
* Updating Windows 10 Image with Latest Windows Updates *

* Obtain and Organize Updates *

First, you will need to obtain updates. Here are resources for Windows updates. Search for Windows updates that apply to your version of Windows. For example, in the search box, type "Windows 10 version 20H2".

Go to Microsoft Update Catalog:

<https://www.catalog.update.microsoft.com/>

Also, see the following location for a list of updates:

<https://support.microsoft.com/en-us/help/4464619>

A list of Servicing Stack Updates (SSU) can be found here. Apply the latest SSU:

<https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/ADV990001>

Note that only the most recent update of each type is needed.

NOTE: I would suggest searching for and downloading the latest version of each of the following updates:

- Servicing Stack update
- Windows Safe OS Dynamic Update
- Windows Setup Dynamic Update
- Windows cumulative update
- Cumulative Update for .NET Framework 3.5 and 4.8
- Security Update for Adobe Flash Player (going away after Dec. 2020)
- Intel Microcode Updates (see Microsoft KB4589212 for 20H2) - Not for all CPUs

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

Create a directory structure like this:

```
C:\WindowsUpdates
  Microcode      < See notes below about the Microcode updates
  LCU            < Place the Latest Cumulative Update in this folder
  Other          < Other updates such as .NET updates, Adobe flash Updates
  SafeOS_DU     < For the SafeOS (WinRE) Dynamic Update
  Setup_DU      < For the Setup Dynamic Update
  SSU           < Servicing Stack Updates
  PE_Files      < Files to add to the boot.wim (Win PE)
```

NOTE: When downloading updates from the Microsoft Update Catalog, the Setup DU will show a description of "**Windows 10 Dynamic Update**". The SafeOS DU will show "**Safe OS Dynamic Update**". Note that the structure above is only a suggestion, however, that is what I will use in the examples that follow.

If you want to inject the CPU Microcode updates, place the update in the "Other" folder. The "Microcode" folder is used only for the installation of updates from a batch file as described [here](#). Be aware that Microcode updates do not apply to every CPU so consider leaving it in the "Microcode" folder and installing it manually on only those systems where it is needed.

The following table shows the order of operations. The cells in red in the table relate to language items and features on demand which we don't deal with in this document. If you want to see a sample of how to address those items, look at this blog from Microsoft:

<https://techcommunity.microsoft.com/t5/windows-it-pro-blog/updating-windows-10-media-with-dynamic-update-packages/ba-p/982477>

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

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

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

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 had an image mounted with DISM, unmount the image and 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...

```
dism /Unmount-Image /MountDir:"C:\Project\Mount" /Discard  
dism /Cleanup-WIM  
dism /cleanup-mountpoints
```

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 again.

END IMPORTANT NOTES

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  
  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 the Latest Cumulative Update in this folder  
  Other           < Other updates such as .NET updates, Adobe flash Updates  
  SafeOS_DU      < For the SafeOS (WinRE) Dynamic Update  
  Setup_DU       < For the Setup Dynamic Update  
  SSU            < Servicing Stack Updates  
  PE_Files       < Files such as scripts to add to the boot.wim (Win PE)
```

TIP: All folders should have only a single update. The "Other" and "PE_Files" folders can contain multiple files. This is because only one Cumulative Update is needed, and the same is true for other items such as the SSU - only the latest version of each is needed.

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

- Update WinRE.wim
- Update WinPE (Boot.wim)
- Update the main OS (install.wim)
- Install additional updates (.NET updates, Adobe Flash 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 10. Place the .msu file update to a folder, for example, **C:\WinUpdates\LCU\windows10.0-kb4016871-x64_27dfce9dbd92670711822de2f5f5ce0151551b7d.msu**.

Tip: You can rename the file to something shorter to make it easier to work with. For example, **KB4016871.msu**.

Make sure to grab all the available updates and put them in the folders noted earlier.

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

There are 3 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 **boot.wim**, and then, finally, the **install.wim**.

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

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 10 Pro.

Mount the image of your preferred Windows 10 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 neither an SSU (Servicing Stack Update) file nor a Safe OS Dynamic Update, skip this section on updating the **WinRE.wim**.

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"
```

The next 2 commands will inject the SSU (Servicing Stack Update) and Safe OS Dynamic Update into WinRE. If there is no SSU Update or Safe OS Dynamic Update, skip that item. Note that the use of **/LogPath** here and in all further commands is optional.

```
Dism /Add-Package /Image:"C:\Project\WinRE_Mount"  
/PackagePath="C:\WinUpdates\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 such as the LCU and SSU, 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 the **"other"** category where multiple **"other"** updates 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 also 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 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 **SSU** (Servicing Stack Update) and the **LCU** (Latest Cumulative Update)

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:\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 on 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:\WinUpdates\LCU" /LogPath="C:\Project\Logs\dism.log"
```

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

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

Updating the main OS is similar to updating the **WinRE.wim** and the **Boot.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'"
```

- 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 SSU, the LCU, and finally optional components such as the .NET cumulative update, etc.

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:\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"
```

Now we install the "**other**" updates (.NET cumulative update, 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"
```

As with the boot.wim, we will not yet unmount and export the install.wim since we will need to copy some files from it while it is mounted.

Some helpful commands (completely 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 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"
```

NOTE: When you apply cumulative updates, it will increase the build revision number of Windows. The revision number can be found in the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\UBR
```

```
*****  
* Updating the Base Image 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 in this folder). 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 duplicates of files located elsewhere on the media. These files should be identical, however, after applying updates, some files may not be in sync. The Setup Dynamic Update should update those files but that does not appear to happen. There are also a few instances of setup.exe which should be in sync but may not be. To resolve this, we will run several robocopy commands. These commands will copy only duplicate files that are newer than the duplicate files.

To perform this synchronization, run these commands:

```
robocopy "c:\project\winpe_mount" "c:\project\iso_files" setup.exe /xo  
/xx /xl /r:0 /w:0
```

```
robocopy "c:\project\winpe_mount\sources"  
"c:\project\iso_files\sources" setup.exe /xo /xx /xl /r:0 /w:0
```

```
robocopy "c:\project\winpe_mount\windows\system32"  
"c:\project\iso_files\sources" *.* /e /ndl /xo /xx /xl /np /r:0 /w:0
```

```
robocopy "c:\project\mount\windows\system32"  
"c:\project\iso_files\sources" *.* /e /ndl /xo /xx /xl /np /r:0 /w:0
```

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.

Now, close the boot.wim and export it. Then 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, the export adds that image to the file. You now have an image file with both index 1 and 2 updated images.

```
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:\Win10PROx64.iso
```

Replace the 3 occurrences of **c:\project\iso_files** with the correct path to where your files are located. Also, replace the **c:\Win10PROx64.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 10 Image with Latest Updates *
*****
```