Updating a Windows Image with Windows Updates

By Hannes Sehestedt
X64 Only Edition
Last Update: June 16, 2025

****
* Updating a Windows Image with Windows Updates *
****

****
* Obtain and Organize Updates *
****

First, you will need to obtain updates from the **Microsoft Update Catalog**. Search for Windows updates that apply to your version of Windows. For example, in the search box, type the following line. Include the quotes as shown:

"Windows 11 version 24H2" x64

To narrow your search to updates released in a specific time period, try including a month and year as in this example:

"Windows 11 version 24H2" x64 2025-06

The Microsoft Update Catalog is located here:

Microsoft Update Catalog

Note that some updates such as the **Safe OS Dynamic Update** and the **Setup Dynamic Update** are not released every month. In fact, it's possible that there may be no such update for your version of Windows. You should determine when your build of Windows was released and search back in time to that date for these updates. For example, looking at the update history for Windows 11 23H2 (see link below) I see that the first updates were available in October, 2022. So, in the update catalog I would work my way back in time looking for each type of update and only grabbing the most recent one of each update type. I would go back as far as October of 2022. **EXCEPTION: OOBE ZDP** updates are **NOT** cumulative. As a result, you should download **ALL** available updates of this type, not just the most recent. This type of update is rare.

See the following link for Windows 10 and 11 update history. Note that this does not list the individual updates:

Windows 10 Update History

Windows 11 Update History

NOTE: You should search for and download the latest version of each of the following updates:

- Windows LCU/SSU (Latest Cumulative Update. **See notes below**)
- Standalone SSU (Very rare!)
- Windows Safe OS Dynamic Update
- Windows Setup Dynamic Update
- Cumulative Update for .NET Framework 3.5, 4.8, and 4.8.1
- Windows OOBE ZDP Updates. Rare. For OOBE ZDP updates, download **ALL** of these updates, not just the most recent because these are **NOT** cumulative.

IMPORTANT: You need to understand how the combined **Latest Cumulative Update (LCU)** and **Servicing Stack Update (SSU)** are distributed.

Prior to Windows 11 24H2, the LCU/SSU was a cumulative update. In other words, only the latest update was needed because it contains all previous updates as well as the latest updates.

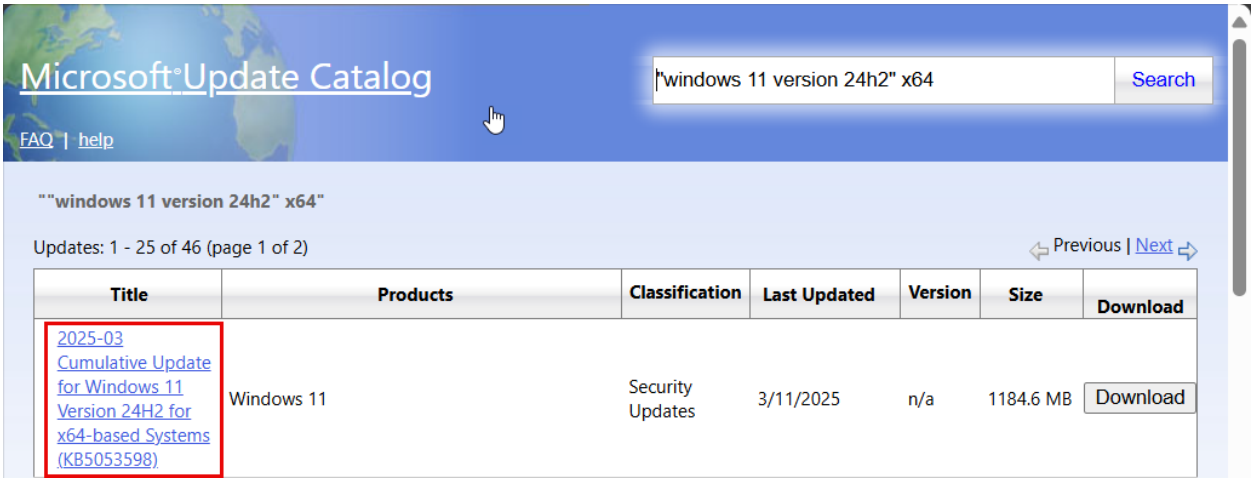
With Windows 11 24H2+ this has changed. When you visit the Microsoft Update Catalog and select a cumulative update for download, you will see that either one single file is presented for download, or two files will be presented.

If only one file is presented, then this is a **checkpoint** update that is very much the same as a previous cumulative update and this will be the only file needed for the LCU/SSU category. If two files are shown, you will need to download both. One of these is a **checkpoint** update and the other is a **differential** update that contains all updates released since that last checkpoint update.

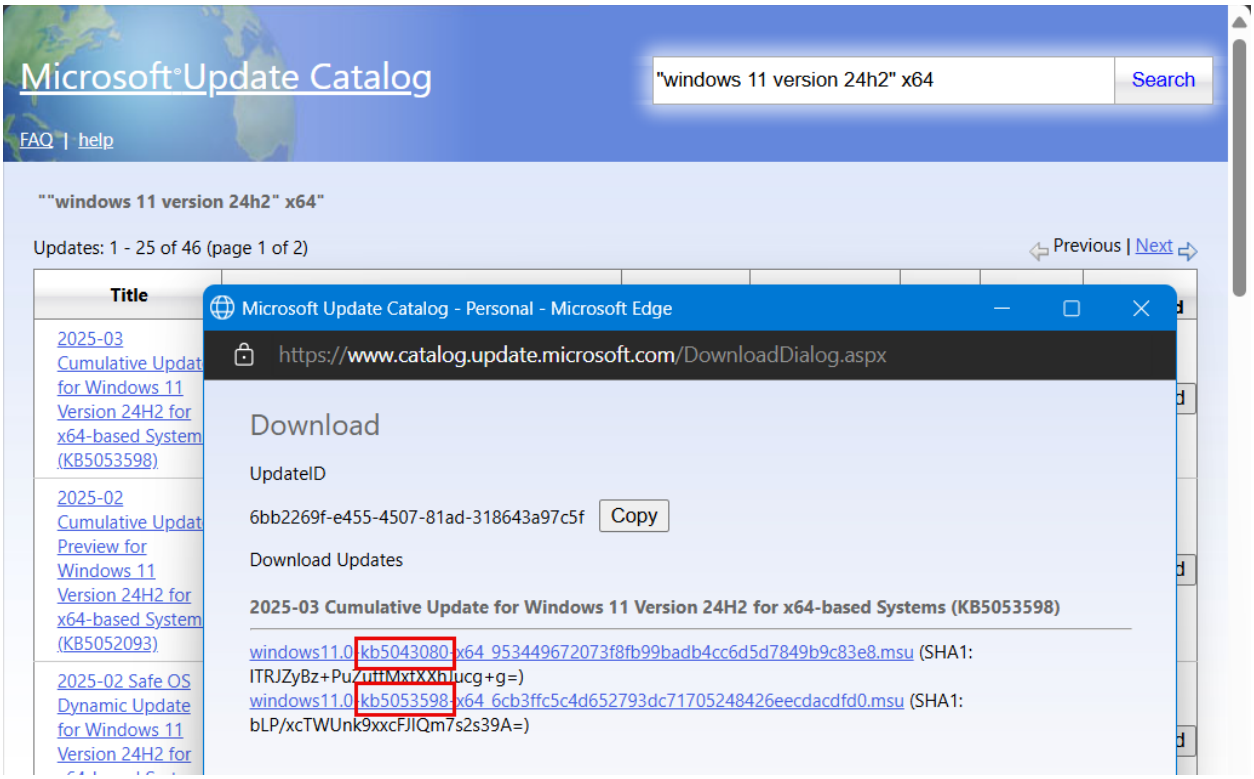
End of Important Note

The images below show how the checkpoint and differential updates can be identified.

First, let's visit the Microsoft Update Catalog. Note here the latest cumulative update:



If we click on the **Download** button for this update, we will see the following:



Note that the above image shows two downloads for the cumulative update. If we now examine the Windows Update History, we can see that the older of the updates originates back in September of 2024 and that it corresponds with build 26100.1742:

This tells us that KB5043080 will upgrade any build of 24H2 to build 26100.1742. If you examine every update following that update (at least up to June, 2025), you will find that they all include KB5043080 plus a newer differential update. So, if you have a build earlier than 1742, then the checkpoint update should be applied first to bring it to build 1742. For build 1742 or newer, only the differential update needs to be applied to bring the image current since all updates for the checkpoint update are already included.

NOTE: If unsure, you can safely apply both the checkpoint update and the differential update (if available). The system will automatically handle the updates correctly if the checkpoint updates are already included in your image.

Save the updates to a folder. In this example, we are saving to **C:\WinUpdates**.

Create a directory structure like this:

```
C:\WinUpdates
      LCU               < Place the Latest Cumulative Update in this folder
      SSU               < If a standalone SSU exists, add it here
      Other             < Other updates such as .NET updates and OOBE ZDP updates
      SafeOS_DU         < For the SafeOS (WinRE) Dynamic Update
      Setup_DU          < For the Setup Dynamic Update
      PE_Files          < Files to add to the boot.wim (Win PE)
```

NOTE: The folder "PE_Files" is not for Windows update files. This is for custom scripts or other files that you may want to add to the boot.wim. Most people will never use this so you can safely omit this folder entirely if you wish to do so.

NOTE: When downloading updates from the Microsoft Update Catalog, the **Setup DU** will show a description in the **Products** heading such as **Windows 10 and later Dynamic Update**. The **SafeOS DU** will show **Windows Safe OS Dynamic Update**. Make sure to check both the "Title" and "Products" entries to make sure that you are downloading the appropriate updates for your image(s). Note that the structure above is only a suggestion, however, that is what I will use in the examples that follow.

Updates are cumulative. This means that the latest update contains all the previous updates. As a result, you should apply the latest update of any given kind to Windows. **DO NOT** apply multiple updates of the same kind.

EXCEPTION #1: As noted above, if the most recent LCU/SSU shows two files, you should download both. Do this only when two updates are listed under the same update link in the Update Catalog.

EXCEPTION #2: The **OOBE ZDP** updates are not cumulative. Download **ALL** of the OOBE ZDP updates that are available for your version of Windows.

EXCEPTION #3: In the **Other** folder, you can place multiple different kinds of updates. Typically, I only place the latest .NET update in here but other things could potentially be placed in here as well. The **PE_Files** folder can also have multiple different generic files, but you will likely never use this folder.

Note: When a newer version of an update becomes available, it is okay to apply this to a Windows image that has had a previous update applied. The new update will intelligently update only those components that need to be updated.

The following table shows the order in which updates should be applied. The number in parenthesis in the table relates to language items and features on demand which we don't deal with in this document. You can simply skip those items.

	WinRE (winre.wim)	Main OS (install.wim)	WinPE (boot.wim)	New media
Add SSU Dynamic Update	1	9	17	
Add Lang Pack	(2)	(10)	(18)	
Add Localized Optional Packages	(3)		(19)	
Add Font Support	(4)		(20)	
Add TTS Support	(5)		(21)	
Update Lang.ini			(22)	
Add Features On Demand		(11)		
Add Safe OS Dynamic Update	6			
Add Setup Dynamic Update				26
Add setup.exe from WinPE				27
Add boot manager from WinPE				28
Add LCU		12	23	
Clean Image	7	13	24	
Add Optional Components		14		
Add .NET and .NET Updates, OOBE ZDP, Other		15		
Export Image	8	16	25	

NOTE: Microsoft now ships the Latest Cumulative Update (LCU) and Servicing Stack Update (SSU) as a single combined package in the LCU update. The SSU should be extracted from the combined package in order to install it as is shown in the steps below. However, there may be rare cases where a separate Standalone SSU is still published. If that happens, apply that SSU first before the combined LCU / SSU.

```
************************
* Applying the Updates *
************************
```

IMPORTANT:

When updating Windows images, if you apply the latest updates every month when they are released, I suggest that you apply the updates to a clean Windows image rather than applying updates to an already previously updated image. The reason for this is that certain types of updates such as enabling NetFX3, can have pending operations which will cause problems for updates being applied thereafter. In practice, I've never encountered a problem with this, but it is something to be aware of, especially if you add features outside of the scope of this document.

Make sure you have the **Windows ADK** installed (only the **Deployment Tools** are needed). If not, download and install it now as described in Prerequisites.

Go to **Start** > **All Apps** > **Windows Kits** and open **Deployment and Imaging Tools Environment** in elevated mode (as Admin).

You will see a command prompt with a long path. Type **cd ** and hit **Enter** to go back to the root. This makes the prompt shorter and the command line less cluttered.

If you have any difficulties and need to start over, in particular, if you have an image mounted with DISM, perform a cleanup as in this example:

First, you can check to see if there are any open DISM mounts like this:

```
Dism /Get-MountedImageInfo
```

Then, if there are any open mounts run these commands. Not needed if there are no open mounts.

```
DISM /Unmount-Image /MountDir:"C:\Project\Mount" /Discard
DISM /Cleanup-WIM
DISM /Cleanup-Mountpoints
C:\windows\system32\takeown /f "C:\Project\*.*" /r /d y
icacls "C:\Project\*.*" /T /grant %username%:F
```

Finally, clear the contents of the directories and make sure to empty the recycle bin as this mount can still cause difficulties if stuck open in the recycle bin.

IMPORTANT: If you find that the above does not work and you cannot delete the folder where an image was mounted, log off, log back on, and then try the above again. If you still have difficulties, make sure you have cleaned out the recycle bin, reboot the system and try the above commands one more time.

Start by creating a directory structure for your project as shown below. This is an example, use whatever you want. This is the structure that I will use:

```
C:\Project
      Assets            < Will be used to store files once updated
      ISO_Files         < Used to store the original Windows files to be updated
      Logs              < Used to save log files from the commands being run
      Mount             < Used for mounting the main OS image (install.wim)
      Scratch           < Used by the DISM command
      SSU               < Used to save the extracted SSU
      Temp              < Used by the DISM command
      WIM               < Used to store the updated install.wim
      WINPE             < Used to hold the original boot.wim file before updating
      WINRE             < Used to hold the original winre.wim file before updating
      WINPE_MOUNT       < Used for mounting the boot.wim
      WINRE_MOUNT       < Used for mounting the winre.com
```

Create a directory to place Windows Updates into. Example: **C:\WinUpdates**.

NOTE: In this example, I'm going to create a directory structure for the Windows updates that looks like this:

```
C:\WinUpdates
      LCU               < Place LCU/SSU Checkpoint and Differential updates here
      SSU               < If a standalone SSU exists, add it here
      Other             < Other updates such as .NET updates and OOBE ZDP updates
      SafeOS_DU         < For the SafeOS (WinRE) Dynamic Update
      Setup_DU          < For the Setup Dynamic Update
      PE_Files          < Files such as scripts to add to the boot.wim (Win PE)
```

TIP: All folders should have only a single update with the exceptions that were noted earlier. The **Other** and **PE_Files** folders can contain multiple files.

In the sections that follow, we will perform these operations:

- Update the recovery image (WinRE.wim)
- Update the main OS (install.wim)
- Update WinPE and Windows Setup (Boot.wim)
- Install additional updates (.NET updates, OOBE ZDP updates, etc.)

Extract the contents of a Windows ISO image or copy the files from a bootable Windows flash drive to **C:\Project\ISO_Files.**

Get the Windows update package(s) from the Microsoft Update Catalog. For example, grab the latest cumulative update listed for Windows. Place the **.msu** file update to a folder, for example, **C:\WinUpdates\LCU\windows10.0-kb4016871-x64_27dfce9dbd92670711822de2f5f5ce0151551b7d.msu.**

Tip: Rename the file to make it easier to work with. Example, **KB4016871.msu.**

Make sure to grab all the available updates and put them in the folders noted earlier. It's perfectly fine if some update types have no updates available.

```
********************************
* Summary of the Update Process *
********************************
```

There are three files that we need to update from the Windows distribution:

install.wim	Holds the main Windows OS
boot.wim	Holds WinPE - Windows Preinstallation Environment
winre.wim	Holds WinRE - Windows Recovery Environment

The first component that we want to update is the **winre.wim**, but this file is located WITHIN the **install.wim**. As a result, we will first need to mount the **install.wim** so that we can retrieve the **WinRE.wim**. Think of this as being like having a ZIP file inside of another ZIP file. Once the **winre.wim** is updated, we will update the **install.wim**, and then, finally, the **boot.wim**.

In addition, we will update other files outside of these WIM files.

At this time, we need to sync a few files that may be mismatched.

With all those pieces updated, we will gather all the files from your original Windows distribution media and replace the original files that we updated with those updated versions.

When that is done, you can create a new ISO image or media with the updated Windows files.

```
******************
* End of Summary *
******************
```

Open an elevated PowerShell or command prompt. Enter the following command to check what editions are included in the main OS image:

```
DISM /Get-WimInfo /WimFile:"C:\Project\ISO_Files\Sources\install.wim"
```

Note the index number of your selected edition. In this example we are using an index number of 6 which, in my image, represents Windows 11 Pro.

Mount the image of your preferred Windows edition using its index number, index 6 in this example:

```
DISM /mount-image /imagefile:"C:\Project\ISO_Files\sources\install.wim"
/index:6 /mountdir:"C:\Project\Mount"
```

By mounting the **install.wim**, we extract the contents of the WIM to another location (**C:\Project\Mount**). We can alter, add or remove files here, then when we dismount, all the changes are written back to the original file. Think of it much like unzipping a ZIP file, changing some files or adding / removing files, then re-zipping.

```
********************
* Update WinRE.wim *
********************
```

If you have none of these updates skip this section on updating the **WinRE.wim**: A Standalone SSU, an LCU, or a Safe OS Dynamic Update.

From the directory where the **install.wim** is mounted, copy the **winre.wim** file. In this example you would copy **winre.wim** from **c:\Project\Mount\Windows\System32\Recovery** to **C:\Project\WinRE**

```
copy /B C:\Project\Mount\Windows\System32\Recovery\winre.wim
C:\Project\WinRE
```

Mount index number 1 in the **winre.wim** (winre.wim has only one index)

```
DISM /mount-image /imagefile:"C:\Project\WinRE\WinRE.wim" /index:1
/mountdir:"C:\Project\WinRE_Mount"
```

Although the SSU and LCU are now combined into the LCU, there could be rare cases where a Standalone SSU published. In order to handle this, you should first apply the Standalone SSU, if it exists.

```
DISM /Add-Package /Image:"C:\Project\WinRE_Mount"
/PackagePath="C:\WinUpdates\SSU" /LogPath="C:\Project\Logs\dism.log"
```

The following commands will extract the SSU (if one exists) and inject the SSU and Safe OS Dynamic Update into WinRE. If there is no LCU / SSU Update or Safe OS Dynamic Update, skip that item. The use of **/LogPath** here and in all further commands is optional.

NOTE: The "f:" in the following command is NOT a drive letter. Don't change it.

```
expand "C:\WinUpdates\LCU\*.MSU" /f:"SSU*.cab" "C:\Project\SSU"

DISM /Add-Package /Image:"C:\Project\WinRE_Mount"
/PackagePath="C:\Project\SSU" /LogPath="C:\Project\Logs\dism.log"

DISM /Add-Package /Image:"C:\Project\WinRE_Mount"
/PackagePath="C:\WinUpdates\SafeOS_DU" /LogPath="C:\Project\Logs\dism.log"
```

TIP: Be aware that there are several ways to add packages to a WIM file. You can specify just a single file to inject into the WIM, you can specify a list of updates to inject, or you can point to a folder and have all updates located in the folder injected into the WIM. Here are some examples illustrating this:

Example: Adding a cumulative update to a mounted image:

```
DISM /Add-Package /Image:"C:\Project\Mount"
/PackagePath="C:\WinUpdates\kb4016871.msu"
/LogPath="C:\Project\Logs\dism.log"
```

Example: add multiple updates by specifying updates (this is one long command):

```
DISM /Add-Package /Image:"C:\Project\Mount"
/PackagePath="C:\WinUpdates\kb00001.msu"
/PackagePath="C:\WinUpdates\kb00002.msu"
/PackagePath="C:\WinUpdates\kb00003.msu" /LogPath="C:\Project\Logs\dism.log"
```

Example: Adding multiple updates in a folder by just pointing to the folder:

```
DISM /Add-Package /Image:"C:\Project\Mount" /PackagePath="C:\WinUpdates"
/LogPath="C:\Project\Logs\dism.log"
```

Note that for several of the update types, the above is completely unnecessary because you only need the latest version of those updates, meaning that only one will be installed. But this may be helpful for those categories where multiple updates may need to be installed.

END TIP

The next command performs a cleanup operation:

```
DISM /Cleanup-Image /Image:"C:\Project\WinRE_Mount" /StartComponentCleanup
```

Now, unmount the image:

```
DISM /Unmount-Image /MountDir:"C:\Project\WinRE_Mount" /Commit
```

Once unmounted, the updates that you applied will have been committed to the **winre.wim** file located in the **C:\Project\WinRE** folder.

Export index 1 from the **WinRE.wim** (there is only one index in **WinRE.wim**):

```
DISM /Export-Image /SourceImageFile:"C:\Project\WinRE\WinRE.wim"
/SourceIndex:1 /DestinationImageFile:"C:\Project\Assets\WinRE.wim"
```

A note about the "export" operation: When we performed the cleanup operation above, it marks duplicate files such as those replaced by updates for deletion. It is not until an export is performed that this data gets cleaned up. This is why you perform an export operation rather than simply copying the image file from the source to the destination when you are done updating it.

****
* Update the Main OS (install.wim) *
****

Updating the main OS is similar to updating the **WinRE.wim**. Here is an overview of the differences:

- As before, we will inject updates and then perform a cleanup EXCEPT that we will install optional components **AFTER** the cleanup because some actions such as enabling the NetFX3 capability (not covered in this document) require post installation actions. This would cause a failure if a cleanup were performed with those actions pending. You can run this command against a mounted image and look for "**State : InstallPending**" to see if such a state exists:

  ```
  PowerShell "Get-WindowsCapability -path 'C:\Project\Mount'"
  ```

As an example, if NetFX3 is enabled, you would see:

```
Name  : NetFX3~~~~
State : InstallPending
```

- We will copy the **WinRE.wim** to the **install.wim** before we perform the cleanup.

NOTE: If you are choosing **NOT** to update the WinRE and WinPE components (for example, if no updates were available for these), then simply skip any steps later that ask you to copy those files (winre.wim and boot.wim).

For the **install.wim**, the components that we will install will be the Standalone SSU, the SSU from the combined LCU / SSU package, the LCU, and finally optional components such as the .NET cumulative update, OOBE ZDP updates, etc. Again, recall that the SSU was previously extracted from the combined LCU / SSU package but we will first install the Standalone SSU if one exists.

Once again, I won't explain each command since we have already done so. I'll just show examples of the commands that need to be run. Recall that the **install.wim** is already mounted (that was just about the first thing that we did).

```
DISM /Add-Package /Image:"C:\Project\Mount" /PackagePath="C:\WinUpdates\SSU"
/LogPath="C:\Project\Logs\dism.log"
```

```
DISM /Add-Package /Image:"C:\Project\Mount" /PackagePath="C:\Project\SSU"
/LogPath="C:\Project\Logs\dism.log"
```

```
DISM /Add-Package /Image:"C:\Project\Mount" /PackagePath="C:\WinUpdates\LCU"
/LogPath="C:\Project\Logs\dism.log"
```

We will now copy the updated recovery image (**WinRE.wim**) to the mount location for the **install.wim**:

```
copy /B "C:\Project\Assets\WinRE.wim"
"C:\Project\Mount\Windows\System32\Recovery" /Y
```

Perform the cleanup:

```
DISM /Cleanup-Image /Image:"C:\Project\Mount" /StartComponentCleanup
/ResetBase /ScratchDir:"C:\Project\Temp"
```

IMPORTANT: When updating the install.wim, be aware that if the image was previously updated, it is possible that image cleanup may fail. This is because a cleanup cannot occur if there are pending operations. A common source of this issue is enabling NetFX3 in your image. If you do this, then you should apply future updates to a clean image created before you applied other updates or at least before you installed NetFX3. You will also note that we apply updates from the **Other** category **AFTER** we perform the cleanup. This is for the same reason - because certain updates in this category can leave pending operations. It is for this reason that I earlier suggested applying updates to a clean Windows image rather than one that that you have already updated previously. If you are updating only the items shown in this document and not adding things like NetFX3 then this is not going to be a problem.

If you want to check to see if your image has any pending operations, run this command now:

```
PowerShell "Get-WindowsCapability -path 'C:\Project\Mount'
```

As an example, if NetFX3 is enabled, you would see:

```
Name  : NetFX3~~~~
State : InstallPending
```

As an alternative, you could run the command below to filter and show only items with "InstallPending":

```
PowerShell "Get-WindowsCapability -path 'C:\Project\Mount' | Where-Object {
$_.State -eq 'InstallPending' }"
```

End of Important Note

****
* Update WinPE image (boot.wim) *
****

Updating the **boot.wim** is the same as for the **WinRE.wim**. The only differences are:

- The boot.wim contains 2 images (index 1 and 2) so we need to update both.
- WinPE is located on the original media in the **\sources** folder, so we will copy it from there.
- The updates we will inject are the Standalone **SSU**, the SSU from the LCU package, and the LCU. Recall that we have already extracted the SSU from the combined LCU / SSU package.

I won't explain the purpose of each command again; I'll just supply the commands here.

Start by copying **boot.wim** from **C:\Project\ISO_Files\Sources** to **C:\Project\WinPE**:

```
copy /B "C:\Project\ISO_Files\sources\boot.wim" "C:\Project\WinPE"
```

Continue with the following commands:

```
DISM /mount-image /imagefile:"C:\Project\WinPE\boot.wim" /index:1
/mountdir:"C:\Project\WinPE_Mount"

DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\WinUpdates\SSU" /LogPath="C:\Project\Logs\dism.log"

DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\Project\SSU" /LogPath="C:\Project\Logs\dism.log"

DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\WinUpdates\LCU" /LogPath="C:\Project\Logs\dism.log"

DISM /Cleanup-Image /Image:"C:\Project\WinPE_Mount" /StartComponentCleanup

DISM /Unmount-Image /MountDir:"C:\Project\WinPE_Mount" /Commit

DISM /Export-Image /SourceImageFile:"C:\Project\WinPE\boot.wim"
/SourceIndex:1 /DestinationImageFile:"C:\Project\Assets\boot.wim"
```

To update index 2 in the **boot.wim**, run the commands below. Take note that we reference **index:2** in the first command this time. In addition, note that we are not yet unmounting or exporting this second index because we will need to copy some files from the boot.wim later while it is still mounted.

```
DISM /mount-image /imagefile:"C:\Project\WinPE\boot.wim" /index:2
/mountdir:"C:\Project\WinPE_Mount"

DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\WinUpdates\SSU" /LogPath="C:\Project\Logs\dism.log"

DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\Project\SSU" /LogPath="C:\Project\Logs\dism.log"
```

```
DISM /Add-Package /Image:"C:\Project\WinPE_Mount"
/PackagePath="C:\WinUpdates\LCU" /LogPath="C:\Project\Logs\dism.log"

DISM /Cleanup-Image /Image:"C:\Project\WinPE_Mount" /StartComponentCleanup
```

```
************************
* Install Other Updates *
************************
```

Now we install the **other** updates (.NET cumulative update, OOBE ZDP updates, etc.). You can place all these updates in the **other** folder. The next command will install them all:

```
DISM /Add-Package /Image:"C:\Project\Mount"
/PackagePath="C:\WinUpdates\Other" /LogPath="C:\Project\Logs\dism.log"
```

We will not yet unmount and export the **install.wim** since we will need to copy some files from it while it is still mounted.

Some helpful commands (optional):

You can verify the packages that were installed with this command:

```
DISM /Get-Packages /image:"C:\Project\Mount"
```

The above command points to a folder before you dismount and commit the changes.

Enter the following command to check what editions are included in an image along with the associated index numbers:

```
dism /Get-WimInfo /WimFile:"C:\Project\Assets\install.wim"
```

```
**********************************
* Updating the Base Image, Syncing *
* Files, and Finalizing the Image  *
**********************************
```

For the "Setup Dynamic Update" we simply extract the contents to our media directly. To do so, run this command (replace **SetupDU.CAB** with the name of your file for the Setup Dynamic Update). Skip this command if you have no Setup Dynamic Update file. To apply the Setup Dynamic Update, run this command:

```
Expand "C:\Project\Setup_DU\SetupDU.CAB" -F:*
"C:\Project\ISO_Files\Sources"
```

NOTE: the **F:** is **NOT** a drive letter. Don't change this!

Inside both the **boot.wim** (index number 2) and the **install.wim** (the index number that you have been working with) are some files that are supposed to be duplicates of each other. It is possible that these files are now no longer identical and that they are out of sync. The commands below correct this.

To perform the synchronization, run these commands:

```
Copy /b /y "c:\project\winpe_mount\sources\setup.exe"
"c:\project\iso_files\sources\setup.exe"

Copy /b /y "c:\project\winpe_mount\sources\setuphost.exe"
"c:\project\iso_files\sources\setuphost.exe"

copy /b /y "c:\project\winpe_mount\windows\boot\efi\bootmgfw.efi"
"c:\project\iso_files\efi\boot\bootx64.efi"

copy /b /y "c:\project\winpe_mount\windows\boot\efi\bootmgr.efi"
"c:\project\iso_files\bootmgr.efi"
```

NOTE: If you want to add other files to be available during Windows setup, add the files to the **C:\Project\WinPE_Mount** directory now. When Windows setup starts, these files will be available on the Ramdrive (**X:**). As an example, you might want to do this if you need scripts to be available during setup. Please note that this is something that is **very rarely used** so you can skip this unless you have a very specific reason to copy files to this location.

Now, we will close the **boot.wim** and export it. Then we will do the same for the **install.wim**:

```
DISM /Unmount-Image /MountDir:"C:\Project\WinPE_Mount" /Commit

DISM /Export-Image /Bootable /SourceImageFile:"C:\Project\WinPE\boot.wim"
/SourceIndex:2 /DestinationImageFile:"C:\Project\Assets\boot.wim"
```

Note that when we export index number 2, because the destination file is the same as when we exported index 1 earlier, the export adds that image to the file. You now have an image file with both index 1 and 2 updated. Also, note that for the second index of the **boot.wim**, we mark it as bootable with the **/Bootable** switch.

Now we can unmount and export the **install.wim** file:

```
DISM /Unmount-Image /MountDir:"C:\Project\Mount" /Commit

DISM /Export-Image
/SourceImageFile:"C:\Project\ISO_Files\sources\install.wim" /SourceIndex:6
/DestinationImageFile:"C:\Project\Assets\install.wim"
```

When you export the **install.wim**, the new **install.wim** will have only one index (index number 1) because you have exported only 1 Windows edition - the one associated with index 6 in this example. So even if your original index number was 6, that edition of Windows will now be index 1 in the new image.

Finally, we need to copy the updated **boot.wim** file and the updated **install.wim** to the base image:

```
copy /B "C:\Project\Assets\boot.wim" "C:\Project\ISO_Files\Sources" /Y

copy /B "C:\Project\Assets\install.wim" "C:\Project\ISO_Files\Sources"
/Y
```

In the folder **C:\Project\ISO_Files** you now have all your files from your Windows Distribution with all the updates in place.

You can now follow the steps in Methods to Create Bootable Images, DVD, Drives to create bootable media from these files.

If you simply want to create a bootable ISO image, enter the following command. Be careful; there are spots where you might expect a space, but none is present:

```
oscdimg.exe -m -o -u2 -udfver102 -
bootdata:2#p0,e,bc:\project\iso_files\boot\etfsboot.com#pEF,e,bc:\project\iso
_files\efi\microsoft\boot\efisys.bin c:\project\iso_files c:\Win11PRO.iso
```

Replace the three occurrences of **c:\project\iso_files** with the correct path to where your files are located. Also, replace the **c:\Win11PRO.iso** at the end of command with the correct path and name for the output file that you want to save. If that path contains spaces, enclose it in quotes.

****
* End of Procedure for Updating Windows Image with Latest Updates *
****