40555A Networking Fundamentals

Module 8: Troubleshooting networks in Windows

Contents

 © 2019 Microsoft Corporation. All rights reserved.

Microsoft

Learning objectives based on MTA exam objectives

#	Lesson title	Learning objectives	Exam objectives mapped
1	Overview of configuration and troubleshooting tools	• Use the settings app to configure network settings • Use Network and Sharing Center to configure network settings • Use command-line tools to configure networking • Use Windows PowerShell cmdlets to configure networking	3.6.1 Tools 3.6.2 Tracert 3.6.3 Pathping 3.6.5 Ipconfig 3.6.6 Netstat
2	Troubleshooting basic connectivity	• Describe a troubleshooting approach to networking problems • Verify Internet Protocol (IP) configuration • Test connectivity • Determine network service and host status	3.6.1 Tools 3.6.4 Telnet
3	Troubleshooting name resolution	• Investigate a name resolution problem	3.6.1 Tools 3.4.4 Steps in the name resolution process

		• Use NSLookup to troubleshoot name resolution • Interpret output from NSLookup • Test a Domain Name System (DNS) server	3.4.1 DNS

Module overview

Configuring network settings is a common administrative task, and for many organizations, it accounts for a significant percentage of overall administrative effort. The Windows 10 operating system includes several tools that you can use to set up and troubleshoot both wired and wireless network connections more efficiently. To support an organization's network infrastructure, it's important that you understand how to use these tools to configure and troubleshoot network connections.

Objectives

After completing this module, you will be able to:

- Describe the available network configuration tools in Windows 10.
- Troubleshoot basic connectivity issues.
- Troubleshoot name resolution.

Microsoft

Lesson 1: Overview of configuration and troubleshooting tools

Before you can determine a Windows 10 device's network settings, you must select an appropriate tool. Windows 10 provides a number of tools that you can use to manage and configure network components, and which one you decide to use depends on your situation and what you are trying to achieve.

The Windows 10 network architecture simplifies network management and the configuration of network connections. When you understand the network architecture and the tools that Windows 10 provides for troubleshooting network connections, you'll be better prepared to configure and troubleshoot network clients, and to support your users.

Objectives

After you complete this lesson, you'll be able to:

- Use the Settings app to configure network settings.
- Use Network and Sharing Center to configure network settings.
- Use command-line tools to configure networking.
- Use Windows PowerShell cmdlets to configure networking.

The Settings app

You can use the Settings app, which Figure 1 depicts, to access network-related settings in Windows 10. To access a device's network settings, open **Settings**, and then select **Network & Internet**.

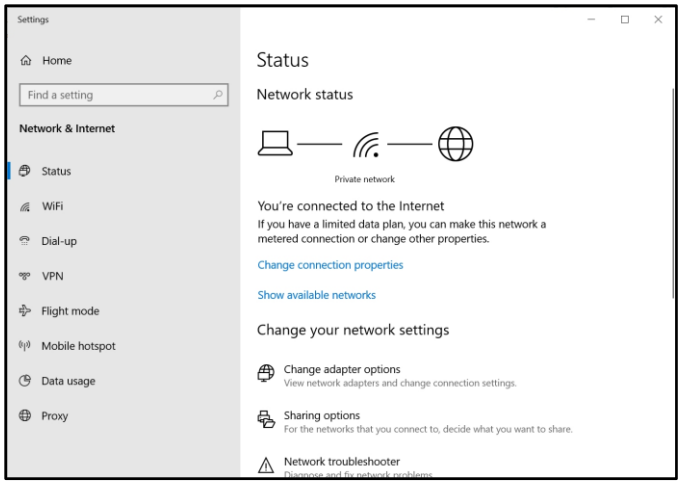

Figure 1. Private network on the Settings app

If you're using a wired connection, select Ethernet. If you are using a wireless connection, select **Wi-Fi**.

Note

You also can access Network & Internet by selecting the network icon in the notification area, and then selecting **Network & Internet settings**.

Microsoft

From within Ethernet or Wi-Fi, you can:

- Change adapter options. You can configure the network adapter settings by selecting from the list of network adapters, and then configuring the properties for each, including:
    - Internet Protocol Version 6 (TCP/IPv6). Use this property to configure the IPv6 settings manually for a given adapter.
    - Internet Protocol Version 4 (TCP/IPv4). Use this property to configure the IPv4 settings manually for a given adapter.

Note

Remember that for almost all situations, using an automatically assigned IP configuration is appropriate. It's also the default.

- Change advanced sharing options. You can configure the following sharing options:
    - Network discovery
    - File and printer sharing
    - Sharing for public folders
    - Media-streaming options
    - The encryption level to use for file-sharing connections
- Launch the Network and Sharing Center. Use this tool to configure most network settings. We'll learn more about this in the next topic.
- Enable and configure a homegroup. You can enable and configure *homegroups*, which are collections of computers that deploy on a home network and share resources such as files and printers. When your device is part of a homegroup, you can share images, media files, documents, and printing devices with other devices in your homegroup.

Note

Although domain-joined computers cannot create homegroups, they can connect to existing homegroups.

- Configure Windows Defender Firewall. You can launch the Windows Defender Firewall tool, and configure Windows Defender Firewall rules, notifications, and advanced settings.

Network and Sharing Center

The Windows 10 Network and Sharing Center, which Figure 2 depicts, provides a clear indication of the status for any wired or wireless connection. You can use it to create additional network connections by using a wizard-driven interface.

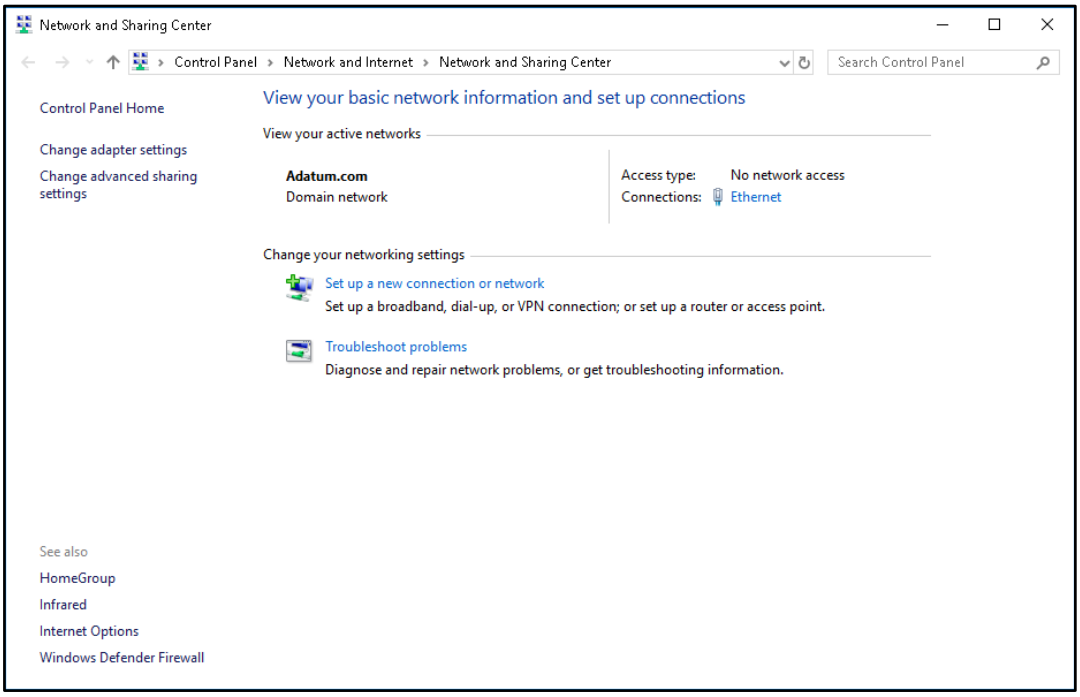

Figure 2. Network and Sharing Center

Microsoft

The Network and Sharing Center also provides links for accessing other network-related tools, including:

- Change adapter settings. Select this link to access a list of your network connections. Typically, there will be one for each network interface card (NIC) installed in your computer. There will also be an entry for each virtual private network (VPN) configured on your computer. To configure specific network settings for one of the adapters, right-click or access the context menu for the appropriate adapter, and then select **Properties**. You can then configure the required settings, as Figure 3 illustrates:

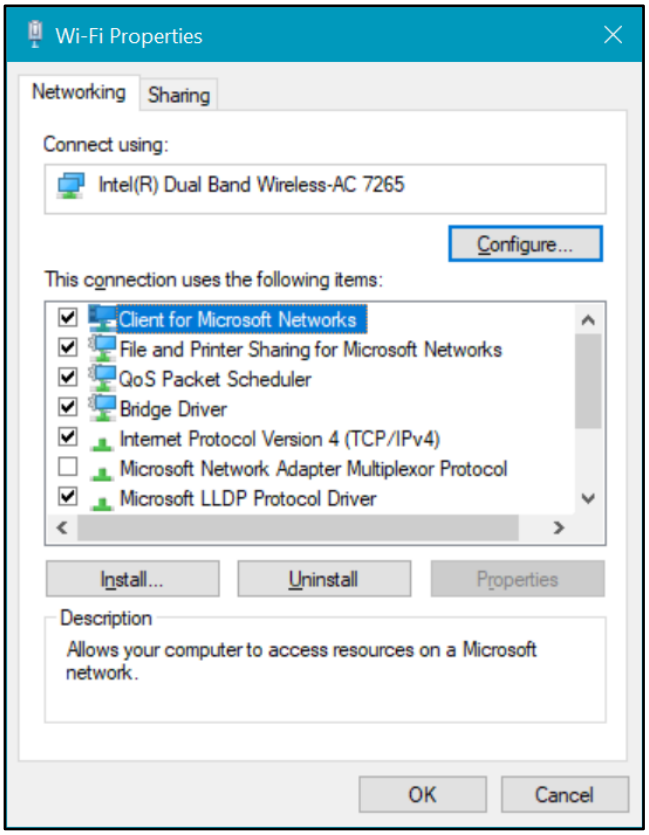

Figure 3. Wi-Fi Properties Networking tab

Note

It's worth remembering that if you install certain apps or enable some Windows 10 features, additional NICs might display here. For example, if you enable the Windows 10 Client Hyper-V feature, you might notice vEthernet adapters listed.

- Change advanced sharing settings. From this option, you can determine how file sharing is managed for your network interfaces. Windows separates your network connections into Private, and Guest, or Public. Clearly, you might want to be more restrictive over what you share when connected to a Guest or Public network. Available options are:
    - Network discovery. This can be turned on or off. Typically, discovery is enabled for Private interfaces, and disabled for Guest or Public interfaces.
    - File and print sharing. Turning this on enables you to share resources on your computer. This is usually disabled on Guest or Public interfaces but is often enabled for Private interfaces.
    - Public folder sharing. The **C:\Users\Public** folder contains subfolders that can be shared to provide access to other users across your network. This is ill-advised for Guest or Public networks. (Note that the public file sharing does not impact specific folder sharing.)
    - File sharing connections. This option enables you to specify the encryption level used when sharing is enabled.
    - Password-protected sharing. By enabling this setting, you restrict shared access only to users that have a user account on your computer. This should be enabled except in circumstances when you want your guests to have access to your public files.
- Internet Options. This link opens the **Internet Explorer Options** dialog box. From here, you can configure Internet Explorer options.

Microsoft

- Windows Defender Firewall. As you learned in earlier modules of the course, a firewall manages the flow of network traffic. Windows Defender Firewall is a host firewall that enables you to manage inbound and outbound traffic on your computer. It's worth remembering that a poorly (or incorrectly) configured firewall can prevent a computer from communicating correctly on a network.
- Network and Internet Troubleshooting Wizard. This wizard enables you to get Windows to check for common network problems and make suggestions about resolutions. It's always worth running this tool if you experience problems. While it's not always successful it can be a time-saver if Windows can identify a problem. To launch this wizard, select the **Troubleshoot problems** link.

Note

Although you can configure many network settings from within the Network and Sharing Center, you're encouraged to use the Settings app or Windows PowerShell cmdlets.

Command-line tools

Although we tend to think of Windows as being a graphical operating system, it does provide a rich collection of command-line tools as well. The following table summarizes tools that you can use for network configuration and troubleshooting.

Tool	Description
IPConfig	The **ipconfig** command displays the current TCP/IP network configuration. Additionally, you can use **ipconfig** to refresh DHCP and DNS settings.
Ping	You use the **ping** command to verify IP-level connectivity to another TCP/IP computer. This command sends and receives ICMP echo request messages and displays the receipt of corresponding echo reply messages. The **ping** command is the

Tool	Description
	primary TCP/IP command that you can use to troubleshoot connectivity.
Tracert	The **tracert** command determines the path taken to a destination computer by sending ICMP echo requests. The path that displays is the list of router interfaces between a source and a destination. This command also determines the latency and speed of the failed router. These results might not be accurate if the router is busy, because the router might assign these specific packets a low priority.
Pathping	The **pathping** command traces a network route similar to how the Tracert tool works. However, **pathping** provides more-detailed statistics on the individual steps—or *hops*—through the network. The command provides more detail because it sends 100 packets for each router, which enables it to establish trends.
NSLookup	The **nslookup** command displays information that you can use to diagnose the DNS infrastructure. You can use the command to confirm a connection to the DNS server and the existence of required records.
Netstat	You can use the **netstat** command to discover information about ports that your client computer and other remote systems are using.
NBTSTAT	When troubleshooting NetBIOS over TCP/IP, you can use the **nbtstat** command-line tool. It displays registered names on the local or remote computer, displays the NetBIOS name cache, and enables you to determine how NetBIOS names are being resolved. (This tool was covered in detail in Module 6, "Name resolution.")
Netsh	You can use the **netsh** command-line tool to review and configure network settings.

Microsoft

Note

Firewalls might block Internet Control Message Protocol (ICMP) requests. Therefore, you might receive false negatives when using Ping as a troubleshooting tool.

IPConfig

One of the most fundamental of all tools is IPConfig. You can use IPConfig with the parameters in the following table.

Parameter	Meaning
ipconfig /all	View detailed configuration information.
ipconfig /release	Release a leased configuration back to the DHCP server.
ipconfig /renew	Renew a leased configuration.
ipconfig /displaydns	View the DNS resolver cache entries.
ipconfig /flushdns	Purge the DNS resolver cache.
ipconfig /registerdns	Register or update the client's host name with the DNS server.

Using Windows PowerShell cmdlets to manage network settings

Although you can use the previously described graphical tools to perform all network configuration and management tasks, it might be quicker or more efficient to use command-line tools and scripts. Windows operating systems have provided command prompts for most network-management tools. However, Windows PowerShell provides

several network-specific cmdlets that you can use to configure, manage, and troubleshoot Windows network connections.

The following table lists some of the network-related Windows PowerShell cmdlets and their purposes.

Cmdlet	Purpose
GetNetIPAddress	Retrieves information about IP address configuration.
Get-NetIPv4Protocol	Retrieves information about IPv4 protocol configuration. (The **Get-NetIPv6Protocol** cmdlet returns the same information for the IPv6 protocol).
Get-NetIPInterface	Returns a list of interfaces and their configurations. This does not include IPv4 configuration of the interface.
Set-NetIPAddress	Sets information about the IP address configuration.
Set-NetIPv4Protocol	Sets information about the IPv4 protocol configuration (the Set-NetIPv6Protocol cmdlet returns the same information for the IPv6 protocol.)
Set-NetIPInterface	Modifies the properties of an IP interface.
Get-NetRoute	Returns a list of routes in the local routing table.
Test-Connection	Runs similar connectivity tests as those that the **ping** command runs. An example is the following command that determines whether the connection is working to a computer known as LON-DC1: `test-connection lon-dc1`
Resolve-Dnsname	Provides a similar function to the NSLookup tool (mentioned above).
Get NetConnectionProfile	Obtains the type of network (public, private, or domain) to which a network adapter is connected.

Microsoft

Cmdlet	Purpose
Clear-DnsClientCache	Clears the client's resolver cache, similar to the **IPConfig /flushdns** command.
Get-DnsClient	Retrieves configuration details specific to the different network interfaces on the computer that you specify.
Get-DnsClientCache	Retrieves the contents of the local domain name system (DNS) client cache, similar to the **IPConfig /displaydns** command.
Get-DnsClientGlobalSetting	Retrieves global DNS client settings, such as the suffix search list.
Get-DnsClientServerAddress	Retrieves one or more DNS server IP addresses associated with the interfaces on the computer.
Register-DnsClient	Registers all of the IP addresses on the computer on the configured DNS server.
Set-DnsClient	Sets the interface-specific DNS client configurations on the computer.
Set-DnsClientGlobalSetting	Configures global DNS client settings, such as the suffix search list.
Set-DnsClientServerAddress	Configures one or more DNS server IP addresses associated with the interfaces on the computer.

The following example depicts how you can use Windows PowerShell to configure the IPv4 settings for a network connection, by typing the following command at a command prompt, and then pressing Enter:

```
Set-NetIPAddress –InterfaceAlias Wi-Fi –IPAddress 172.16.16.1
```

Lesson 2: Troubleshooting basic connectivity

Networks are pretty reliable; however, from time-to-time you might encounter an issue where a computer is unable to connect to a service or an app. It's important that you know how to approach the troubleshooting process and understand the tools available to help you. In this lesson, we'll discuss basic network configuration and how to troubleshoot network problems.

Objectives

After you complete this lesson, you'll be able to:

- Describe a troubleshooting approach to networking problems.
- Verify IP configuration.
- Test connectivity.
- Determine network service and host status.

A troubleshooting approach to networking

When you troubleshoot any problem, you use a particular approach to solve it. If your car doesn't start in the morning, you go through a mental checklist: Has it run out of gas? Is the battery flat?

It's the same with computer problems. If a computer doesn't connect to a server, there are any number of reasons why. Your job is to assess the possible reasons and using your experience and a logical approach to problem-solving, come up with a probable cause.

If there are several probable causes, then you must go through each in turn to eliminate those that are not the cause until you are left with only one. You then test your deductions and assumptions.

Microsoft

A typical computer problem goes through several identifiable phases, which Figure 4 on the next page illustrates:

- Reporting. This phase is when the user reports a problem.
- Gathering. After receiving the report, you gather information about the symptoms. If a user says, "The internet is not working," that's probably inaccurate. They most likely mean that their computer cannot connect to a specific service on the internet.

  You must ask questions to find out more about the nature of the problem. A useful question could be, "How many people are affected?" If only one user is affected, then the problem is most likely with their computer. If more than one person is affected, then the problem is significantly more likely to be with the server or the connecting infrastructure.

- Developing. During this phase, you attempt to determine possible causes and then develop an action plan (that is, what you propose to do). There are two ways to this:

  - Linear. Take a logical step-by-step approach. If your car won't start, you go through the steps of attempting to start the car, testing each stage as you go: Check you're at the right car, and you have the right key. Verify the ignition works and that the engine is turning over. Once you encounter a failure in a step, then you've determined the problem. This approach can often work well, but in complex systems (such as computer networking) where many factors can play a part in a problem, this method might be too time-consuming.

  - Subtractive. In this approach, you create a mental picture of the system's components. If your car won't start, you identify the components that could be the issue: the security system (door, ignition key, alarm); the ignition system (electrical connectors, battery, computer systems, immobilizer); the engine (fuel, combustion, exhaust). (You get the idea.)

  Having drawn up your mental picture, you use your experience and details about the reported symptoms to eliminate the components one by one. So, if your car engine turns over but doesn't start, you know from this symptom that it cannot be the door, ignition key, or battery. You move on to examine other components. In a computer system, you'd be thinking about basic IP configuration and connectivity, infrastructure, routing, and server-based services and apps.

- Implementing. After planning your course of action, you implement your plan. To resolve serious problems, you should consider the impact on service availability that your proposed changes might have. For example, if you think a server restart might solve the problem for your single user but there are other users connected to that server, you'd have to consider the ramifications of disrupting those other users. It's possible that your plan will not work. In that case, you repeat the previous step, consider another component, and draw up another plan of action.
- Documenting. When you finally resolve the problem, you record the details of the problem and its solution. This can help in the future if someone reports problem with similar symptoms.

Again, Figure 4 illustrates the trouble-shooting phases:

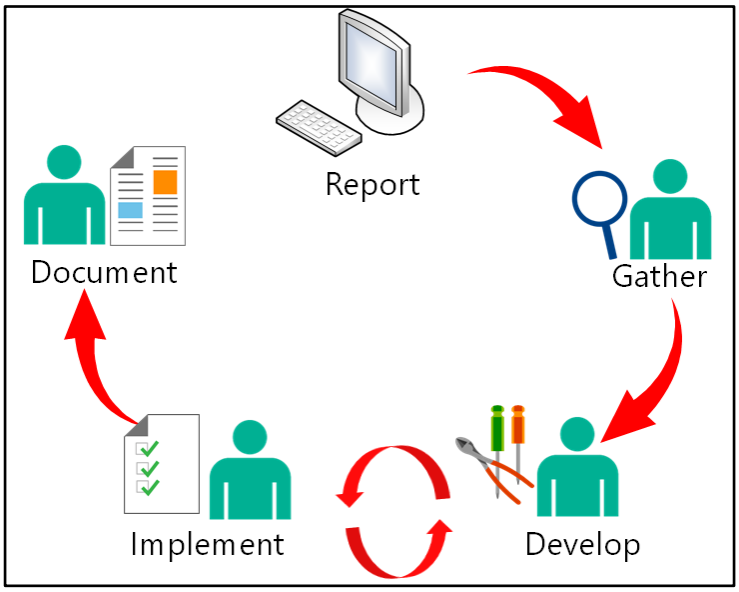

Figure 4. Computer problem phases

Microsoft

How computers establish connections with each other

In Figure 5 on the next page, the computer wants to connect to a web server. The basic steps that the computer performs are as follows:

1. When the computer starts, the NIC initializes. Because the computer has a wireless NIC, it attempts to connect to the configured wireless access point (WAP).
2. The WAP issues an IP configuration that's appropriate for the local subnet.
3. The client begins to establish a connection to the web server, www.microsoft.com. This requires a connection to the internet, initially to resolve the name to an IP address.
4. A DNS server is petitioned for the required host record. It responds with an IP address.
5. The computer attempts to establish a connection to the IPv4 address provided. Specifically, it wants to open an HTTPS connection; this uses TCP over port 443. The firewall that protects the network containing the web server must be configured to allow traffic of TCP over port 443 to transit. The computer must also either know a route to the network that hosts the web server or must pass the network traffic to its configured default gateway.
6. Finally, the web server must be running on the designated server, and the web service must be listening for requests. If any authentication is required, then the computer that wants to connect must provide the required authentication.

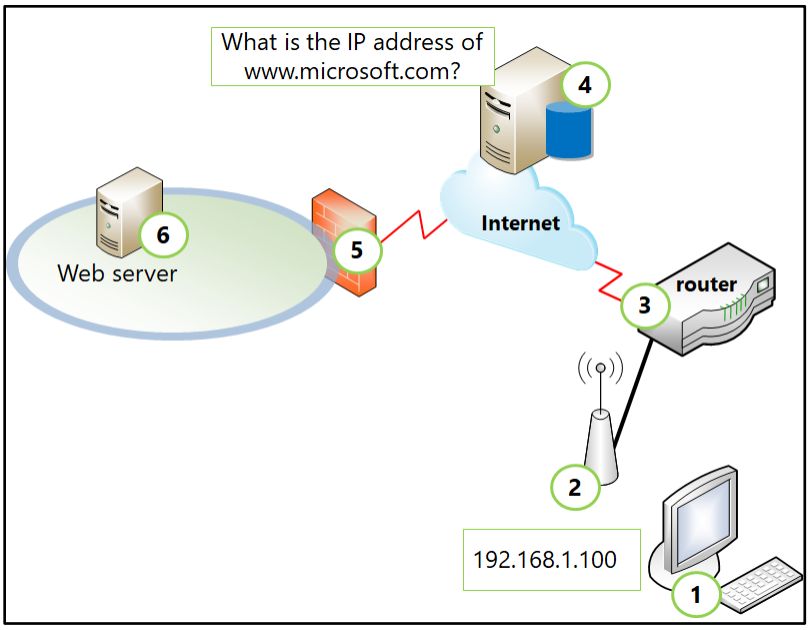

Figure 5. Process for establishing a wireless connection

You can agree that a lot goes into the process of establishing a connection to a web server. This is also a simplified overview.

Note

We have not mentioned digital certificates. These might be required on the server to identify it to the client computer. (HTTPS uses a certificate to verify that the server that's connecting to is who it claims to be). You might also need a certificate to authenticate the client computer to its WAP, and to the server it wants to connect to. While digital certificates are out of the scope of this course, they play a crucial part in helping secure networks. However, they're also a possible source of problems in more complex networks.

From this high-level overview, we can identify basic elements in the connection process:

- Wired or wireless connectivity
- IPv4 or IPv6 configuration
- Routing infrastructure

Microsoft

- Name resolution
- Firewall configuration
- Service availability
- Authentication

Activity: Verifying IP configuration

To determine the local IPv4 configuration, use the **IPConfig /all** command, or the Windows PowerShell cmdlets **Get-NetIPAddress** and **Get-NetIPv4Protocol**. These commands provide information about the local computer, including the:

- IP address
- Subnet mask
- Host name
- DNS server configuration
- DNS suffixes
- Media access control (MAC) address
- Method by which the IP configuration was obtained, such as whether Windows used DHCP to obtain the IP configuration

After running the commands, compare the output from another computer that's in the same subnet as the host with the problem. When you study the output, it's important to remember that:

- The IP address must be in the same host range for the given subnet as the other local computer, while being unique within the subnet.
- The subnet mask must match that of the other local host. If the subnet mask doesn't match, the computer has an incorrect network ID, which can cause communication failures, particularly to remote subnets.

- The default gateway must match that of the other local host. If the default gateway is incorrect or missing, the computer cannot communicate with remote subnets.
- If the DNS server is incorrect or missing, the computer might not resolve names, and communication can fail.

DHCP configures most computers, so if the configuration doesn't match that of the other local host, verify that the computer can obtain an IP address correctly using the following commands:

1. **IPConfig /release**. Use this command to open an elevated command prompt and release the existing address.
2. **IPConfig /renew**. Then use this command to renew the address.
3. **IPConfig /all**. Finally, use this command to review the local IP configuration.

If the host currently has an IP address in the range 169.254.0.0 to 169.254.255.254, the computer most likely failed to obtain a dynamically assigned address. This Automatic Private IP Addressing (APIPA) indicates one of three problems:

- Failure to connect to the DHCP server.
- Issues with the DHCP server configuration.
- Problem with one of the DHCP's scopes.

Activity

In this activity, you will review your network configuration. You can perform this task on any Windows 10 computer to which you have local administrator access.

View IPv4 configuration from the GUI

1. Sign in as a local administrator.
2. Select **Start**, select **Settings**, select **Network & Internet**, and then select **Network and Sharing Center**.

Microsoft

3. In the **Network and Sharing Center**, select **Ethernet**. If you have a Wi-Fi adapter, the network interface might have a different name; for example, Wi-Fi (the SSID).
4. In the **Ethernet Status** dialog box, select **Details**. This window displays the same configuration information for this adapter as the **Ipconfig** command would return.
5. In the **Network Connection Details** window, select **Close**.
6. In the **Ethernet Status** dialog box, select **Properties**. From here you can configure protocols.
7. Select **Internet Protocol Version 4 (TCP/IPv4)**, and then select **Properties**. You can configure the IP address, subnet mask, default gateway, and Domain Name System (DNS) servers on this window.
8. Select **Advanced**. In the **Advanced TCP/IP Settings** window, you can configure additional settings such as additional IP addresses, DNS settings, and Windows Internet Name Service (WINS) servers for NetBIOS name resolution.
9. Close all open windows without modifying any settings.

View IPv4 configuration from a command line

1. Select **Start**, and then enter **Windows PowerShell**.
2. In the returned list of programs, right-click or access the content menu of **Windows PowerShell**, and then select **Run as administrator**.
3. At the Windows PowerShell command prompt, enter the following command, and then press **Enter**:

```
Get-NetIPAddress
```

4. Next, enter the following command, and then press **Enter**:

```
Get-NetIPv4Protocol
```

5. Finally, enter the following command, and then press **Enter**:

```
netsh interface ipv4 show config
```

   The current IPv4 configuration displays.

6. At the Windows PowerShell command prompt, enter the following command, and then press **Enter**:

```
ipconfig /all
```

Completion steps

In this activity, you reviewed the IPv4 configuration by using the Network and Sharing Center, and with Windows PowerShell command-line tools.

After you complete the activity, no settings should have been changed because the processes you used only displayed the information—it didn't make any changes.

Note

When verifying network configuration settings, you need to check the configuration against other devices in the same subnet to ensure the configuration is appropriate. Think back to what we learned about IP addressing, subnetting, and so on. This is where that knowledge is useful.

Test communications

After you verify the local computer's network configuration, you might need to perform some basic connectivity tests to help identify where the problem lies. These basic tests will:

1. Verify basic connectivity.
2. Identify each hop between two systems.

Microsoft

3. Test name resolution.

4. Test connectivity to a specific remote-host server process.

Verify basic connectivity

If the computer has a valid IP configuration but cannot communicate with one or more remote hosts, verify connectivity with the **Portqry**, **Ping**, and **Telnet** commands, or equivalent Windows PowerShell cmdlets.

For example, the **ping** command confirms two-way communication between two devices. This means that if **ping** fails, the local computer's configuration might not be the problem's cause.

```
ping www.microsoft.com
```

You can use either **ping** or the Windows PowerShell **test-connection** cmdlet to ensure communication with a logical process, such as to:

- Ping the remote computer.
- Ping the remote gateway.
- Ping the local IP address.
- Ping the loopback address 127.0.0.1. (This verifies the integrity of the local protocol stack).

Note

If you can ping by IP address and not by name, that indicates a possible name resolution problem.

Identify each hop between two systems

You can use **pathping** and **tracert** to identify each hop between the source and destination systems. If communication fails, these tools can help you identify how many hops are successful, and at which hop the system communication fails.

Although **Tracert** records the hops through which packets travel, **pathping** provides more information about the routing process. **ping** and **pathping** both use ICMP packets to test connectivity to every router between the local host and the remote destination host. **pathping** then calculates statistics about the routes that the packets are using, and the routers involved, including:

- hop number
- round-trip time
- packet loss
- host names
- IP addresses or intermediate hosts

To test routing connectivity to a remote host using **pathping**, open a command prompt, enter the following command, and then press **Enter**:

```
pathping www.microsoft.com
```

The output displays all hops between the local and destination hosts, and the statistical output.

Test name resolution

You can use **NSLookup** to ensure that the DNS server is available. **NSLookup** contains a record for the computer with which you are attempting to communicate. This functionality is vital, because if DNS is not working correctly, even if the computer is available, you might not be able to communicate by using computer names. You can also use the **Resolve-dnsname** Windows PowerShell cmdlet. Due to the complexity of name resolution, we'll examine it in more detail in the next lesson.

Microsoft

Test connectivity to a specific remote-host server process

If you can communicate successfully with a remote host by using the Ping tool, but you cannot access an application on the remote host, it's possible that the remote host isn't listening for your request on the expected port or that local or remote firewalls are blocking your request. You can use the **Portqry** and **Telnet** tools to help identify the cause. We'll discuss this more in the next topic.

Determining status of network services and hosts

Having determined that the underlying network configuration is as it should be and that name resolution is working correctly (which we'll cover in the next lesson), you must verify that a remote service or app is running and available. One way to test this is to verify that the TCP or UDP port used by a service or app is listening.

Verify port availability

To determine whether the remote computer is listening on the expected port, use the **Portqry** or **telnet** tools.

Note

The Portqry tool is available for download from the Microsoft Download website.

For example, to determine if the HTTP port is accessible, at a command prompt, enter the following command, and then press **Enter**:

```
PortQry -n server -e 80
```

The result should resemble the following example:

```
TCP port 80 (http service): LISTENING
```

Note

A returned message indicating that the port is FILTERED or NOT LISTENING can signify that a firewall along the path between the two hosts is blocking the request, or that the application uses a different port, or has failed on the remote host. If other hosts on the local subnet can communicate successfully, the problem most likely exists within the local firewall configuration settings.

You also can use the **telnet** tool to verify that a port is listening. For example, if you want to verify SMTP functionality, you can open a Telnet session to port 25 on the destination host.

Note

You can enable Telnet through Control Panel. In **Programs and Features**, use **Turn Features on or off**, and then select the Telnet program.

At a command prompt, enter the following command, and then press **Enter**.

```
telnet
```

At the Microsoft Telnet prompt, enter the following command, and then press **Enter**:

```
Open LON-dc1.adatum.com 25
```

If the port is available, you should receive a message similar to the following:

```
220 site.adatum.com Microsoft Exchange Server
```


Microsoft

Note

To use **telnet** and **Portqry** to troubleshoot applications, you must understand which ports your applications use. (As you might remember, this was discussed earlier in the course.)

You also can use the **netstat** command to discover information about ports that your client computer and other remote systems are using. The following command lists the active connections on your client computer:

```
Netstat –n
```

Lesson 3: Troubleshooting name resolution

Since almost all connections are established by using names rather than IP addresses, it's important that the name resolution services are working correctly. In this lesson, we'll explore methods you can use to troubleshoot name resolution.

Objectives

After you complete this lesson, you'll be able to:

- Investigate a name resolution problem.
- Use NSLookup to troubleshoot name resolution.
- Interpret output from NSLookup.
- Test a DNS server.

Investigating a name resolution problem

If you can't connect to a remote host and if you suspect a name resolution problem, you can troubleshoot name resolution by using the following process:

1. Open an elevated command prompt, and then clear the DNS resolver cache by typing the following command, and then pressing **Enter**:

```
IPConfig /flushdns
```

Note

Alternately, you can use the Windows PowerShell **Clear-DnsClientCache** cmdlet.

Microsoft

2. Attempt to verify connectivity to a remote host by using its IP address. This helps you identify whether the issue is because of name resolution. You can use the **Ping** command or the **test-connection** Windows PowerShell cmdlet. If the **ping** command succeeds with the IP address but fails by the host name, the problem is with name resolution.

Note

The remote host must allow inbound ICMP echo packets through its firewall for this test to be viable.

3. Attempt to verify connectivity to the remote host by its hostname, using the Fully Qualified Domain Name (FQDN) followed by a period. For example, enter the following command at the Windows PowerShell command prompt, and then press **Enter**:

```
Test-connection LON-cl1.adatum.com
```

Note

You can also use the **ping** command.

If the test is successful, the problem likely doesn't relate to name resolution.

4. If the test is unsuccessful, edit the **C:\windows\system32\drivers\etc\hosts** text file, and then add the appropriate entry to the end of the file. For example, add this line, and then save the file:

```
172.16.0.51   LON-cl1.adatum.com
```

5. Perform the procedure to test by host name (step 3) again. Name resolution now should be successful.

6. Examine the DNS resolver cache to verify that the name resolved correctly. To do this, enter the following command at a command prompt, and then press **Enter**:

```
IPConfig /displaydns
```

Note

You also can use the Windows PowerShell **Get-DnsClientCache** cmdlet.

7. Remove the entry that you added to the hosts file, and then clear the resolver cache again.

Assuming that step 6 was successful, this indicates a problem with name resolution. You must investigate further.

Using NSLookup

You can use **NSLookup** from the command prompt to examine the stages that name resolution passes through. At the command prompt, enter the following command, press Enter, and then examine the contents of the filename.txt file to identify the failed stage in the name-resolution process:

```
NSLookup.exe -d2 LON-cl1.adatum.com. > filename.txt
```

The equivalent Windows PowerShell command is:

```
Resolve-dnsname lon-cl1.adatum.com. > filename.txt
```

The output from the command(s) is redirected to the named text file. You can examine its contents using a text editor such as Notepad.

Interpreting the output from NSLookup

You should understand how to interpret the **NSLookup** command output so that you can identify whether the name resolution problem exists with the client computer's configuration, the name server, or the configuration of records within the name server-zone database.

Microsoft

In the first section of the following output sample, the client resolver performs a reverse lookup to determine the DNS server host name. You can find the following query in the QUESTIONS section:

```
10.0.16.172.in-addr.arpa, type = PTR, class = IN
```

Note

This step is entirely automatic. You will notice that the address 10.0.16.172 is the reverse of 172.16.0.10—the name server's IP address. That's how reverse lookups work.

The returned result, **name = LON-dc1.adatum.com** identifies the host name of the petitioned DNS server, as Figure 6 depicts:

```
------------
SendRequest(), len 41
  HEADER:
      opcode = QUERY, id = 1, rcode = NOERROR
      header flags: query, want recursion
      questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
      10.0.16.172.in-addr.arpa, type = PTR, class = IN

------------
------------
Got answer (73 bytes):
  HEADER:
      opcode = QUERY, id = 1, rcode = NOERROR
      header flags: response, auth. answer, want recursion, recursion avail.
      questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
      10.0.16.172.in-addr.arpa, type = PTR, class = IN
  ANSWERS:
  -> 10.0.16.172.in-addr.arpa
      type = PTR, class = IN, dlen = 20
      name = LON-dc1.adatum.com
      ttl = 1200 (20 mins)

------------
Server: LON-dc1.adatum.com
Address: 172.16.0.10
```

Figure 6. Example of output from NSLookup

Microsoft

In Figure 7, the client resolver performs a recursive query of the DNS server for the host LON-cl1.adatum.com, type = A, class = IN. The returned result is in the ANSWERS section.

Note

The answer also includes a time-to-live (TTL) value, which determines how long the record remains valid.

```
------------
SendRequest(), len 36
  HEADER:
      opcode = QUERY, id = 2, rcode = NOERROR
      header flags: query, want recursion
      questions = 1, answers = 0, authority records = 0, additional = 0

  QUESTIONS:
      LON-cl1.adatum.com, type = A, class = IN

------------
------------
Got answer (52 bytes):
  HEADER:
      opcode = QUERY, id = 2, rcode = NOERROR
      header flags: response, auth. answer, want recursion, recursion avail.
      questions = 1, answers = 1, authority records = 0, additional = 0

  QUESTIONS:
      LON-cl1.adatum.com, type = A, class = IN
  ANSWERS:
  -> LON-cl1.adatum.com
      type = A, class = IN, dlen = 4
      internet address = 172.16.0.51
      ttl = 1200 (20 mins)
```

Figure 7. Another example of output from NSLookup

If you can resolve a computer's name successfully but you can't connect to an application on that computer, investigate whether the local or remote firewalls are blocking your attempt.

Testing a DNS server

The final steps in name resolution testing are to make sure that any name servers are running and working correctly. Issues can occur when you don't configure the DNS server, its zones, and its resource records properly. When resource records cause issues, it can be difficult to identify the issue because configuration problems are not always obvious.

Figure 8 depicts the **Monitoring** tab of the DNS server, TOR-SVR1. From this tab, you can select to:

- Perform a simple query against the server. This verifies that the server can perform a DNS query for zones for which it is authoritative.
- Perform a recursive query to other DNS servers. This verifies that the server can perform a query up the DNS tree to other DNS servers.

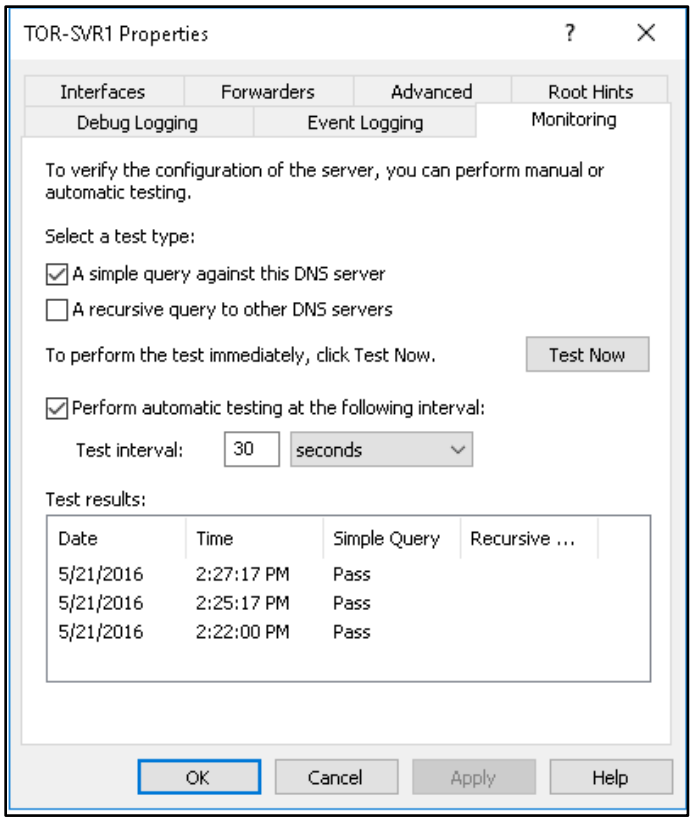

Figure 8. TOR-SVR1 Properties dialog box with the Monitoring tab active

Microsoft

The following table lists possible configuration issues that can cause DNS problems.

Issue	Result
Missing records	Records for a host are not on the DNS server. They might have been scavenged prematurely. This can result in workstations not being able to connect with each other.
Incomplete records	Records that are missing the information required to locate the resource they represent can cause clients requesting the resource to use invalid information. An example of an incomplete record is a service record that doesn't contain a needed port address.
Incorrectly configured records	Records that point to an invalid IP address or have invalid information in their configuration will cause problems when DNS clients try to find resources.

Aside from the testing we discussed using NSLookup, you can also use the **Monitoring** tab on the **DNS server Properties** dialog box.

Note

You'll need permissions to sign into the DNS server, which means you'll most likely need a local administrator account. With this account you can open the DNS console and locate the DNS server you have suspicions about.

From the **Monitoring** tab, you can configure a test that allows the DNS server to determine whether it can resolve simple local queries, and successfully perform a recursive query to ensure that the server can communicate with upstream servers. You also can schedule these tests for regular intervals. These are basic tests, but they provide a good place to start troubleshooting the DNS service.

Possible causes for a test failure include:

- The DNS server service has failed.
- The upstream server is not available on the network.

Microsoft

Learning in action: Troubleshooting networks

Scenario

You've been called the to head offices of Lucerne Publishing in Kensington, London. There have been a number of network-related problems reported, and the on-site IT support staff are unable to identify the causes of these problems. You arrive, review the helpdesk tickets, and set about trying to resolve the problems.

Questions

1. **Study the incident record and answer the question.**

Incident record (support ticket)
Incident reference number: 801124
Date of call August 23 Time of call 13:30 Department Marketing department Status OPEN
Incident details A user in the Marketing department can't connect to the corporate web server.
Additional information No other users are affected. The user cannot connect to other web servers.
Plan of action

Visit the user's computer and verify problem: A. Verify the user's IP configuration. B. Verify that the user's computer is properly connected to the network. C. Verify that the web server is online.
Resolution

What's the best plan of action for the reported problem?

A. Verify the user's IP configuration

B. Verify that the user's computer is properly connected to the network

C. Verify that the web server is online

2. Study the incident record.

Incident record (support ticket)
Incident reference number: 801217
Date of call August 25 Time of call 11:35 Department Publishing department Status OPEN
Incident details

Microsoft

In the Publishing Department, there's been a major problem. All users are reporting connectivity issues with a departmental server.
Additional information The server is on the same network segment as the users in the department.
Plan of action A. Verify that the default gateway for the subnet is online. B. Check the network configuration for the departmental computers. C. Verify that the server is online. D. Determine whether the required services are running on the server.
Resolution

3. **What's the best plan of action for the reported problem? Choose all that apply.**

   A. Verify that the default gateway for the subnet is online.

   B. Check the network configuration for the departmental computers.

   C. Verify that the server is online.

   D. Determine whether the required services are running on the server.

4. **Study the incident record.**

Incident record (support ticket)
Incident reference number: 801229
Date of call August 31 Time of call 17:05 Department Editorial department Status OPEN
Incident details A user in the editorial team has been unable to connect to a Microsoft Word document in her author's web server in the cloud. When she opens a web browser and attempts to enter the required URL, she receives an error.
Additional information Other users are reporting problems in the same department, although they're attempting to connect to other internet-based resources. Local servers seem to be unaffected.
Plan of action A. Use **ping** to verify connectivity to the web server. B. Check the editor's IP configuration. C. Use **portqry** to make sure that the web server is listening. D. Use **tracert** to make certain that the routing infrastructure is working correctly.
Resolution

Microsoft

5. **What's the best plan of action for the reported problem? Choose all that apply.**

   A. Use Ping to verify connectivity to the web server.

   B. Check the editor's IP configuration.

   C. Use Portqry to make sure that the web server is listening.

   D. Use tracert to make certain that the routing infrastructure is working correctly.

Test your knowledge

Select the correct option for the following questions.

1. **Which tool enables you to verify that a service is listening on a remote server?**

   A. Ping

   B. NSLookup

   C. Tracert

   D. Portqry

2. **You want to release a leased IPv4 address back to a DHCP server. Which command should you use?**

   A. Telnet

   B. Ipconfig

   C. NSLookup

   D. Tracert

3. **Which two tools enable you to verify that a port is available on a server?**

   A. Ping

   B. Portqry

   C. Tracert

   D. Telnet

Microsoft

Fill in the blanks for the following questions.

4. **You can use the Windows PowerShell ( ) cmdlet to test name resolution.**
5. **When testing name resolution, you should always start by clearing the ( ) cache.**
6. **You are unable to successfully ( ) a remote computer. However, this could be a false negative as firewalls often block ICMP traffic.**
7. **True or false: You are able to connect to a web server using the URL http://Server1.LucernePublishing.com, but attempting to connect using the https:// prefix is unsuccessful. This is a name resolution problem.**

   True

   False

8. **True or false: You must reconfigure the local network settings on a Windows 10 computer. You want to do so using the IPConfig command from the command prompt. This should work fine.**

   True

   False

Study the scenarios and answer the questions.

9. **Josh at Fourth Coffee calls you up during lunch. He explains that all his customers say they cannot connect to web servers through his Wi-Fi. It was working this morning, but in the last hour there have been problems. He explains that his desktop computers in the back office are working fine.**

   What would you check first?

10. **You're enjoying an espresso and biscotti at Fourth Coffee. Looking at your phone, which is connected to Fourth Coffee's free Wi-Fi, you notice a message arriving in WhatsApp from Josh, the Fourth Coffee owner. He's experiencing problems with his office machines and wants your help in the back. You gulp down your coffee and grabbing your biscotti you head to the offices. When you arrive, you decide to check the basic computer network configuration. When you run IPConfig, you notice that the computer has an IP address of 169.254.0.176.**

    What might be the problem?

Microsoft

Glossary

Term	Definition
DNS	*The Domain Name System provides a hierarchical name resolution service. It is the basis of name resolution on the internet*
DNS zone	*A database containing resource records*
Forward lookup	*A type of DNS zone used to resolve names to IP addresses*
IPCONFIG	*A command-line tool used to review the IP configuration of a host computer*
HOSTS	*A text file containing IP address to hostname mappings*
NSLookup	*A name resolution troubleshooting command-line tool*
Reverse lookup	*A type of DNS zone used to resolve IP addresses to names*