**Windows Update Download and Patching Scripts Documentation**

This page provides documentation for two PowerShell scripts: DownloadWindowsUpdates.ps1 and PatchWindowsImage.ps1. These scripts automate the process of downloading Windows updates from the Microsoft Update Catalogue and patching a Windows installation image (install.wim) offline using DISM. The scripts are designed for use in IT environments to create updated Windows installation media.

The scripts are signed by Marinus van Deventer and are compatible with PowerShell 5.1+, with optimal performance in PowerShell 7.4 (reference: <https://learn.microsoft.com/en-us/powershell/module/?view=powershell-7.4>).

**DownloadWindowsUpdates.ps1**

**Overview**

This script downloads Windows updates from the Microsoft Update Catalogue using the MSCatalogLTS module. It installs or updates the module if needed, searches for updates based on queries, filters by architecture, downloads the files to a month-year subfolder, and logs the process.

**Prerequisites**

* PowerShell 7+ for best performance (though compatible with 5.1+).
* Administrative privileges to install modules.
* The MSCatalogLTS module (automatically installed/updated by the script).
* Internet access to the Microsoft Update Catalogue.

**Parameters**

* **SearchQueries** (Mandatory, String[]): Array of search strings for updates (e.g., @("2025-08 Cumulative Update for Windows 11 Version 24H2 for x64-based Systems")).
* **DownloadPath** (Mandatory, String): Base directory to save downloaded updates (subfolder with monthyear will be created; created if not exists).
* **Architecture** (String): Filter by architecture (all, x64, x86, arm64; default: x64).
* **IncludePreview** (Switch): Include preview updates (default: $false).
* **LogFile** (String): Path to the log file (default: $PSScriptRoot\logs\DownloadUpdates.log).
* **StrictSearch** (Switch): Use strict (exact match) searching (default: $true).

**Usage Example**

text

.\DownloadWindowsUpdates.ps1 -SearchQueries @("2025-08 Cumulative Update for Windows 11 Version 24H2 for x64-based Systems") -DownloadPath "C:\Updates" -Architecture "x64"

**Changelog**

* v1.4 - Updated default LogFile to use script's current folder ($PSScriptRoot\logs\DownloadUpdates.log) per new instructions.
* v1.3 - Updated .EXAMPLE to use "2025-08 Cumulative Update for Windows 11 Version 24H2 for x64-based Systems" as the search query.
* v1.2 - Added check/create for DownloadPath if not exists; check module version before updating; updated return object.
* v1.1 - Added automatic creation of monthyear subfolder (e.g., 0825) under DownloadPath; updated return object.
* v1.0 - Initial version.

**How it Works**

1. **Initialization**:
   * Checks PowerShell version and warns if below 7.
   * Creates the log directory if needed.
   * Creates the DownloadPath if it does not exist.
   * Installs or updates the MSCatalogLTS module, then imports it.
2. **Subfolder Creation**:
   * Generates a month-year string (e.g., 0825 for August 2025).
   * Creates the subfolder under DownloadPath.
3. **Main Logic**:
   * Loops through each search query.
   * Searches the catalogue using Get-MSCatalogUpdate with parameters.
   * Selects the latest update.
   * Downloads it to the subfolder using Save-MSCatalogUpdate.
   * Logs progress and results.
4. **Output**:
   * Returns a PSCustomObject with status, downloaded updates, log file, and download path.

**Troubleshooting**

* **No updates found**: Ensure the search query is accurate (e.g., include KB number or exact title). Use web search for latest KB numbers.
* **Module installation fails**: Check internet access or run as administrator.
* **Path issues**: Ensure DownloadPath is writable; script creates it if missing.

**PatchWindowsImage.ps1**

**Overview**

This script patches a Windows image (install.wim) offline using DISM by adding update packages. It copies the source folder with a month-year suffix, mounts the image, adds .msu and .cab packages in order (SSU first), commits changes, and unmounts. It handles errors and logs progress.

**Prerequisites**

* PowerShell 5.1+ (compatible, but 7+ recommended).
* Administrative privileges (required for DISM).
* DISM tool (built-in on Windows 11).
* The source folder must contain a valid install.wim.

**Parameters**

* **SourceFolder** (Mandatory, String): Path to the source folder containing \sources\install.wim.
* **UpdatesFolder** (Mandatory, String): Path to the folder containing .msu or .cab update files.
* **MountDir** (Mandatory, String): Directory to mount the image (created if not exists).
* **ImageIndex** (Int): Index of the image edition to patch (default: 3 for Enterprise).
* **LogFile** (String): Path to the log file (default: $PSScriptRoot\logs\PatchWim.log).
* **ContinueOnPackageError** (Switch): Continues adding packages if one fails (default: $false).

**Usage Example**

text

.\PatchWindowsImage.ps1 -SourceFolder "C:\Windows-11-24H2" -UpdatesFolder "C:\Updates\0825" -MountDir "C:\Mount" -ImageIndex 3

**Changelog**

* v1.6.2 - Updated default LogFile to use script's current folder ($PSScriptRoot\logs\PatchWim.log) per instructions.
* v1.6.1 - Fixed op\_Addition error by forcing $updateFiles to be an array using @() on Get-ChildItem for .msu files, preventing issues when single file is found.
* v1.6 - Improved readability of $patchedRoot derivation by adding spaces around concatenation operators.
* v1.5 - Fixed DISM argument passing by using explicit argument arrays to handle paths with spaces and prevent Error 87 (unknown option).
* v1.4 - Changed parameter from WimFile to SourceFolder; updated derivation, validation, and example accordingly.
* v1.3 - Added check/create for MountDir if not exists; removed ValidateScript for MountDir parameter.
* v1.2 - Added sorting of updates by ReleaseType (SSU priority) and CreationTime ascending using DISM /Get-PackageInfo; updated progress.
* v1.1 - Added copy of source root folder with monthyear suffix (e.g., \_0825); changed default ImageIndex to 3 for Enterprise; updated return object.
* v1.0 - Initial version.

**How it Works**

1. **Initialization**:
   * Checks for admin privileges.
   * Creates the log directory if needed.
   * Creates the MountDir if it does not exist and ensures it is empty.
   * Copies the SourceFolder to a new folder with month-year suffix.
2. **Mount the Image**:
   * Uses DISM /Mount-Wim to mount the specified image index to MountDir.
3. **Get and Sort Update Files**:
   * Collects .msu and .cab files from UpdatesFolder.
   * Analyzes each with /Get-PackageInfo to determine ReleaseType and CreationTime for sorting (SSU first).
4. **Add Packages**:
   * Adds sorted packages with /Add-Package.
   * Logs progress and handles errors (continues if ContinueOnPackageError is true).
5. **Unmount and Commit**:
   * Uses DISM /Unmount-Wim /Commit to save changes.
6. **Cleanup and Output**:
   * Discards if error occurs.
   * Returns status object with patched folder and WIM path.

**Troubleshooting**

* **Mount failure (Error 87)**: Ensure paths have no extra quotes; script uses array args to handle spaces.
* **Package add failure**: Check DISM log (C:\Windows\Logs\DISM\dism.log) for details; ensure SSU is added first.
* **No updates found**: Verify UpdatesFolder has valid .msu/.cab files.
* **op\_Addition error**: Fixed in v1.6.1; ensure using latest script.

**Manual Guide: Download and Patch Windows Updates**

If you prefer not to use the scripts, follow these manual steps to download updates, patch the install.wim, and create a bootable ISO.

**Step 1: Download Updates Manually**

1. Open a browser and navigate to the Microsoft Update Catalogue.
2. Search for the desired KB number or title (e.g., "2025-08 Cumulative Update for Windows 11 Version 24H2 for x64-based Systems").
3. Filter by architecture (x64) and version (24H2).
4. Download the .msu or .cab file to a folder (e.g., C:\Updates\0825).
5. Repeat for any SSU or other updates.

**Step 2: Patch install.wim Manually Using DISM**

1. Open Command Prompt as Administrator.
2. Mount the image:

text

DISM /Mount-Wim /WimFile:C:\Windows-11-24H2\sources\install.wim /Index:3 /MountDir:C:\Mount

1. Add packages (SSU first, then CU):

text

DISM /Image:C:\Mount /Add-Package /PackagePath:C:\Updates\0825\windows11.0-kb<SSU>.msu

DISM /Image:C:\Mount /Add-Package /PackagePath:C:\Updates\0825\windows11.0-kb<CU>.msu

1. Unmount and commit:

text

DISM /Unmount-Wim /MountDir:C:\Mount /Commit

**Step 3: Create Bootable ISO Manually Using Oscdimg**

1. Ensure Oscdimg is in PATH or use full path.
2. Run:

text

oscdimg.exe -m -o -u2 -udfver102 -bootdata:2#p,0,e,b"C:\Windows-11-24H2\_0825\etfsboot.com"#pEF,e,b"C:\Windows-11-24H2\_0825\efi\microsoft\boot\efisys.bin" -lWindows11\_24H2 C:\Windows-11-24H2\_0825 C:\temp\scripts\Windows-11-24H2\_0825.iso

Reference: Latest DISM docs and Oscdimg options. If issues, check DISM log at C:\Windows\Logs\DISM\dism.log.