Windows Commands

Most important site for learning and understanding Windows Commands:
https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands

Process Explorer v17.06 (as of this day - 10/25/25)
https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer

Taskkill
https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/taskkill

Diskpart
https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart

Get-Process
https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-process?view=powershell-5.1

Signal
https://learn.microsoft.com/en-us/cpp/c-runtime-library/reference/signal?view=msvc-170

—----------

Start-Transcript

In the case of Start-Transcript, you can call it like this:

Start-Transcript -Path C:\Transcript.txt

This will write the contents of the session to C:\Transcript.txt. When you want to stop recording you need to call Stop-Transcript. The file created is a plain text file where the commands executed and their outputs are stored.

Self-extracting Executable

While it is common to install software using the Windows Installer, it is helpful for you to know how to install software using the command line.

Self-extractor packages are executable files (.exe) that are run in the Windows interface by clicking on them or running from the command line. Software installed by an IT professional onto an end user's computer will likely use this format. Software installation package, update package, or hotfix package created with the Microsoft Self-Extractor, can be executed using the following command lines:

- /extract:[path]: Extracts the content of the package to the path folder. If a path isn't specified, then a Browse dialog box appears.
- /log:[path to log file]: Enables verbose logging (more detailed information recorded in the log file) for the update installation.
- /lang:lcid: Sets the user interface to the specified locale when multiple locales are available in the package.
- /quiet: Runs the package in silent mode.
- /passive: Runs the update without any interaction from the user.
- /norestart: Prevents prompting of the user when a restart of the computer is needed.
- /forcerestart: Forces a restart of the computer as soon as the update is finished.

You can always type /?, /h, or /help from the command line to view these options.

Resources

- Windows Server performance troubleshooting documentation - Microsoft list of articles on common Windows Server errors, troubleshooting, and solutions.
- How to scan and repair disks with Windows 10 Check Disk - Instructions for using the CHKDSK command.
- Overview of Disk Management - Lists uses for the Windows Disk Management system utility, along with links to step-by-step instructions for using the utility.
- How to use Event Viewer on Windows 10 - A walkthrough tour of Windows Event Viewer with screenshots and detailed explanations of each part of the tool.
- Registry - Microsoft article about the Windows Registry.
- How to use System Configuration tool on Windows 10 - Tutorial for using the Windows System Configuration tool.

Supplemental Reading for OS Deployment Methods

OS Deployment Methods

In this reading, you will learn about operating system (OS) deployment methods, including the use of disk cloning. A cloned disk is an identical copy of a hard drive. Cloning is often used when an Enterprise company purchases a large number of identical computers. The IT Support Administrators for the company are responsible for installing and configuring the computers to meet the needs of the company and its network. Disk cloning is used to save time on this type of deployment. IT Administrators will select one of the new computers to install and configure needed items, such as the OS, utilities, tools, network settings, software, drivers, firmware, and more. Then they make a clone of this first hard drive. The cloned disk is used to copy the entire disk image over to the remaining new computers so that the IT Admins do not need to repeat the same installation and configuration steps on each new computer. They may keep a copy of the original disk from this deployment to reimage the systems if a clean OS install is required (e.g., following a virus or malware infection, OS corruption, etc.).

Cloned disks have uses beyond deploying OSs. They can be used to test new software and configurations in a lab environment before applying the updates to similar production systems. Cloning can also be used for system migrations, data backups, disk archival, or to make a copy of a hard drive for investigative or auditing purposes.

Tools for duplicating disks

Hard disk duplicator

Hard drive duplicators are machines that can make identical copies of hard drives. The original drive is inserted into the duplicator machine along with one or more blank hard drives as targets. Disk duplicators can have anywhere from a single target bay for limited disk cloning (example use: law enforcement investigations) up to 100+ target bays for industrial use (example use: computer manufacturing). If the target drives are not blank, the duplicator machine can wipe the drives. The target drives usually need to share the same characteristics (e.g., interface, form factor, transfer rate) of the original drive. The targets should also have the same or greater storage capacity than the original.

The hard drive duplicator may have an LCD interface built-in to the machine and/or a management software/HTML interface, the latter of which can be accessed over a networked or directly-connected computer or server. The duplicator interface can be used to initiate and manage disk cloning and/or disk wiping (reformatting). Most duplicators copy data sector-by-sector. The time to transfer data from the original to the target drives depends on multiple variables. The machine's user manual should be consulted to calculate duplication time.

Disk cloning software

Hard drives can also be cloned using software. This method allows the original and target to be different media from one another. For example, a hard drive can be cloned from an IDE drive to an SSD drive, a CD-ROM/DVD, removable USB drive, cloud-based systems, or other storage media, and vice versa. Software cloning supports full disk copies (including the OS, all settings, software, and data) or copies of selected partitions of the drive (useful for data-only or OS-only copies). Disk cloning software is often used by IT Administrators who need to deploy disk images across a network to target workstations or to cloud-based systems. Cloud platforms normally offer a virtual machine (VM) cloning tool as part of their services. **VM cloning is the most efficient method for cloning servers and workstations. VM cloning takes a few seconds to deploy new systems.**

A few examples of disk cloning software include:

- **NinjaOne Backup** - Cloud-based cloning, backup, and data recovery service designed for managed service providers (MSPs) and remote workplaces.
- **Acronis Cyber Protect Home Office** - Desktop and mobile device cloning software that works with Windows, Apple, and Android systems. Designed for end users. Supports backup, recovery, data migration, and disk replication. Includes an anti-malware service that can overcome ransomware attacks.
- **Barracuda Intronis Backup** - Cloud-based cloning and backup service on a SaaS platform. Designed for MSPs who support small to mid-sized businesses. Can integrate with professional services automation (PSA) and remote monitoring and management (RMM) packages.
- **ManageEngine OS Deployer** - Software for replications, migrations, standardizing system configurations, security, and more. Can create images of Windows, macOS, and Linux operating systems with all drivers, system configurations, and user profiles. These images can be saved to a locally stored library. The library is available to deploy OSs to new, migrated, or recovered systems as needed.
- **EaseUS Todo Backup** - Free Windows-compatible software for differential, incremental, and full backups, as well as disaster recovery. Supports copying from NAS, RAID, and USB drives.

Methods for deploying disk clones

The sections above have described disk clone deployment through copied hard drives, image libraries, network storage, and cloud-based deployments. There are some other options for cloned disk deployments:

Flash drive distribution

OSs can be distributed on flash drives. IT professionals can format flash drives to be bootable prior to copying a cloned disk image to the flash drive. Target systems should be set to boot from removable media in the BIOS. After inserting a flash drive containing the OS into an individual computer, restart the system and follow the prompts to install the OS. Microsoft offers this method as an option for Windows installations. Linux systems can also be booted and installed from flash drives.

The Linux dd command

The Linux/Unix dd command is a built-in utility for converting and copying files. On Linux/Unix-based OSs, most items are treated as files, including block (storage) devices. This characteristic makes it possible for the dd command to clone and wipe disks.

For more information on disk cloning and OS deployment techniques, please visit:

- How to clone a hard drive on Windows - Step-by-step guide with screenshots on how to clone a hard drive using the software Macrium Reflect Free.
- Best Hard Drive Duplicator/Cloner Docking Station for 2022 - Comparison guide to popular hard drive duplicator machines.
- OS deployment methods with Configuration Manager - Microsoft's guide to options for deploying Windows in a network environment.
- dd(1) - Linux manual page - The manual for the Linux dd command, which describes how to use the command and lists the available optional flags.

Terms

~ (tilda) - A *shortcut* for the path of your home directory.
Example: If in the “music” folder, instead of writing > cd c:\users\dave\documents
instead, write > cd ~\documents to switch *to the documents.

ls - Command parameter for “List Directory”, which is the alias of:
Dir -Command parameter for “List Directory, which is the alias as well..

‘ (back tick) - Used to “escape the space” when typing with spaces. Literally means that the next character after the back tick should be treated literally. Example: creating a folder named “My Cool Folder!” written as > My’ Cool’ Folder!

-Recurse (dash-Recurse) - Instructs a command to operate on all items in the specified directory **and its subdirectories**. For example, when used with `Get-ChildItem`, it lists all nested files/folders. With `Copy-Item`, it copies the entire directory structure and contents.

-Verbose (dash-Verbose) - Displays detailed information about the steps a command is performing as it runs. Useful for troubleshooting, logging, or understanding what’s happening behind the scenes during execution.

> Echo -

> (greater than/) - A redirector command that lets the user change where the standard output will go. If we send it to a location where a file exists, it will overwrite the file. If there’s no file, it will create one. If we don’t want to overwrite an existing file, then use: >>

| (pipe) - Is used to send the output of one command to the input of another.

Example: > cat words.txt | Select-String st > st_words.txt
This command will grab the “words.txt” file, select words with “st” in them, and > those words into a new document “st_words.txt”

stdout (1)- Standard output.
stderr (2)- Standard error.
$null - It’s “nothing”.
Example: > rm secure_file 2> errors.txt – Tells powershell to redirect the standard error stream to the file instead.
Example: > rm secure_file 2> $null - Tells powershell to hide the error, or send it to the black hole.

Dynamic Link Library (DLL)

A few common DLLs used by Windows include:

- .drv files - Device drivers manage the operation of physical devices such as printers.
- .ocx files - Active X controls provide controls like the program object for selecting a date from a calendar.
- .cpl files - Control panel files manage each of the functions found in the Windows Control Panel.

DLL dependencies can be broken when:

- Overwriting DLL dependencies - It is possible for an application to overwrite the DLL dependency of another app, causing the other app to fail.
- Deleting DLL files - Some applications and malware may delete the DLLs needed by other applications installed on a system.
- Applying upgrades or fixes to DLLs - Can cause a problem called “DLL hell” where an application installs a new version of the shared DLL for a computer system. However, other applications that are dependent on the shared DLL have not yet been updated to be compatible with the new version of the DLL. This causes the other applications to fail when the end user tries to launch them.
- Rolling-back to previous DLL versions - A user may try to reinstall an older application that stopped working after a shared DLL file was upgraded by a newer app. However, the reinstallation of the app that uses the old DLL version can overwrite the new DLL file. This DLL version roll-back can cause the newer app with the shared DLL dependency to fail the next time it tries to run.

Microsoft has remedied these problems through the use of:

- Windows File Protection - The Windows OS controls the updates and deletions of system DLL files. Windows File Protection will allow only applications with valid digital signatures to update and delete DLL files.
- Private DLLs - Removes the sharing option from DLLs by creating a private version of the DLL and storing it in the application’s root folder. Changes to the shared version of the DLL will not affect the application’s private copy.
- .NET Framework assembly versioning - Resolves the “DLL hell” problem by allowing an application to add an updated version of a DLL file without removing the older version of the DLL file. This prevents the malfunction of applications that have dependencies on the older DLL file. The DLL versions can be found in the "C:\Windows\assembly" path and are placed in the Global Assembly Cache (GAC). The GAC contains the .NET “Strong Name Assembly” of each DLL file version. This “Strong Name Assembly” includes the:
    - name of the assembly - multiple DLL files can share the assembly name

- version number - differentiates the version of DLLs
- culture - country or region where the application is deployed, can be “neutral”
- public key token - a unique 16-character key assigned to an assembly when it is built

Side-by-side assemblies

DLLs and dependencies can also be located in side-by-side assemblies. A side-by-side assembly is a public or private resource collection that is available to applications during run time. Side-by-side assemblies contain XML files called manifests. The manifests contain data similar to the configuration settings and other data that applications traditionally stored in the Windows registry. Instead of registering this data in the Windows registry, the applications store shared side-by-side assembly manifests in the WinSxS folder of the computer. Private manifests are stored inside the application’s folder or they can be embedded in an application or assembly. The metadata of a manifest may include:

- Names - Manages file naming.
- Resource collections - Can include one or more DLLs, COM servers, Windows classes, interfaces, and/or type libraries.
- Classes - Included if versioning is used.
- Dependencies - Applications and assemblies can create dependencies to other side-by-side assemblies.

As an IT Support professional, this concept should be considered when troubleshooting application issues. If the application’s configuration settings are not found in the Windows registry, they might be located in the manifest from the app’s side-by-side assembly.

When a device is plugged into a Windows machine, 2 things happen:

1. The hardware is ID’d: https://learn.microsoft.com/en-us/windows-hardware/drivers/install/step-1--the-new-device-is-identified
    a. The process: https://learn.microsoft.com/en-us/windows-hardware/drivers/install/hardware-ids
2. A Driver Package for the Device is Selected: https://learn.microsoft.com/en-us/windows-hardware/drivers/install/step-2--a-driver-for-the-device-is-selected

---

Types of Windows updates

There are several types of updates that the Windows Update Client might find for your Windows system.

- Critical updates address critical bugs that are not security related. These are widely released fixes for a specific problem.
- Definition updates are widely released and frequent updates to a product's definition database. Definition databases are used to detect specific types of objects on your system, such as malicious code, phishing websites, or junk mail.
- Driver updates: Drivers are software that control the input and output of devices running on your system. This software may be updated when new versions of the driver become available for your devices or if you install a new device on your system.
- Feature packs add new product functionality to your system. This functionality is first distributed as an update to a product currently running on your system. It is usually included in the next full product release.
- Security updates are widely released patches for a security related vulnerability. Security vulnerabilities are rated by severity as being critical, important, moderate, or low.

a) Critical vulnerabilities pose an active threat. Patch should be installed immediately.
b) Important vulnerabilities pose a likely threat. Patch should be installed as soon as possible.
c) Moderate vulnerabilities pose a potential threat. Patch should be installed soon.
d) Low severity vulnerabilities are not an immediate threat, but a patch is recommended.

- Service packs collect all tested hotfixes, security updates, critical updates, and general updates together and distribute them as a set. A service pack also may contain new fixes or design changes requested by customers.
- General updates are widely released fixes for specific problems. They address noncritical bugs that are not security related.
- Update rollups collect a set of tested hotfixes and updates that target a specific area, such as a component or service. These fixes and updates are packaged together for easy deployment.
- Security-only updates collect all the new security updates from a given month for distribution through the Windows Server Update Services (see below). These updates are called “Security Only Quality Update” when you download them and will be rated as “Important.”
- New OS: A new version of the Windows operating system may also be deployed through the Windows Update Client. For example, Windows 10 and 11 were both delivered as updates to a previously installed OS.

Windows Paging Files

A paging file is an optional tool that uses hard drive space to supplement a system’s RAM capacity. The paging file offloads data from RAM that has not been used recently by the system. Paging files can also be used for system crash dumps or to extend the

system commit charge when the computer is in peak usage. However, paging files may not be necessary in systems with a large amount of RAM.

Page file sizing

Determining the size needed for a paging file depends on each system's unique needs and uses. Variables that have an impact on page file sizes include:

- System crash dump requirements - A system crash dump is generated when a system crashes. A page file can be allocated to accept the Memory.dmp. Crash dumps have several size options that can be useful for various troubleshooting purposes. The page file needs to be large enough to accept the size of the selected crash dump. If the page file is not large enough, the system will not be able to generate the crash dump file. If the system is configured to manage page dumps, the system will automatically size the page files based on the crash dump settings. There are multiple crash dump options. Two common options include:
    - **Small memory dump**: This setting will save the minimum amount of info needed to troubleshoot a system crash. The paging file must have at least **2 MB of hard drive space allocated** to it on the boot volume of the Windows system. It should also be configured to generate a new page file for each system crash to save a record of system problems. This history is stored in the Small Dump Directory which is located in the %SystemRoot%\Minidump file path.
        - **To configure a small memory dump file, run the following command using the cmd utility**:

Wmic recoveros set **DebugInfoType = 3**

- Alternatively, this option can be configured in the registry:

Set the **CrashDumpEnabled** DWORD value to **3**

- To set a folder as the Small Dump Directory, use the following command line:

> Wmic recoveros set MiniDumpDirectory = <folderpath>

- Alternatively, the directory option can be set in the registry:

> Set the MinidumpDir Expandable String Value to <folderpath>

- Complete memory dump: The option records the contents of system memory when the computer stops unexpectedly. This option isn't available on computers that have 2 or more GB of RAM. If you select this option, you must have a paging file on the boot volume that is sufficient to hold all the physical RAM plus 1 MB. The file is stored as specified in %SystemRoot%\Memory.dmp by default. The extra megabyte is required for a complete memory dump file because Windows writes a header in addition to dumping the memory contents. The header contains a crash dump signature and specifies the values of some kernel variables. The header information doesn't require a full megabyte of space, but Windows sizes your paging file in increments of megabytes.

    - To configure a complete memory dump file, run the following command using the cmd utility:

> wmic recoveros set DebugInfoType = 1

- Alternatively, a complete memory dump file can be configured in the registry:
  Set the CrashDumpEnabled DWORD value to 1

- To set a memory dump file, use the following command line:

> wmic recoveros set DebugFilePath = <folderpath>

- Alternatively, the memory dump file can be set in the registry:
  Set the DumpFile Expandable String Value to <folderpath>
- To indicate that the system should not overwrite kernel memory dumps or other complete memory dumps, which may be valuable for troubleshooting system problems, use the command:

wmic recoveros set OverwriteExistingDebugFile = 0

- Alternatively, the overwrite setting can be turned off in the registry:
    - Set the Overwrite DWORD value to 0
- Peak usage or expected peak usage of the system commit charge - The system commit limit is the total of RAM plus the amount of disk space reserved for paging files. The system commit charge must be equal to or less than the system commit limit. If a page file is not in place, then the system commit limit is less than the system's RAM amount. The purpose of these measurements is to prevent the system from overpromising available memory. If this system commit limit is exceeded, Windows or the applications in use may stop functioning properly. So, it is a best practice to assess the amount of disk storage allocated to the page files periodically to ensure there is sufficient space for what the system needs during peak usage. It is fine to reserve 128 GB or more for the page files, if there is sufficient space on the hard drive to dedicate a reserve of this size. However, it might be a waste of available storage space if the system only needs a small fraction of the reserved disk space. If disk space is low, then consider adding more RAM, more hard drive storage, or offload non-system files to network or cloud storage.

- Space needed to offload data from RAM - Page files can serve to store modified pages that are not currently in use. This keeps the information easily accessible in case it is needed again by the system, without overburdening RAM storage. The modified pages to be stored on the hard drive are recorded in the \Memory\Modified Page List Bytes directory. If the page file is not large enough, some of the pages added to the Modified Page List Bytes might not be written to the page file. If this happens, the page file either needs to be expanded or additional page filles should be added to the system. To assess if the page file is too small, the following conditions must be true:
    - \Memory\Available MBytes indicates more physical memory is needed.
    - A significant amount of memory exists in the modified page list.
    - \Paging Files(*)% Usage (existing page files) are almost full.

Supplemental reading for Remote Connections in Windows

For more information about managing shared resources in Windows, check out the link here.

Remote Connections in Windows
Connecting securely to remote machines is an important task for deploying services. Secure Shell (SSH) was developed in the 1990s to address this issue. This reading will cover what SSH is, the features it enables, and common SSH clients and their key features in Windows.

SSH
Secure Shell (SSH) is a network protocol that gives users a secure way to access a computer over an unsecured network. SSH enables secure remote access to SSH-enabled network systems or devices and automated processes. It also allows for secure remote access to transfer files, use commands and manage network infrastructure.

OpenSSH
OpenSSH is the open-source version of the Secure Shell (SSH) tools used by administrators of Linux and other non-Windows for cross-platform remote systems management. OpenSSH has been added to Windows (as of autumn 2018) and is included in Windows Server and Windows client.

Common SSH Clients
An SSH client is a program that establishes secure and authenticated SSH connections to SSH servers. The following common SSH clients are Windows compatible:

PuTTY is a terminal emulator and the inspiration for all subsequent remote access systems.
- Features: This tool offers Telnet, SSH, Rlogin (A remote login tool for use with UNIX-based machines on your network), and raw socket connections plus Secure File Transfer Protocol (SFTP) and Secure Copy Protocol (SCP) for file transfers between two hosts.
- Protocols: SCP, SSH, Telnet, rlogin, and raw socket connection.

SecureCRT is a remote access system available for macOS, Linux, iOS, and Windows.
- Features: It offers terminal emulation and file transfer through an SSH tunnel. It enables connections through many protocols and has usability features like tabbed sessions and customizable menus.
- Protocols: SSH1, SSH2, Telnet, and Telnet/SSL

SmarTTY is a free SSH client with a multi-tabbed interface to allow multiple simultaneous connections.
- Features: This tool includes SCP capabilities for file transfers. It also includes usability features like auto-completion, file panel, and package management.
- Protocols: SSH and SCP

mRemoteNG is a remote desktop system with a tabbed interface for multiple simultaneous connections.
- Features: The system enables connections with Remote Desktop Protocol (RDP), Telnet (two-way text communication via virtual terminal connections), Rlogin, Virtual Network Computing (VNC, a graphics-based desktop sharing system), and SSH.
- Protocols: RDP, VNC, SSH, Telnet, HTTP/HTTPS, rlogin, Raw Socket Connections, Powershell remoting

MobaXterm is a remote access system built for Unix and Linux, and Windows.
- Features: Features include an embedded X server (a graphical interface akin to windows), X11 forwarding (a way to run applications over a remote connection), and easy display exportation to let X11 applications know which screen to run on.
- Protocols: SSH, X11, RDP, VNC

Key Takeaways
Secure Shell(SSH) is a way to securely connect two remote machines over an unsecured network.
- You can use SSH to remotely control, transfer files from, and manage network resources for SSH-enabled clients.
- OpenSSH is an open-source version for cross-platform management.
- There are many common Window-compatible SSH clients with various features to fit any need, including PuTTY, SecureCRT, SmarTTY, mRemoteNG, and MobaXterm.

Resources
- Download PuTTY
- Download SecureCRT
- Download SmarTTY
- Download mRemoteNG
- Download MobaXterm

Thought Mode

Microsoft's notes on PowerShell: https://learn.microsoft.com/en-us/shows/getting-started-with-microsoft-powershell/

Github "reddit" knowledgebase of PowerShell:
https://github.com/PowerShell/PowerShell

> cd ..\documents

- .. = go up one directory
- \documents = enter the **Documents** folder from there

> copy ~"\videos\IT Lessons" "c:\users\dave\OneDrive\IT Class Notes" -Recurse -Verbose

- Copies all files/folder as well as the original folder
- Demonstrates all actions done to perform command

> rm c:\users\dave\documents\random.txt -force

- Tells the system we *for sure* want to remove a potentially important file/directory

Useful Prompting

> get-help ls } Gives a summary of the command's parameters, in this case, the summary for ls

> get-help ls -full} Gives more details, and examples of how to use the command

> ls -force (location)} -Force parameter will show hidden and system files with ls

> history - Used to look up previously entered commands, which can be selected for repeated use

> Compress-Archive - Used to compress or zip files into a compressed file. Maximum 2GB. For more info, check out this link: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.archive/compress-archive?view=powershell-7.5&viewFallbackFrom=powershell-5.0

The Compress-Archive cmdlet ignores hidden files and folders when creating or updating the archive file. On non-Windows machines, this includes files and folders with name that begins with the period (.) character.

To ensure hidden files and folders are compressed into the archive, use the .NET API instead.

Example of Installing a Package Manager:

> Install-Package -Name sysinternals	**Installs a package named "sysinternals"**
> Get-Package -name sysinternals	**Confirms the package is installed**

> clear; Get-Process | Sort CPU -descending | Select -first 10 -Property ID,ProcessName,CPU

>