**Module 3 – Configuring Windows 11 Devices**

**Module Overview**

After you’ve completed the installation of Windows 11, you’ll want to configure the operating system. Windows 11 provides a number of tools with which you can perform post-installation configuration. You must know which tool or technique to use in a specific scenario. It’s also important to know how to remotely administer Windows 11, especially in large organizations where physical access to your users’ devices can be difficult.

In this module, you’ll learn how to use the Settings app and Control Panel to configure Windows 11. You’ll also learn how to use Windows Admin Center to remotely manage devices, and also Quick Assist to offer remote help to yours users.

**Objectives**

After completing this module, you will be able to:

* Select a suitable local management tool.
* Use Windows Admin Center to configure Windows devices.
* Use Quick Assist to manage remote computers.
* Use Windows PowerShell cmdlets and scripts.

Lesson 1

**Local configuration tools in Windows 11**

There are numerous tools with which you can configure Windows 11. These include Control Panel, the Settings app, and a collection of management snap-ins that are accessed through the Management Console. It’s important that you know how to use these various tools, and when to use each one.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Use the Settings app.
* Use Control Panel.
* Describe the Management Console.

**Using the Settings app**

The Settings app is now the primary configuration tool for locally configuring Windows 11 computers and devices. Since the early versions of Windows 8, Microsoft has been increasing the reach of this app with each consecutive build.

These days, you should consider Settings as the default administrative and configuration interface. By default, a link exists in the Start menu for accessing Settings. You can also add a small Settings button to the lower right of the Start menu by personalizing the Start > Folders settings. Enable the Settings option.

For anyone familiar with the Settings app in Windows 10, there’s a broad consistency in the look and feel of the app in Windows 11. However, many things have been moved or grouped elsewhere in the app. It seems that Microsoft were keen to create a more logical structure than had evolved over ten Feature updates of Windows 10.

The main navigation pane consists of the elements described in the table below.

|  |  |
| --- | --- |
| Section | Description |
| System | Enables you to configure display settings, sound devices, notifications, focus assist, power and battery settings, storage, nearby sharing, multitasking options, activation, access recovery tools, and configure Remote Desktop. |
| Bluetooth & devices | Allows you to review paired and connected devices, manage Bluetooth settings and devices, and manage printers, scanners, and cameras. You can also configure mouse, touchpad, and pen & Windows ink settings. Provides access to USB settings. |
| Network & internet | Displays fundamental network connection information. Enables you to access and configure WiFi and Ethernet settings, create and manage VPNs, and manage a hotspot. You can also access Advanced network settings |
| Personalization | Consolidates all the personalization settings in a single location. This includes desktop backgrounds, color schemes and themes, the lock screen, fonts, and touch keyboard. You can also access settings for Start and the Taskbar. |
| Apps | Provides access to Apps & features, default apps, and optional features. Also enables you to configure apps for websites, video playback options, and startup app behavior. |
| Accounts | Enables you to configure your account settings. The specifics vary based on the type of account you’re signed in with (for example, a Microsoft account, a Microsoft 365 account, an AD DS domain account, or a local account). You can review account details for email and other apps if these are additional accounts aside from your sign in account.  There’s also access to Sign-in options, such as Widows Hello. In addition, depending on your permissions, you can also add additional accounts for use on this computer by accessing Family & other users or Access work or school.  Curiously, Windows Backup is also accessible from Accounts. In fact, however, the backup tools accessible here are more about synchronizing your data, apps, and preferences between your devices than, say, actually backing anything up. |
| Time & language | Enables you to configure the date and time, language and regional settings, typing options, and speech settings. |
| Gaming | Consolidates the gaming features and controls. |
| Accessibility | Enables you to configure text size, visual effects, mouse and touch settings, and cursor. You can also configure visual aids, like magnifier, color filters, and contrasting themes. Finally, you can configure Narrator from here. |
| Privacy & security | Provides access to security and privacy related settings. If you select Windows Security, the Windows Security app opens in a separate window. Settings accessible here include: device encryption, Find my device, and For developers. This last setting determines whether you are able to sideload apps. Also accessible is a collection of App permissions, including permissions relating to location, camera use, microphone use, notification settings and so on. |
| Windows Update | Enables you to review and configure Windows Update settings, as discussed in the preceding module. |

**Using Control Panel**

When you want to make a configuration change, the Settings app is the best place to start. If it’s necessary for you to make the configuration change using another tool, such as Control Panel, any link you select in Settings opens the appropriate tool for you as necessary.

For example, if you type **Power** in the **Find a setting** text box in the **Settings** app, and then select **Show all results**, you’ll notice that a number of options display that link to Control Panel items. For example, **Edit power plan**. Selecting such an item opens Control Panel in the appropriate location for the change you want to make.

By default, Control Panel displays the configurable settings in your computer by category.

You can change the view to Large icons or small icons. The display then shows an uncategorized list of configurable settings.

These categories are described in the following table.

|  |  |
| --- | --- |
| Category | Description |
| System and Security | Provides access to Security and Maintenance, Windows Defender Firewall, System, Power Options, File History, Backup and Restore (Windows 7), BitLocker Drive Encryption, Storage Spaces, Work Folders, and Windows Tools. |
| Network and Internet | Enables you to select Network and Sharing Center, and Internet Options. |
| Hardware and Sound | Enables you to configure Devices and Printers, AutoPlay, Sound, Power Options, Pen and Touch, Tablet PC Settings, and depending on your device type, Windows Mobility Center. |
| Programs | Provides access to Programs and Features, and Default Programs. |
| User Accounts | Enables you to manage User Accounts, access Credential Manager, and configure Mail settings. |
| Appearance and Personalization | Includes Taskbar and Navigation, Ease of Access Center, File Explorer Options, and Fonts. |
| Clock and Region | Provides access to Date and Time, and Region settings. |
| Ease of Access | Includes both Ease of Access Center and Speech Recognition. |

If you’re unsure of the setting you want, you can enter a text string in the Search Control Panel text box.

It’s important to remember that although Control Panel isn’t yet redundant in Windows 11, it’s not the primary focus for your local administration.

**What are Windows Tools?**

You can access Windows Tools from the bottom of the All apps list in Start. Select **Windows Tools**, and a list of the available tools is displayed.

There are 36 tools displayed, including:

* **Command Prompt**. Enables you to access the command-line interface in Windows. This tool is superseded by Windows Terminal, which provides Windows PowerShell, Command Prompt, and Azure Cloud Shell.
* **Computer Management**. Provides access to System Tools (including Task Scheduler, Event Viewer, and Device Manager), Storage (including Disk Management), and Services and Applications.
* **Local Security Policy**. Enables you to configure the security policy on the local computer, including: Account Policies, Local Policies, Application Control Policies, and Advanced Audit Policy Configuration.
* **Performance Monitor**. Allows you to review current system performance, gather performance and system related data to logs, and review reports based on current or logged activity.
* **Print Management**. Provides a unified interface for printer management.
* **Registry Editor**. Provides a means to make direct changes to the Windows registry.
* **Services**. Opens a list of all configured services in the local computer, and enables you to start, stop, and configure each service.
* **System Configuration**. Enables you to control the startup behavior of the operating system, including forcing the computer into Safe Mode. Useful in troubleshooting scenarios.
* **Task Scheduler**. Enables you to create and manage scheduled tasks.

**Management Console**

Many of these tools link to Control Panel items previously discussed. However, some use a component in Windows called Management Console.

Management Console enables you to create your own custom tools by adding snap-ins that perform specific management tasks. These snap-ins can be focused on the local computer, but can also be focused on a remote computer.

This assumes the remote computer is online, and that you have permissions to perform the management task required by the snap-in, and finally that the Windows Defender Firewall is configured to allow the required network traffic.

**Creating a management console**

To create your own management consoles, run **Mmc.exe** and then select **File** from the menu, and click **Add/Remove Snap-in**.

You can then add one, several, or many snap-ins to perform the requires tasks. After you’ve configured your management console, you must save it for future use.

There are too many management console snap-ins to list here, but they include:

* Certificates
* Device Manager
* Disk Management
* Event Viewer
* Group Policy Object Editor
* Local Users and Groups
* Services
* TPM Management
* Windows Defender Firewall with Advanced Security

Lesson 2

**Using Windows Admin Center**

Windows Admin Center is a web-based management tool. You can use Windows Admin Center to manage Windows clients, such as Windows 10 and Windows 11. But you can also use it to manage Windows Server and your Azure resources, such as Azure virtual machines. Given the ubiquitous nature of Windows Admin Center, it’s important that you understand how to install, configure and use it to manage your Windows 11 devices.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe Windows Admin Center.
* Install Windows Admin Center.
* Use Windows Admin Center.

**What is Windows Admin Center?**

Windows Admin Center is a web-based management tool. Based around the notion of installable extensions, it enables you to manage a variety of device types, including Windows 10, Windows 11, Windows Server, and Azure resources.

It’s benefits include:

* **Easy to install and use**. You can quickly download and install Windows Admin Center. After installation, the simple web-based interface makes it easy to perform typical management tasks.
* **Coexists**. You use Windows Admin Center together with your existing tools, like Configuration Manager, Windows Tools, and Azure Monitor.
* **Anywhere access**. You can publish Windows Admin Center so that your administrators can use it remotely to manage your infrastructure.
* **Good security**. Uses role-based access control (RBAC) to define the scope of management tasks. You can also use the Gateway authentication feature to provide support for local groups, AD DS groups, and Azure AD groups.
* **Cloud integration**. You can use Windows Admin Center to manage your Azure resources as well as your on-premises resources.
* **Extensible**. Because Windows Admin Center is based on extensions, you can add extensions that enable additional management function.
* **No dependencies**. Doesn’t require underlying software components, or a database or agent.

The only dependency is the requirement of Windows Management Framework 5.1 on managed servers.

**Installing Windows Admin Center**

Use the following procedure to download Windows Admin Center:

1. Open **Microsoft Edge**, and search for **Windows Admin Center download**.
2. Navigate to the Microsoft website, and click the link to download Windows Admin Center.
3. Under the Windows Admin Center heading, verify the file type is shown as MSI, and then select **Continue**.
4. Follow the on-screen instructions to download the required file.
5. Navigate to your **Downloads** folder and select the **WindowsAdminCenter2110.msi** file. Note that the precise filename will vary as additional versions are released.

Next, use the Windows Admin Center Setup wizard to complete installation. During the installation process, you are prompted for the following information:

* **Select a port for the Windows Admin Center site**. The default port 6516 is usually acceptable as it’s well-known, and unlikely to conflict with anything else.
* **Allow Windows Admin Center to modify this machine’s trusted hosts settings**. You’ll usually want to do this is a workgroup context.
* When you connect to a remote computer, you must authenticate with that computer. If the remote computer is part of the same AD DS forest as the computer running Windows Admin Center, then Kerberos authentication is used. But where this isn’t the case, you must configure the target computers as trusted hosts. If you bypass this automated setting, you must configure TrustedHosts manually (see below for details).

Use the following procedure to complete installation:

1. Select the **I accept these terms** check box, and then click **Next**.
2. On the **Send diagnostic data to Microsoft** page, select the required diagnostic reporting level, and then click **Next**.
3. On the **Use Microsoft Update to help keep your computer secure and up-to-date** page, select **Next**.
4. On the **Install Windows Admin Center on Windows 10** page, click **Next**.
5. On the **Installing Window Admin Center** page, unless you want to change the TCP port used by the Windows Admin Center site, and unless you want to manually configure trusted hosts, click **Install**.
6. Installation proceeds. When prompted, click **Finish**.

The first time you run Windows Admin Center, you’re prompted to select a certificate. This is discussed in the next topic.

**Configuring trusted hosts**

When operating in a non-domain environment, you must authenticate with the remote computers you want to manage. Part of this authentication requires that for each remote computer, you require an entry in your list of trusted hosts. That is, on the Windows Admin Center computer, you must add a trusted host for each computer you manage.

You can configure trusted hosts settings by using the following Windows PowerShell command in an elevated Windows PowerShell window.

Set-Item WSMan:localhost\Client\TrustedHosts -Value 'ADATUM-DC1.Adatum.com'

You can also use a wildcard setting:

Set-Item WSMan:\localhost\Client\TrustedHosts -Value '\*'

You should be careful using a wildcard as it poses a potential security risk.

You can specify the remote hosts by IP, FQDN, or NetBIOS name.

**Using Windows Admin Center**

To run Windows Admin Center, click **Start**, and then select **All apps**. Search for and select **Windows Admin Center**.

The first time you open Windows Admin Center, you’re prompted to choose a certificate for authentication. Always select the **Windows Admin Center Client** certificate from the list, and then click **OK**.

Windows Admin Center opens. Any required extension updates are downloaded and installed. If necessary, following these updates, Windows Admin Center reloads.

After Windows Admin Center is updated, you can begin to use it to manage computers and resources. To add a new device to the console, select **+Add**. You can then choose between:

* Servers
* Windows PCs
* Server clusters
* Azure VMs

To manage a Windows 10 or Windows 11 computer, after adding it, select the link representing the device in the **Name** list. The device opens.

If you need to specify credentials to connect:

1. Select the check box next to your target device, and then select **Manage as** on the menu.
2. In the pop out **Specify your credentials** blade, enter the required credentials, and then click **Continue**.
3. Now select the link that represents the device in the **Name** list. The device opens.

**Using Tools**

After you’ve connected to your device, you can manage it. The following table describes the available tools.

|  |  |
| --- | --- |
| Tool | Description |
| Overview | Lists the details about the connected computer, including computer name, hardware summary, operating system summary, and a basic overview of system performance. |
| Apps & features | Enables you to review and add/remove apps or features installed on the device. |
| Azure Monitor | Enables you to add the device to Azure Monitor so that you can gather and review performance counters. You’ll need to sign in to your Azure subscription. |
| Certificates | Provides a graphical indication about the status of your certificates, in addition to a list of certificate stores and their respective certificates. You can also export, renew, and delete specific certificates. |
| Devices | Lists the devices installed in your computer. You can disable devices and update drivers, as needed. |
| Events | Enables you to review the event logs of the managed device. |
| Files and file sharing | Review and manage file volumes and shared folders. |
| Firewall | Review the status of the firewall on network location profiles. Also enables you to review the inbound and outbound firewall rules. |
| Local users & groups | Create and manage users and groups on the local computer. |
| Microsoft Defender for Cloud | Activate Microsoft Defender for Cloud, which provides unified security management and advanced threat protection across hybrid cloud workloads. |
| Networks | Review and configure network connections and connectivity. In preview at the time of writing, you can also Add an Azure Network Adapter to enable a Point-to-Site VPN to an Azure virtual network. |
| Performance Monitor | Create a workspace in which you can review the target device performance. |
| Processes | Provides an interface that contains the same sort of information you’d find in Task Manager; that is, a list of running processes and related details. |
| Registry | Enables you to access and edit the local registry. |
| Scheduled tasks | Manage scheduled tasks on the device. |
| Services | Manage and configure services. |
| Storage | Enables you to manage locally attached storage, including creating and managing volumes, formatting volumes, and managing virtual hard disks (VHDs). |
| Security | Review and manage the security settings on the device, including virus and threat protection. You can also perform an antivirus scan. |

The specific options you have will depend on the type of device or service to which you’ve connected.

**Demonstration: Using Windows Admin Center**

Lesson 3

**Using Windows PowerShell**

Although a graphical user interface provides a friendly frontend for managing your devices, it’s not always the best choice. This is especially true when you’re performing repetitive tasks, or bulk management tasks, such as importing lists of objects. In these, and other, circumstances, using a command line interface can be beneficial. A command line interface can also be a time-saver when performing certain remote management tasks.

In this lesson, you’ll learn about Windows PowerShell. You’ll learn how to create commands from cmdlets, and how to perform remote management using PowerShell.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe Windows PowerShell and Windows Terminal.
* Explain cmdlets and commands.
* Work with Windows PowerShell ISE.
* Describe PowerShell Remoting.
* Explain how to run remote commands.

**Overview of Windows PowerShell**

Windows provides two command line interfaces. These are:

* Command Prompt.
* Windows PowerShell.

Command Prompt has been in Windows since the earliest version, back in 1993. It supports the running of commands that would be familiar to users of operating systems, such as Windows 3.0, that preceded Windows NT.

Windows PowerShell originated as a command line interface for managing Exchange Server 2007. Since then, it’s evolved into a ubiquitous component of both Windows Server and Windows client operating systems, such as Windows 11. PowerShell is interesting. It supports the ability to run commands from Command Prompt for backwards compatibility. But it also supports the ability to run a feature-rich command set of its own.

Windows PowerShell is accessible from Windows Terminal. Windows Terminal provides a tabbed command line interface. You can assign Windows PowerShell to one tab, Azure Cloud Shell to another, and even the Command Prompt to a third tab.

You can use Windows PowerShell to manage a local computer. But you can also use it to easily manage remote computers. Indeed, you can use PowerShell to manage Microsoft 365 and Azure cloud services and resources.

**Cmdlets and commands**

PowerShell commands are constructed from cmdlets. Cmdlets are constructed from verb-noun pairs. For example, **Get-LocalUser** retrieves a list of local user accounts. In this example, the verb is **Get** while the noun is **LocalUser**. Let’s examine this in more detail.

**Verbs**

The name of the verb part of a cmdlet tells you what the cmdlet does. There’s a standard set of approved verbs that are used; this helps provides consistency across all cmdlets. Common verbs include:

* Get. Retrieves a resource.
* Set. Changes the data associated with a resource.
* New. Creates a resource.
* Add. Adds a resource to an object of multiple resources (like adding a member to a group)
* Remove. Deletes a resource from an object that contains multiple resources (like removing a member from a group).

These are just a few examples of the many available verbs.

**Nouns**

The name of the noun in a cmdlet tells you what kinds of resources or objects the cmdlet affects. All cmdlets operating on the same resource must use the same noun.

Nouns sometimes use prefixes. These prefixes can help understand more about what the resource is. For example, LocalUser versus ADUser. The former is a user resource on a local computer, while the latter is a user in Azure AD.

**Parameters**

You specify parameters to modify the actions that a cmdlet performs. You can run cmdlets without any parameters. However, you can also run cmdlets with one or several parameters. To call out a parameter, you precede it with a dash (-).

If you want to pass a value from the parameter, use a space to separate the parameter name and the value.

If the value you pass contains spaces, enclose the text in quotation marks.

**Examples**

The following command displays a list of running services:

Get-Service | Where-Object {$\_.Status -eq "Running"}

The following command displays a list of services that have a name that begins “A”:

Get-Service -Name "A\*"

This next command outputs a list of all local users to a text file formatted for HTML output:

get-LocalUser | ConvertTo-Html > Users.html

The vertical bar is known as the pipe. You can use the pipe to join two or more commands together.

**Windows PowerShell ISE**

Windows PowerShell ISE provides a unified interface that combines a Windows PowerShell command prompt, a script pane, and a list of commands. It’s a great interface to work with scripts, but also to get to grips with PowerShell. For instance, you can use the Command add-on to enter a cmdlet, and can review details of the cmdlet syntax, supported parameters, and other details.

In addition, when you work in the command prompt, Intellisense is used to offer up options about which cmdlet you might want. For example, entering **Get-Local** offers up six possible cmdlets that include the noun **local** in the name.

The script pane enables you to create and edit PowerShell scripts. But it also enables you to run the script. Line at a time, if you prefer, to help debug errors in your script. There’s even an actual debugger for more advanced situations.

**Overview of PowerShell Remoting**

One of the great things about using Windows PowerShell is that you can apply your commands to both local and remote computers with little, if any, modification. For example, some cmdlets accept a computer name parameter that let you specify the name of one or more computers against which to run the command.

Because of this ability, PowerShell Remoting supports three configuration models:

* **One-to-One**. In this scenario, you connect to a one remote computer and run your PowerShell commands on it. The commands run just as if you were running them locally in a Windows PowerShell window.
* **One-to-Many**. In this scenario, your command executes on multiple computers simultaneously. However, you’re not working at each remote computer interactively. Rather, your batch executed remotely, and results are returned to your computer.
* **Many-to-One**. This scenario occurs when multiple administrators connect to the same computer remotely, each with their own remote session. Any administrative tasks run independently of any other.

**Requirements**

**Windows Remote Management**

Before you can use PowerShell Remoting, you must prepare your remote computers. PowerShell Remoting uses the Windows Remote Management (WinRM) interface to communicate. You’ll need to start this service, or more accurately, the WinRM Listener. You’ll also need to enable Windows Remote Management through the Windows Defender Firewall.

The easiest way of completing these tasks is to run one of the following commands:

* At a Command Prompt, run **winrm quickconfig**
* In a Windows PowerShell window, run **Enable-PSremoting -force**

In both cases, you must use an elevated command line.

Windows Server computers are enabled for Windows Remote Management by default. Windows 10 and Windows 11 are both disabled by default.

**Trusted hosts**

As with Windows Admin Center, when you run commands against a remote computer, you must authenticate to that computer. If the computer is part of the same AD DS forest, that’s easy because your existing credentials are acceptable. However, in a workgroup environment, you’ll need to configure trusted hosts in exactly the same way as for Windows Admin Center.

Remember, you can configure trusted hosts settings by using the following Windows PowerShell command in an elevated Windows PowerShell window.

Set-Item WSMan:localhost\Client\TrustedHosts -Value ‘ADATUM-DC1.Adatum.com'

**Running remote commands**

There are a couple of ways you can use to execute your PowerShell commands on a remote computer. As mentioned earlier, one way is to use the -ComputerName parameter. But there are others.

**Temporary sessions**

To create a temporary remote session, that is, one which is in place for the duration of running the command, use the **Invoke-Command** cmdlet.

For example:

Invoke-Command –ComputerName ADATUM-CL1 –ScriptBlock {Get-Service -Name "A\*"}

You can specify multiple computer names by separating names with a comma. For example:

Invoke-Command –ComputerName ADATUM-CL1,ADATUM-CL2 –ScriptBlock {Get-Service}

**Persistent sessions**

To create a persistent remote session, that is, one which remains in place until you terminate it, use the **New-PSSession** cmdlet.

For example:

$S = New-PSSession -Computername ADATUM-CL1

Then use:

Enter-PSSession $S

You now see a PowerShell prompt that includes the target computer’s name:

[ADATUM-CL1]: PS C:\>

As before, you can specify multiple computer names in both these examples by separating names with a comma.

When you want to close the remote session, use:

[ADATUM-CL1]: PS C:\> Exit-PSSession

**Running scripts remotely**

To run a local script on remote computers, use the **FilePath** parameter with **Invoke-Command**.

For example, the following command runs Test.ps1 on the ADATUM-CL1 computer:

Invoke-Command -ComputerName ADATUM-CL1 –FilePath C:\Scripts\Test.ps1

The results of the script are returned to the local computer.

Using the *FilePath* parameter means there’s no need to copy any files to the remote computers.

**Demonstration: Using PowerShell Remoting**

Lesson 4

**Implementing remote management**

In a small networked environment, in might be possible for support staff to physically attend a user’s desk and help resolve computer problems. However, in large networks, this becomes impractical. Given that many users now work from home, at least some of the time, then it’s important to consider remote management options.

You’ve already learned that you can use Windows PowerShell Remoting to manage remote servers and workstations. However, it’s also possible to use graphical tools to provide technical support, or to manage remote servers.

**Lesson Objectives**

After completing this lesson, you will be able to:

* Describe RSAT
* Use Remote Desktop
* Use Quick Assist

**Remote Server Administration Tools**

Windows Server provides numerous management tools for server administrators. These tools are installed as part of server roles that are added to a specific server computers.

However, this has meant that in order to manage a server, you must be physically at the server using the installed tools. This is not best practice. Working interactively with a server presents security and physical environmental risks. You can’t spill coffee over your server if you’re not standing next to it.

The Remote Server Administration Tools (RSAT) enable you to install the required management tools for your servers on your workstations. In earlier versions of Windows, such as Windows 7, you had to download an installer file for RSAT to install the required tools. In Windows 10 and Windows 11, you use the Settings app to add features to Windows.

The available server management tools include:

|  |  |
| --- | --- |
| Tool | Tool |
| * Active Directory Certificate Services Tools * Active Directory Domain Services and Lightweight Directory Services Tools * BitLocker Drive Encryption Administration Utilities * DHCP Server Tools * DNS Server Tools * Data Center Bridging and LLDP Tools * Failover Clustering Tools * File Services Tools * Group Policy Management Tools * IP Address Management (IPAM) Client | * Network Controller Management Tools * Network Load Balancing Tools * Remote Access Management Tools * Remote Desktop Services Tools * Server Manager * Shielded VM Tools * Storage Migration Service Management Tools * Storage Replica Module for Windows PowerShell * System Insights Module for Windows PowerShell * Volume Activation Tools * Windows Server Update Services Tools |

It’s important to note that after installing RSAT, it’s necessary for your servers to allow for remote management. Some services use WinRM, in which case, the necessary listener service and firewall exceptions are already defined on your server. But other services use network connectivity over Named Pipes and Remote Procedure Calls (RPCs). These use different TCP ports for connectivity, and you’ll need to configure Windows Defender Firewall to allow this connectivity.

**Remote Desktop**

One possible solution to avoiding the need of installing RSAT and configuring the firewall is to use Remote Desktop. Remote Desktop uses the Remote Desktop Protocol (RDP) which works over TCP port 3389. This minimizes the configuration work that you’ll need to undertake on target devices as you’ll only need to define a single inbound firewall rule to allow TCP 3389.

Having done that, you can use Remote Desktop to connect to target servers and workstations. This facility gives you complete remote control of the target device. However, it does force the sign out of any interactively signed in users. This doesn’t make it ideal for troubleshooting client workstations, especially when you might want to interact with those users; for example, to provide guidance.

But Remote Desktop is a convenient way of connecting to your servers. However, it’s worth considering that exposing the well-known RDP port to the internet would be ill-advised. So, using RDP connections to manage Azure-based resources, such as VMs, is not recommended.

The easiest way to enable Remote Desktop is to use the **Settings** app. In **Settings**, select **System** and then select **Remote Desktop**. Enable the **Remote Desktop** setting. You can also choose to configure which users are permitted to connect with RDP. By default, members of the Local Administrators group can connect.

Having enabled Remote Desktop, you use the Remote Desktop Connection (RDC) program (mstsc.exe) to establish remote connections.

When you run RDC, you connect to a remote computer by specifying its IP address of computer name. You’ll also need to provide authentication credentials. You can choose to configure a number of additional settings to enhance your connection experience. These are described in the following table.

|  |  |
| --- | --- |
| Tab | Option |
| General | Enter the computer and username, and then select whether to save the connection as an RDP file. |
| Display | Select the remote display’s screen size and color quality. |
| Local Resources | Use the remote computer’s resources in your session, such as a printer or clipboard. |
| Experience | Configure the way you want the remote session to appear visually. The more features that you add, the more bandwidth it utilizes. |
| Advanced | Tell the Remote Desktop client how to behave if the RDP server fails to prove its authenticity. You can choose whether to connect without warning or to receive a warning, and whether you want to connect or prevent the connection. |

Microsoft recommend that you use Windows Admin Center or Windows PowerShell Remoting to remotely manage your devices rather than using Remote Desktop.

**Quick Assist**

Quick Assist is relatively new, being added to a late version of Windows 10. It’s based on the same RDP protocol as Remote Desktop, but is designed for support staff to offer and provide remote help to specific users. Quick Assist is built-in to Windows 11.

**How it works**

When a user needs help, they run Quick Assist, and then they await instructions from their helper.

Meanwhile you (their helper) open Quick Assist on your computer. You select the option to **Assist another person**. A sign in prompt is displayed. Sign in using an organizational account, such as a Microsoft 365 Azure account.

A code is generated and displayed in Quick Assist on your computer. This code is valid for 10 minutes. Communicate the code to your user.

The user enters the code you provided in the **Code from assistant** text box. They then select **Share screen**.

Back on your computer, a message displays in Quick Assist prompting you to choose between **Take full control** and **View screen**. Make the appropriate selection and then click **Continue**.

Finally, the user must confirm the connection by selecting **Allow** in the Quick Assist dialog box. Connectivity is established.

**What you can do**

After you’ve established a session, you can:

* Communicate with your user using chat
* Annotate the screen(s)
* Control the remote computer (assuming you’re connected with Take full control)

**Demonstration: Using Quick Assist**

**Lab: Configuring Windows 11 devices**

**Question:**In the lab, you didn’t need to configure trusted hosts. Why?

**Module Review and Takeaways**

Review Questions

**Question:**What’s the difference between Command Prompt and Windows PowerShell?

**Question:**What command should you use to configure trusted hosts for wildcard in a non-domain environment?

**Question:**What does **winrm quickconfig** do?

Tools

The following table lists the tools that this module references.

| **Tool** | **How used** | **Where found** |
| --- | --- | --- |
| Windows Admin Center | * Remote management | Download from Microsoft download website |
| Windows PowerShell | * Local and remote command-line and scripted management | Included in Windows 11, accessed from Windows Terminal app |
| Quick Assist | * Help users remotely by offering remote control support | Included in Windows 11. |

Common Issues and Troubleshooting Tips

| **Common Issue** | **Troubleshooting Tip** |
| --- | --- |
| You are unable to establish a remote management session | Check whether trusted hosts are configured.  Ensure that WinRM is enabled on the target host |