# Week 14 PowerShell Computer Information

Contents

[Week 13 PowerShell Computer Information 1](#_Toc171054604)

[Get Computer Information 1](#_Toc171054605)

[Get-DnsClient and Get-DnsClientServerAddress 2](#_Toc171054606)

[Test-NetConnection 3](#_Toc171054607)

[Resolve-DnsName 4](#_Toc171054608)

[Get-NetIPAdress 4](#_Toc171054609)

[Get Network Adapter Information 5](#_Toc171054610)

[Get Active Connections 6](#_Toc171054611)

[Assignment Submission 7](#_Toc171054612)

Time required: 30 minutes

* Comment each line of code as shown in the tutorials and other code examples.
* Follow all directions carefully and accurately.
* Think of the directions as minimum requirements.

# Get Computer Information

Run each of these commands at an Administrative PowerShell prompt on your local computer.

**Get-ComputerInfo** is a PowerShell cmdlet used to retrieve various system information and configuration details on a Windows computer. It provides data about the operating system, hardware components, and software settings.

**Insert a screenshot:**

Click or tap here to enter text.

**Get-Date** returns the current date and time on your system. It provides information such as the year, month, day, hour, minute, and second.

**Insert a screenshot:**

Click or tap here to enter text.

**Get-TimeZone** is used to fetch details about the time zone currently set on your system, including the time zone's name and its UTC offset.

**Insert a screenshot:**

Click or tap here to enter text.

# Get-DnsClient and Get-DnsClientServerAddress

**Get-DnsClient** allows you to retrieve and display DNS (Domain Name System) client configuration information on a Windows computer. DNS is a crucial service that translates human-readable domain names into IP addresses, enabling computers to locate and communicate with each other on a network or the internet.

The **Get-DnsClient** cmdlet provides information related to the DNS client settings on your system. This information typically includes details about the DNS servers your computer is configured to use, whether the DNS server addresses are obtained dynamically or set manually, and other relevant DNS client configuration parameters.

In an IT or network administration context, **Get-DnsClient** is useful for verifying and managing DNS client settings, diagnosing network issues, and ensuring proper DNS resolution. It's an essential tool for maintaining network connectivity and resolving domain names to IP addresses accurately.

**Insert a screenshot:**

Click or tap here to enter text.

**Get-DnsClientServerAddress** is used to retrieve the DNS (Domain Name System) server addresses configured on a Windows computer. DNS servers are responsible for translating human-readable domain names into IP addresses, allowing devices to locate and communicate with each other on a network or the internet.

When you use the `Get-DnsClientServerAddress` cmdlet, it returns information about the DNS server addresses currently set on your system. This information typically includes details about primary and secondary DNS server IP addresses. You can determine whether the DNS server addresses are obtained dynamically from a DHCP server or set manually.

In an IT or network administration context, `Get-DnsClientServerAddress` is useful for verifying the DNS server settings, diagnosing DNS-related issues, and ensuring that the DNS resolution process is correctly configured. It's a valuable tool for maintaining network connectivity and ensuring that DNS requests are handled correctly.

**Insert a screenshot:**

Click or tap here to enter text.

# Test-NetConnection

Test-NetConnection is the PowerShell equivalent of the traditional "ping" command. It allows you to test network connectivity to a specific host or IP address and provides more detailed information about the connection. Here's how you can use it:

**Test-NetConnection**: When you use this cmdlet without specifying a target, it will prompt you to enter the target host or IP address interactively. You can then test the network connection to the provided target, and it will display information about the connection status, latency, and whether it can reach the target successfully.

**Insert a screenshot:**

Click or tap here to enter text.

**Test-NetConnection google.com** : This specific usage tests the network connection to the host "google.com." It will provide detailed information about whether your computer can reach Google's servers, the response time (latency), and various other network-related details.

**Insert a screenshot:**

Click or tap here to enter text.

**Test-NetConnection google.com -TraceRoute** : is a PowerShell command that combines the `Test-NetConnection` cmdlet with the `-TraceRoute` parameter to perform a trace route (traceroute) to the host "google.com."

**-TraceRoute** : This parameter instructs `Test-NetConnection` to perform a traceroute to the specified host. Traceroute is a network diagnostic tool that traces the route a packet takes through the network, showing all the intermediate routers or hops between your computer and the target host. It helps you visualize the path that network traffic follows, which can be useful for diagnosing network issues and understanding the network topology.

By combining `Test-NetConnection` with the `-TraceRoute` parameter and specifying "google.com" as the target, you can perform a traceroute to Google's servers and see the sequence of network hops along the way. This can be valuable for troubleshooting network problems and understanding the network path to a specific destination.

**Insert a screenshot:**

Click or tap here to enter text.

# Resolve-DnsName

**Resolve-DnsName microsoft.com** is a PowerShell command that uses the `Resolve-DnsName` cmdlet to resolve the DNS (Domain Name System) information for the domain "microsoft.com." Here's what this command does:

**Resolve-DnsName** : This cmdlet is used to query DNS records for a specified domain name. When you run it with "microsoft.com" as the argument, it will retrieve various DNS information for the "microsoft.com" domain.

The information you might receive includes:

1. HostName: The primary domain name being resolved, in this case, "microsoft.com."
2. QueryType: The type of DNS record being queried, which can include A (IPv4 address), AAAA (IPv6 address), MX (mail exchange), TXT (text records), and others.
3. QueryResult: The actual result of the DNS query, which could be an IP address, mail server information, or other relevant DNS data.

By using `Resolve-DnsName` in PowerShell, you can obtain DNS information for a specific domain, which is useful for network troubleshooting, verifying DNS configurations, and understanding how a particular domain is set up in the DNS system.

**Insert a screenshot:**

Click or tap here to enter text.

# Get-NetIPAdress

**Get-NetIPAddress** is used to retrieve information about IP (Internet Protocol) addresses associated with network interfaces on a Windows computer. This cmdlet provides details about both IPv4 and IPv6 addresses. Here's what it does:

1. When you run `Get-NetIPAddress` without any parameters, it lists all IP addresses configured on all network interfaces of your computer. This includes IP addresses, their associated network interface, the address type (IPv4 or IPv6), and other related information.
2. You can use various parameters with `Get-NetIPAddress` to filter and refine the results. For example, you can specify a particular network interface, a specific address family (IPv4 or IPv6), or other criteria to focus on specific IP addresses of interest.

In an IT or network administration context, `Get-NetIPAddress` is a valuable tool for managing and troubleshooting network configurations. It allows you to view and verify the IP addresses assigned to your computer's network interfaces, which is essential for tasks such as setting up network connectivity, diagnosing network issues, and ensuring proper network configuration.

**Insert a screenshot:**

Click or tap here to enter text.

# Get Network Adapter Information

**Get-NetAdapter** is used to retrieve information about network adapters (NICs - Network Interface Cards) on a Windows computer.

* It provides details about the physical and virtual network interfaces present on the system.
* This includes the interface name, the MAC address, the status (whether it's enabled or disabled), and other properties.
* It is valuable for network administrators to view and manage network adapters, especially when configuring network settings.

**Insert a screenshot:**

Click or tap here to enter text.

**Get-NetAdapterStatistics** provides statistics related to network adapter performance.

* It offers data on various network-related statistics such as bytes sent and received, packets sent and received, and error statistics.
* This cmdlet is useful for monitoring and diagnosing network traffic and performance issues on specific network adapters.

**sInsert a screenshot:**

Click or tap here to enter text.

**Get-NetAdapterAdvancedProperty** is used to access and view advanced properties of network adapters.

* It allows you to retrieve additional settings and properties of network adapters that may not be visible through basic network configuration.
* This cmdlet is handy for advanced network configuration tasks, especially when you need to access and modify specific settings of a network adapter.

**Insert a screenshot:**

Click or tap here to enter text.

# Get Active Connections

**Get-NetTCPConnection** is used to retrieve information about active TCP (Transmission Control Protocol) connections on a Windows computer.

* It provides details about established and listening TCP connections, including local and remote IP addresses, port numbers, the state of the connection, and process information.
* This cmdlet is helpful for network administrators and IT professionals when they need to monitor and troubleshoot network connections, identify the processes using specific ports, and manage TCP connections on the system.

**Insert a screenshot:**

Click or tap here to enter text.

**Get-NetUDPEndpoint** allows you to obtain information about active UDP (User Datagram Protocol) endpoints on a Windows computer.

* It provides details about open UDP ports and the processes associated with them, along with the local and remote IP addresses.
* This cmdlet is useful for monitoring and diagnosing UDP-based network services and applications, helping you identify which processes are listening on specific UDP ports.

**Insert a screenshot:**

Click or tap here to enter text.

## Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.