or short length are both TCP 135 and 445 ports that need to be opened on the remote computer's firewall if the WMI queries run from the local computer to the successful use of RCP to connect the WMI service on the machine. Remote control and successful demonstration of DCOM objects on the remote computer.

2. User identification

When you run the script for a remote computer and can establish a network connection with the remote computer, then the script can perform actions on that remote computer. But the actions it can take depend on the identity with which script is running on the control computer. For example, I log in to computer A by using a regular domain user account. Then I run the ChangeIPAddress.vbs script and target it to the remote computer B. The script uses RPC to connect to the WMI service on computer B and it changes the IP address of the B computer. city. Why? Who is trying to perform this action on a remote computer? On the local computer (computer A) you are the user and when you run the script by default it represents your identity, meaning that the script will perform its actions with your identity. (your user account). So the script will change the remote computer's IP address, it works for you, a domain user, who is doing this. Otherwise, it will fail when the change requires local administrator credentials.

So when you are sitting at computer A, you are logged in to the domain user and you still want to use your script to change the computer's IP address. You can do them as follows:

Your ChangeIPAddress.vbs script may change as follows:

```
Set objWMIService = GetObject ("winmgmts:" & strComputer & "rootcimv2")
```

with

```
strUser = "Administrator"
strPassword = 'Pa $$ w0rd'
Set objWMIService = GetObject ("winmgmts:" & strComputer & "rootcimv2", strUser, strPassword)
```

The problem here is insecure - the administrator account's password for the remote computer is in the script's text and can be observed.

So how can we remove these first two lines and hide the values ??of strUser and strPassword for the script as arguments when the script is run? Better still is hardening these values ??in the script, but if someone has a running program (like Network Monitor 3.0) then they can get important information and then you compromise the machine. my remote

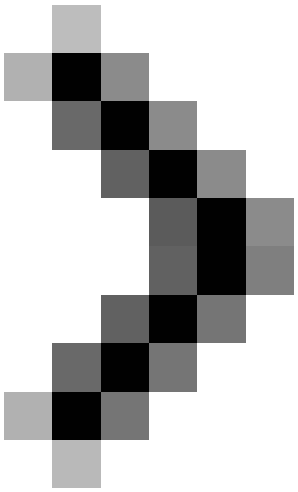
What if you use an advanced command like runas / user: Administrator cmd.exe and then run the script from the

advanced command window without specifying other important information? That might be the best solution for remote scripting, where you want to make sure the script has the proper identity (usually local administration on the target computer) even though it's quite complicated. Obviously, you can simply log into a workstation as a domain administrator account and simply open a command and run the script.

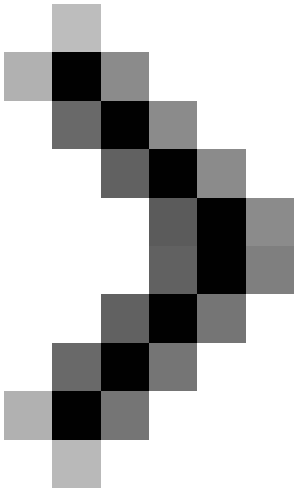
3. Appropriate permission

You are running the script on computer A and the script is assumed to perform some actions on computer B. The script has established a network connection with the WMI service on computer B and is trying to execute the Its action by using the correct identity (usually local administrator information) on computer B. What can cause this script to crash here? Not enough permission! If the script is trying to perform some ACL-controlled actions (such as changing a file object or creating an object in Active Directory or activating a DCOM object) and you don't have permission to like In order to perform that action, the script will fail. Unfortunately, it is often the hardest part of remote scripting with NTFS, DCOM permissions and many other Windows-based permissions. You may have the right permissions but not the right permissions, ie user rights to perform some actions. For example, saying that you want to use a script to delete a logon event on a remote computer, but your identity lacks SeSecurityPrivilege security rights on that remote computer, then your script will fail. .

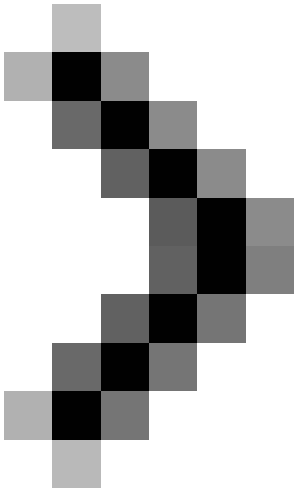
There is a lot to learn about remote scripting. We will continue to introduce you in the next article.



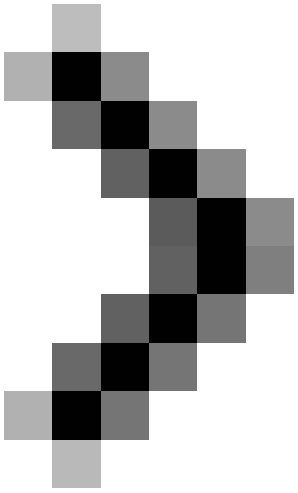
Part 10: Tricks of remote control scenarios



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 Script - Part 9: Understanding remote scripting**" 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.
