

# Scripting with PowerShell

To make configuration changes on a switch, you can log into each switch and type the commands. If you need to do this to a lot of switches, it will take a while. For this task, we are going to write a script and run it in Windows PowerShell.

To connect to the switch via SSH, you need to install an SSH module. Windows 10 comes with an OpenSSH Client (Beta), but it does not support connections to ICX switches. We are going to use a different module called Posh-SSH.

## **Install Posh-SSH**

Open a PowerShell prompt as administrator.

Run the following command to install the module:

```
Install-Module -Name Posh-SSH -RequiredVersion 2.0.1
```

Now you need to import the module with the following command:

```
Import-Module -Name posh-ssh
```

**Note:** *If you have issues importing the module, you may need to adjust your Execution Policy. To do this perform the following:*

See what the current Execution Policy is set to:

```
Get-ExecutionPolicy
```

Since I ran my PS Console as an Administrator, I changed my Policy from restricted to remote signed for the current user (administrator). This can be a security risk, so obey all security policies you may be subject to.

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

## **Connecting to the host:**

To connect to a host, you will need to open a new Posh-SSH session.

```
New-SSHSession -ComputerName 10.10.10.1 -Credential (Get-Credential) -Verbose
```

You can add the username into the Get Credentials: (Get-Credential user1) Where user1 is the username you would login with.

### **Building the stream to accept commands:**

Now that you are connected, you must build the stream to accept commands.

```
$SSHStream = New-SSHShellStream -Index 0
```

### **Entering commands:**

Now that the stream is built, you can enter commands.

**Note:** for commands that have the page break (i.e. show commands), you need to turn off page display. This command is skip-page-display.

```
$SSHStream.WriteLine("skip-page-display")
```

```
$SSHStream.WriteLine("show interface brief")
```

Now you can read your output:

```
$SSHStream.Read()
```

### **Ending the session:**

```
Remove-SSHSession -Index 0
```

\*\*\*\*\* Continued Below \*\*\*\*\*

## Building a script to execute:


To build a script to execute, we are going to use the above principles. Below is a sample script to log into one switch and change the VLAN for port 1/1/13 to untagged VLAN 40.

```
# ***** Begin Sample Script *****
# ICX Switch Script with Posh-SSH
#Open the SSH session
New-SSHSession -ComputerName 10.10.10.1 -Credential (Get-Credential) -AcceptKey -Force -Verbose
$SSHStream = New-SSHShellStream -Index 0
#Break to establish session
sleep 10
# Write commands to the switch
$SSHStream.WriteLine("config t")
$SSHStream.WriteLine("vlan 40")
$SSHStream.WriteLine("untagged e 1/1/13")
sleep 2
# End SSH session
Remove-SSHSession -Index 0

# ***** End Sample Script *****
```

**Note:** The sleep commands pause the session to allow time for commands to run completely.

Copy this script into a text editor and save it as switch.ps1. The .ps1 extension is for PowerShell. Open your PowerShell prompt as administrator. Change directories to the location you saved the script and execute it.

 Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd C:\Users\dellis\Desktop\Scripts
PS C:\Users\dellis\Desktop\Scripts>
PS C:\Users\dellis\Desktop\Scripts>
PS C:\Users\dellis\Desktop\Scripts>
PS C:\Users\dellis\Desktop\Scripts> .\switch-1.ps1
```

Use `cd C:\<FilePath>` to change directories (Where <FilePath> is the path to the saved file)

Use `.\<filename>` to execute the program (Where <filename> is the name of the script to run)

### Adding credentials to the script for automation:

To add credentials to the script so the user do not have to interact with the script use the below code.

```
# Enter user credentials into the script
$User = "user1"
$PWord = ConvertTo-SecureString -String "P@55word1!" -AsPlainText -Force
$Credential = New-Object -TypeName "System.Management.Automation.PSCredential" -ArgumentList
$User, $PWord
```

**Note:** The password will be passed to the switch in plain text.

### Scripting for multiple hosts:

To execute a script for multiple hosts, we need to get the IP Addresses to run against. To do this, we will create a text file called switches.txt with an IP Address on each line to access. Then we will run this through a FOR loop until we run out of IP Addresses. Below is a sample script.

```
# Read hosts into the variable to perform on multiple hosts
foreach($switchip in Get-Content .\switches.txt)
{
    Lines of Code to Execute
    Lines of Code to Execute
}
```

### Sending output to a file:

To send output from a command to a file we will use the following command.

```
Out-File -FilePath .\output.txt -InputObject $SSHStream.Read()
```

The InputObject used here is the \$SSHStream we have used throughout this document and we used the Read() to capture the output that was derived from running the commands.