

Lab 7 Implicit Remoting

In this lab you will learn:

* How to create a temporary implicit remoting session
* How to create a persistent implicit remoting session

Administrators must often access remote systems from workstations that do not have the appropriate cmdlet modules loaded on the computer. The Active Directorymodule is a good example. It would be crazy to install this module on every workstation, just in case the administrator might use that computer at some point. PowerShell has a built in method to solve this problem called “implicit remoting”. From the workstation you make a persistent PSsession using a variable to the server. Then you import the module into the session variable on the server. The module is on the server, and only “proxy commands” are created on the workstation. These proxy commands can only request that the required cmdlet run on the server. Everything seems local, but it is not. You can only have one implicit remoting session to the server and the commands will only run on the server. Let’s see what modules are installed.

On your SRV1-AD VM, let’s find out which modules are available to you. The following command will display these modules.

**Get-module –listavailable**

Scroll down and you should see a module called ActiveDirectory

Now, go to WIN7-WS VM and run the same command. The ActiveDirectory module should not be available because WIN&-WS is not a domain controller. Type the following.

**Get-Adcomputer**

**Question:** What output did you get? \_\_**get-adcomputer : The term 'get-adcomputer' is not recognized as the name of a cmdlet** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now we will proceed to use the AD module remotely.

1. Temporary Implicit Remoting Session
2. From WIN7-WS create a persistent PSsession to the SRV1-AD using a variable. Remember when connecting to a DC from a workstation, the domain admin account must be used.

# Create a PowerShell remote session to a server with the #cmdlets installed.

**$s1\_DC = New-PSsession -Computername SRV1-AD -credential <domain>\administrator**

1. Issue a command to import the AD module to the session you are working in. You are not doing a file transfer to WIN7-WS. Import-Module only works on the local computer; therefore the invoke command is telling the server to local the module and since we are using the session variable, the module cmdlets will be loaded into it.

# Load the module into the session variable on the server  
#Notice we do not use the credential parameter because the credentials built into $s1\_DC

**Invoke-Command -scriptblock {Import-Module ActiveDirectory} -Session $s1\_DC**

1. Once the cmdlets are added to the session variable on the server, we can create an implicit remoting session to the server. The process sends “proxy commands” to WIN7-WS which can only request that a command run on the server you are connected to. To create the implicit remoting session we use the Import-PSsession cmdlet

# add the available cmdlets to your existing session via proxy commands

#**NOTE:** It is a good idea to add a prefix to distinguish the remote commands which appear to be running locally, from any local cmdlets which may have the same name. The –prefix will add a string to the noun portion of the cmdlet. So, Get-Adcomputer can be changed to Get-RemoteAdcomputer.

**Import-PSSession -Session $s1\_DC -Module ActiveDirectory -Prefix Remote**

What was the title of the progress bar? \_\_**Creating implicit remoting module**\_\_\_\_\_\_\_\_\_\_\_\_\_

Now you can test to see if the cmdlets are now available and active

**Get-RemoteADcomputer** - you must use the prefix to ensure you are using the correct cmdlet

Take a screenshot of your output and name the file **LearnName\_L7\_TempImplicit.jpeg**

**Question1**: What was the output? \_\_\_\_\_\_\_**All the computers in the AD**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

If all went well. **Close the session. Run the command again.**

**Question2:** What was the output? \_\_\_\_ **Get-RemoteADComputer : The term 'Get-RemoteADComputer' is not recognized as the name of a cmdlet**

Now close then open PowerShell console and run the command again.

**Question3:** What do the answers to questions 2 and 3 tell you about running PowerShell Implicit commands? \_\_**When user close the session, the command is no longer running on the remote server**

With implicit remoting the module is loaded only into the current session of WIN7-WS if the console is closed or the session object destroyed, then the commands can not be found.

Type: **Remove-PSsession \*** -to remove all sessions from WIN7-WS

1. Persistent Implicit Remoting

If a workstation is going to be used regularly to communicate with a server, then you want to use a persistent implicit remoting session which stores the proxy commands in a module folder on the workstation. To use the module commands you only have to import the commands into the current session using Import-Module.

1. A persistent session starts by creating a persistent PSsession (CIMsessions can also be used. The advantage of CIMsessions is that the computers do not need PowerShell or the Windows operating system installed) and saving the session information to a variable.

**$s1\_DC = New-PSsession -computername SRV1-AD -credential <domain>\administrator**

1. Next, we use the invoke-command to load the ActiveDirectory module into the session object (remember: the session object is always stored on the remote computer)

**Invoke-command -session $s1\_DC -scriptblock {import-module ActiveDirectory}**

1. After loading the module into the session object we use the Export-PSsession command to download the proxy commands to the local system. We are also going to add the parameters commandname to filter only the commands we want and the outputmodule parameter to create a module folder name.

**Export-PSsession -session $s1\_DC -commandname \*AD\* -outputmodule AD\_SRV1**

**Question4:** What is thenature of the warning message? \_\_\_\_

**WARNING: Proxy creation has been skipped for the following command: 'Add-Computer, Add-Content, Add-History,**

**Add-Member, Add-PSSnapin, Add-Type, Read-Host', because it would shadow an existing local command. Use the**

**AllowClobber parameter if you want to shadow existing local commands.**

By default, when importing commands implicitly, PowerShell will not overwrite any existing commands on WIN7-WS that have the same name. This is important to remember if the server module is updated and you wish to overwrite commands on the workstation. You must use the allowclobber and force parameters. Here, we are filtering to get only the commands used for Active Directory with AD in the noun. We are also calling the module folder AD\_SRV1 to indicate what the folder is about. Retype the command below.

**Export-PSsession -session $s1\_DC -commandname \*AD\* `**

**-outputmodule AD\_SRV1 -AllowClobber -Force**

1. At this point it is important to remove the session object avoid any potential naming conflict. The commands are on the local computer you don’t need the object anymore.

**Remove-PSsession -session $s1\_DC**

1. Lastly, you import the AD commands into the current session by using the Import-Module cmdlet. Again, it is a good idea to add a prefix to the command to separate those AD commands which are being using implicitly, from any command in the current session with the same name.

**Import-Module AD\_SRV1 -prefix Remote**

1. Type the command

**Get-RemoteADcomputer -filter \***

The command should have worked. Navigate to the WindowsPowerShell\modules folder on WIN7-WS.  
Type the command **gci -recurse**Take a screen shot of the output. Name the file **LearnName\_L7\_PersistentImplicit.jpeg**

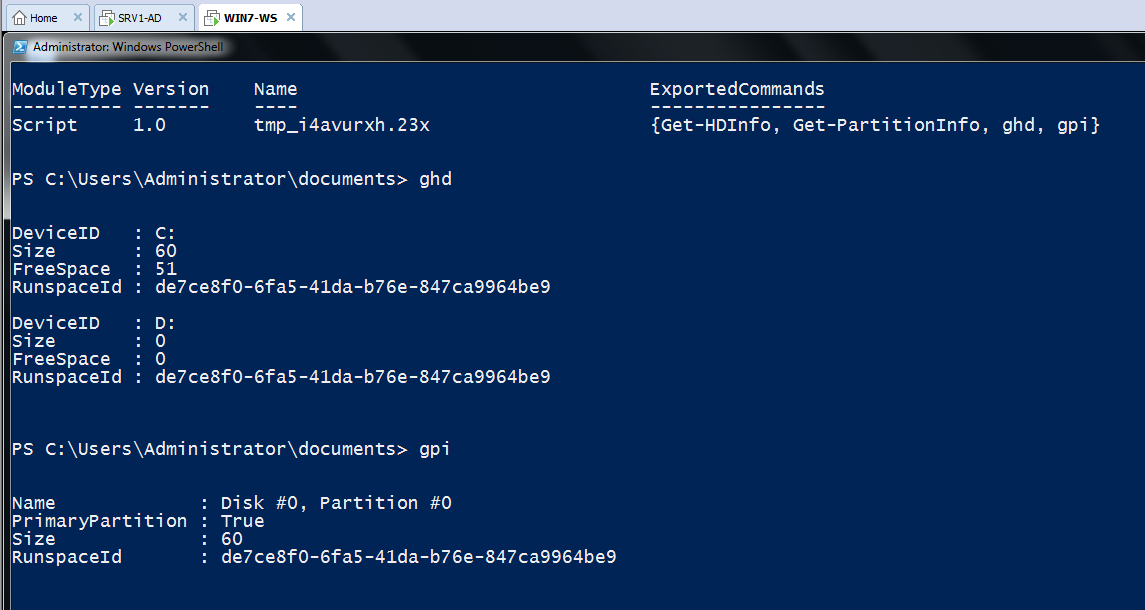
**Exercises:**

In lecture you were told that the DiskInfo module on SRV1-AD would not work implicitly because it made connections to other computers on the domain. In this exercise you will make modifications to the DiskInfo module so that all functions will work correctly, gathering disk and partition information about SRV1-AD. Follow the general guidelines below:

**Note:** Implicit Remoting Sessions do not read the end of line character like running a script locally. I kept getting an error “null pipes are not allowed”, even though the code was correct. I removed all end of line characters and my module ran correctly.

**Exercise 1:**

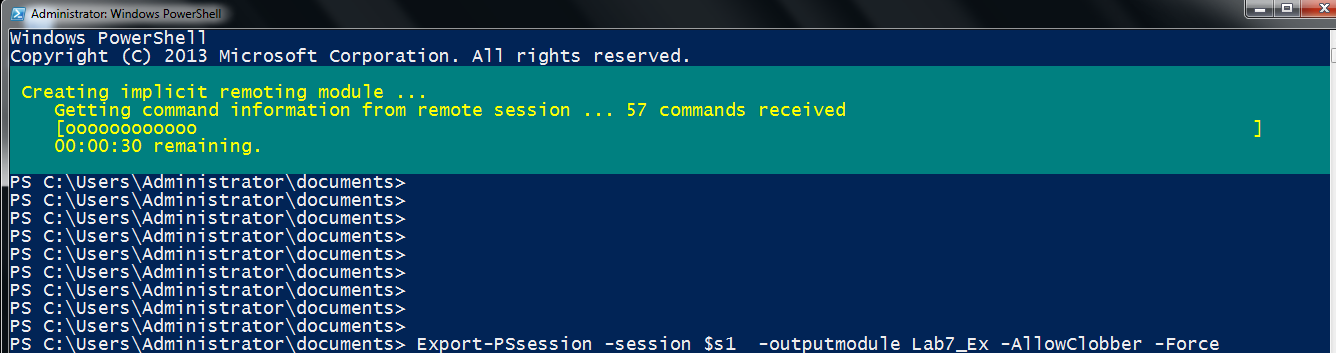
1. Start by copying all code from DiskInfo module into a new ISE tab. Save the new file as L7\_Ex (make sure the module folder name is the same as the file name. Create it manually)
2. Delete the Main Function and the Show-Output function
3. Design Get-HDInfo function to display the results of the command to the console. Do not store in variable.
4. Same for Get-PartitionInfo
5. Manually import the Lab7\_Ex module on SRV1-AD
6. **Create a temporary implicit remoting session** and run the command from WIN7-WS with aliases. Take a screen shot of your output and name the file **LearnName\_L7\_Ex1.jpeg. Make sure you screen shot shows the temporary session name and the name of the VM.** Your output should similar to the screen shot below:



Once you know your module is working proceed to Exercise 2:

**Exercise 2:**

1. Remove all PSsessions on SRV1-AD and WIN7-WS
2. Use your module again to create a persistent implicit remoting session. Name the module folder Lab7\_Ex. Take a screen shot like the one below showing the persistent implicit remoting session being created. **Make sure you show both the progress bar and the command line.** Name the file **LearnName\_L7\_Ex2.jpeg**.



**Grading**

* **LearnName\_L7\_TempImplicit.jpeg**
* **LearnName\_L7\_PersistentImplicit.jpeg**
* **LearnName\_L7\_Ex1.jpeg**
* **LearnName\_L7\_Ex2.jpeg**
* **Lab7\_Ex.psm1**
* **Win500\_Lab7\_ImplicitRemoting.docx**

Zip the files together and submit using the link on My Seneca under Graded Work.