

# Windows 11 PE Audiophile Creation Guide

by samotc

## Introduction

### I. Tools that we will need:

Win10XPE App

<https://github.com/ChrisRfr/Win10XPE>

<https://sourceforge.net/projects/win10xpe/>

<https://www.tenforums.com/software-apps/182581-win10xpe-build-your-own-rescue-media-2-a.html>

Win10PE Strelec:

[https://mega.nz/file/E9d1gALR#87OpVpOgZuBQ8dri-c-NTcgliBkgexgEK7oxVpH\\_Hbl](https://mega.nz/file/E9d1gALR#87OpVpOgZuBQ8dri-c-NTcgliBkgexgEK7oxVpH_Hbl)

Bootice

Dism++

Remote Registry

Ultraiso

WinNTsetup

(all of them contained in win10 PE Strelec)

RegistryChangesView v1.28

[https://www.nirsoft.net/utils/registry\\_changes\\_view.html](https://www.nirsoft.net/utils/registry_changes_view.html)

## II. Implementation sequence:

Basic Win11XPE creation (.ISO file)

ISO ---> USB pendrive (check that it starts and works fine)

Win11XPE "flatboot" installation

Application boot.wim image from ISO file to VHD / USB Pen

Copy ISO boot files to VHD / USB Pen

Add VHD boot to system BCD and check that it boots and works.

Drivers and applications install

Registry Export-Import

Boot VHD / USB Pen and check that everything works

System refinement (disable network if not needed, processes and services removal and clean, according to your needs).

Boot VHD to check functionality.

Mount VHD and save new boot.wim image with Dism ++

Open basic .ISO file with Ultraiso.

Change basic boot.wim to the new one created with Dism ++

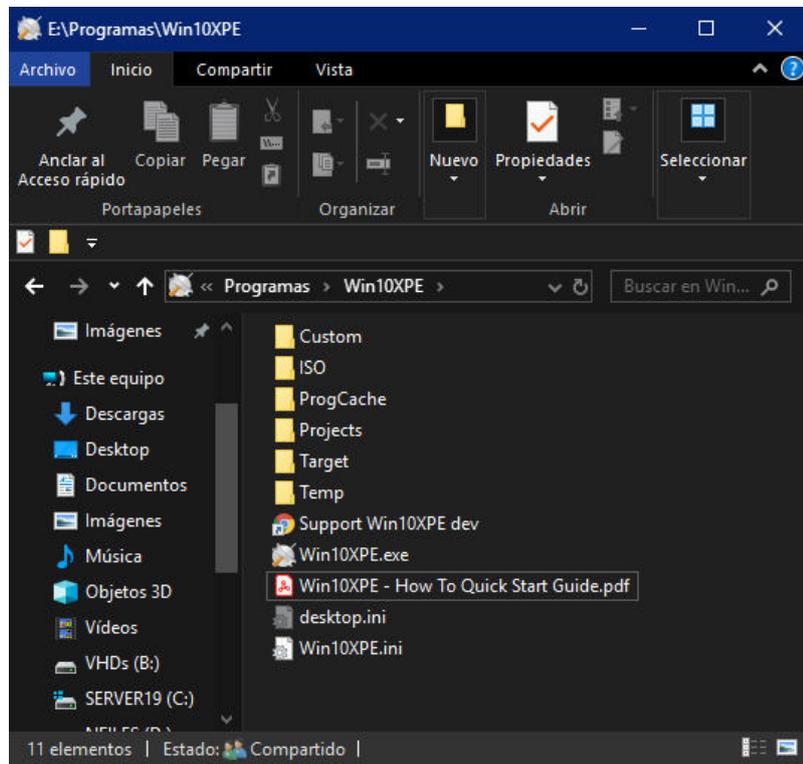
ISO ---> USB pendrive (check that it starts and works fine)

Done.

## 1. Basic Win11XPE creation (.ISO file)

Download the file **Win10XPE\_2021-10-27.7z** from any of the links provided in tools section, and unzip it in a folder.

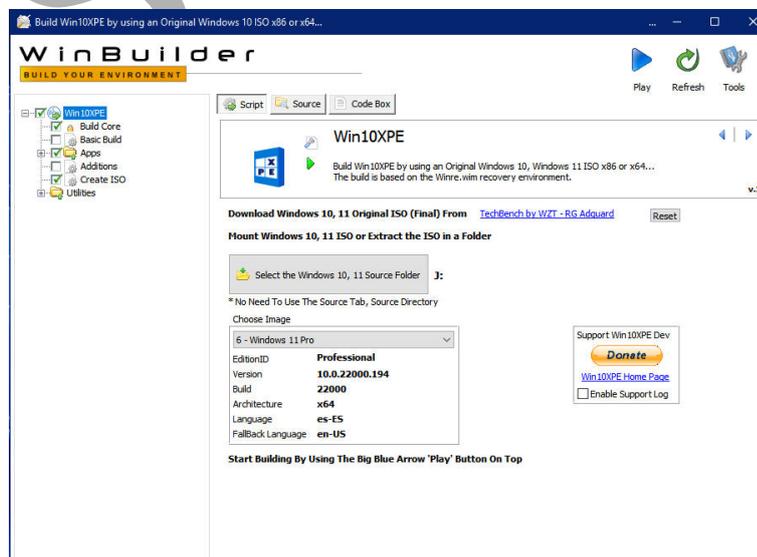
It should look like this:



Here we can see the file **Win10XPE - How To Quick Start Guide.pdf** highlighted, which can be consulted to learn more about the subject.

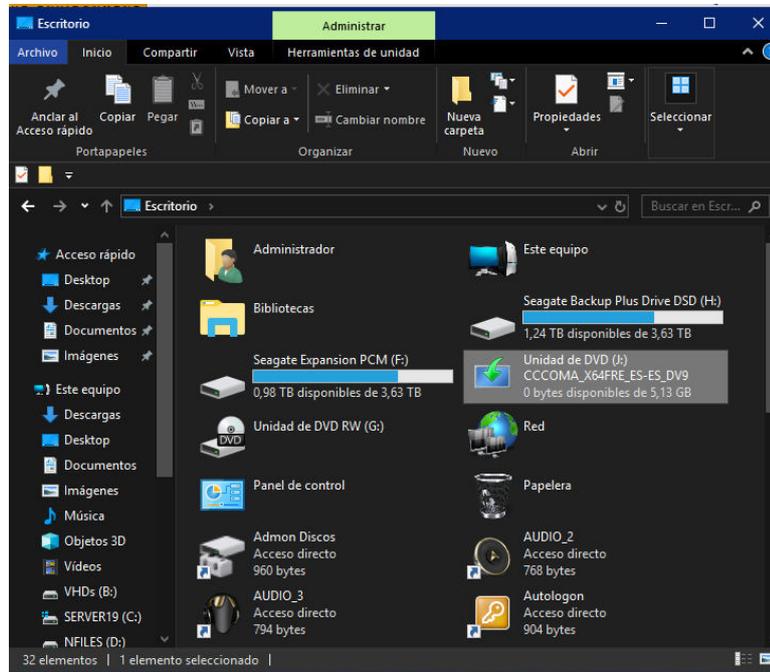
Select **Win10XPE.exe** and double click to run

It will show like this:

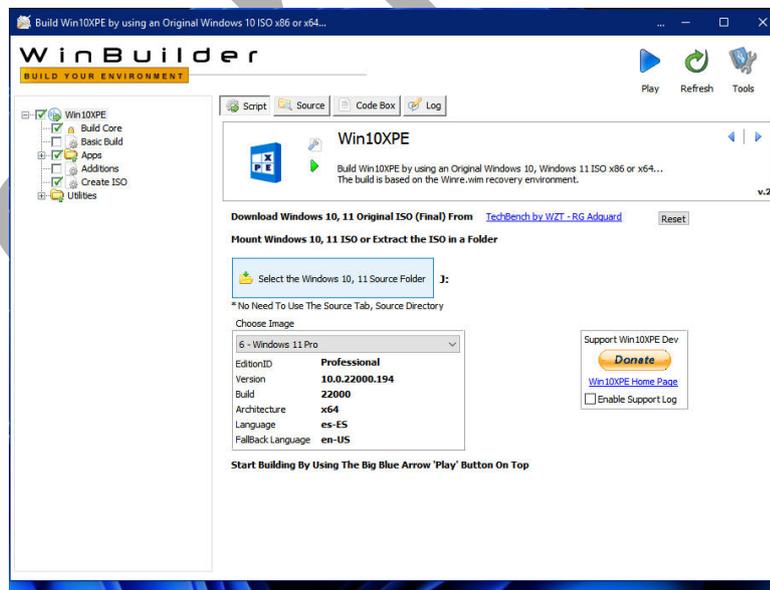


Select Win10XPE box

Now we must mount the file.iso according to windows version/edition we want to load (Win11 or Win10)

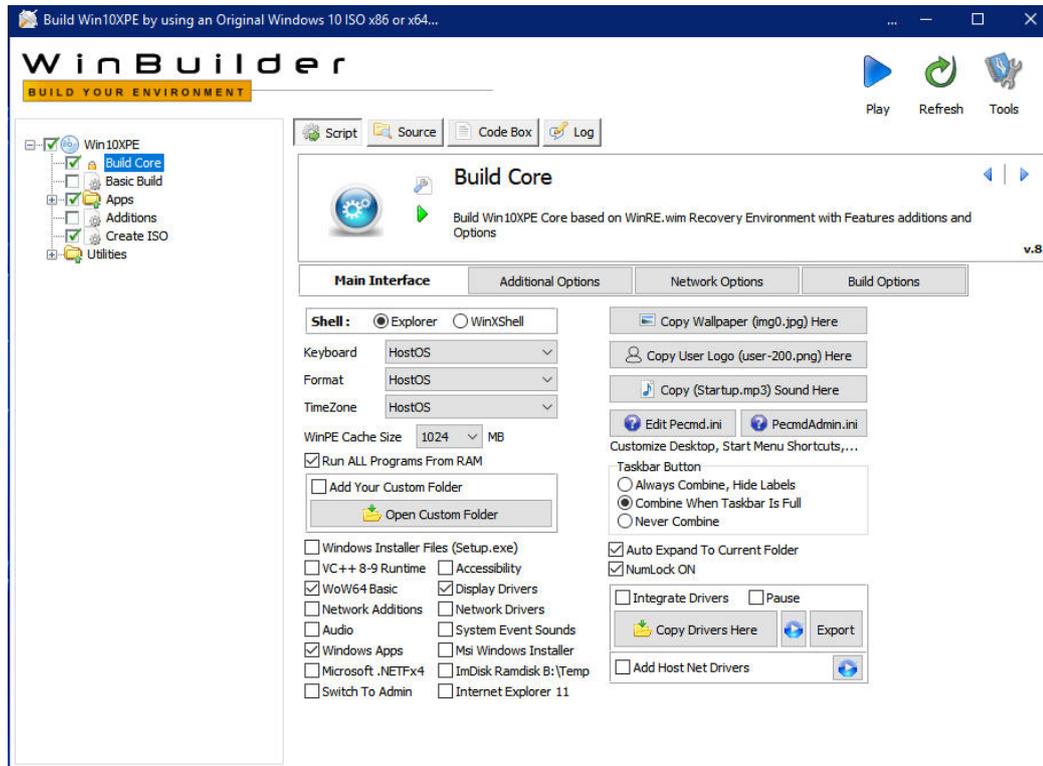


Then select source:



Once source is selected, press the button "Select the Windows 10, 11 folder".  
Now the program performs a bunch of tasks to prepare the necessary Windows 11 or 10 file upload.

Highlight Build Core in the left panel, and then select Main Interface in the right panel. It should show this:

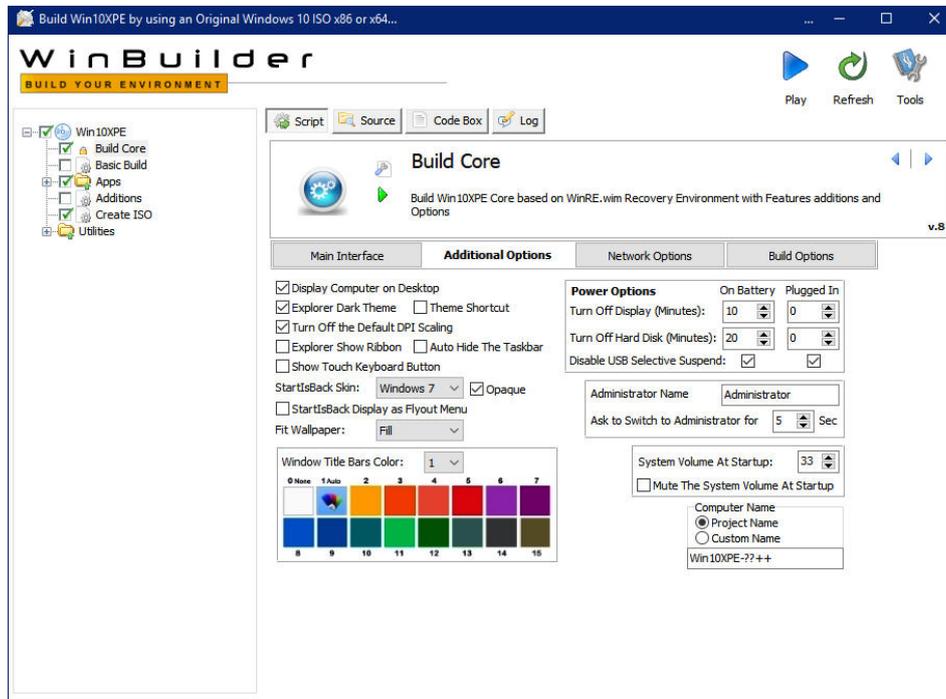


Check the boxes of your interest, here shows checked the ones I selected for my project. I'm trying to be as minimalist as possible, because in sound stuff "less is more". In "Shell" WinXShell takes up less than explorer. If you want to integrate drivers, check "integrate drivers" and then click "Copy Drivers Here". They must be in .inf format, setup.exe is not supported. I don't use it because it doesn't support exact drivers for my DACs or my NVIDIA graphics card. Anyway, I only install the Windows Display Basic Drivers because they need fewer resources, processes and services, and I think they are enough for an only audio system. Anybody who wants to change the default wallpaper, can do it by clicking the "Copy Wallpaper (img0.jpg) Here" button. Any image file you choose must be renamed to img0.jpg. Same with "Copy User Logon" (file "user-200.png") and "Copy (Startup.mp3) Here". As for "Edit pecmd.ini" and "PecomdAdmin.ini", it is not necessary to configure them now, it can be done later, so it will be explained later too, since it will be easier to see the changes according to our modifications in those .ini files. Checking "Msi Windows Installer" may be necessary for the installation of some drivers / applications.

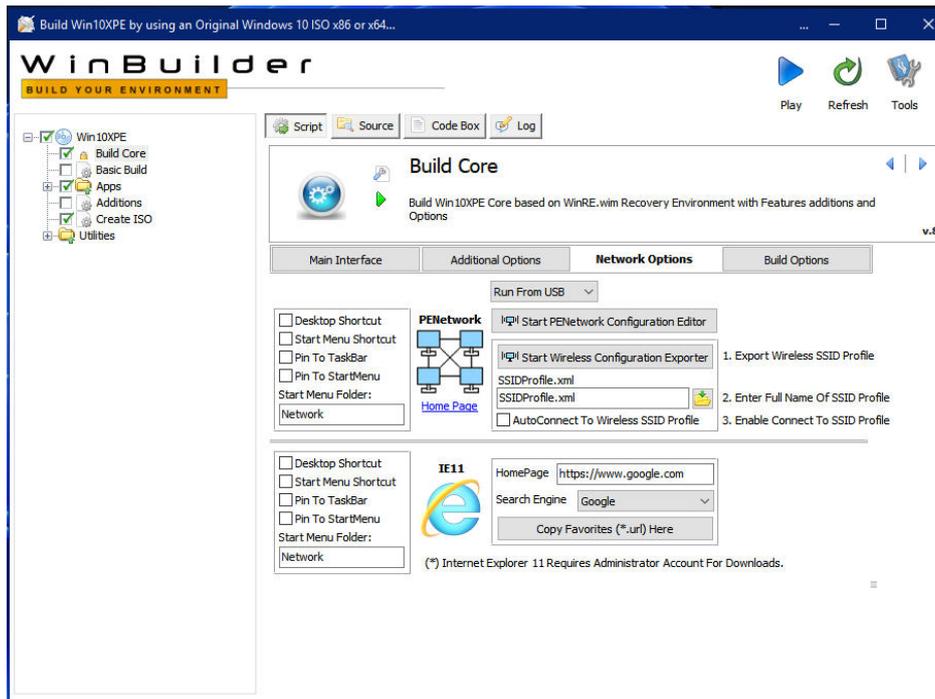
Finally, check the box "Run All Programs for RAM", as this is the most interesting thing about Windows PE, it is able to run applications in RAM, specially audio ones.

Once the options we wanted have been established in the Main Interface, we display "Additional Options" by clicking on that button.

This screen should show:

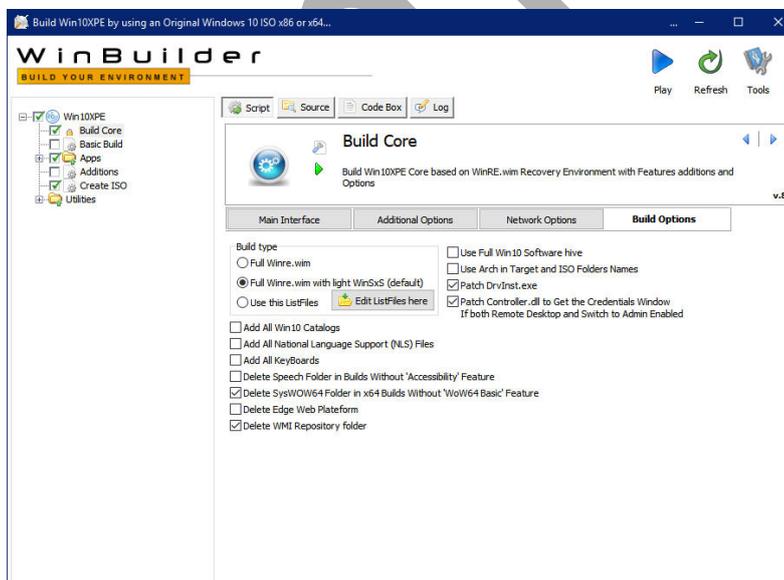


Choose the options that suit you best, that you need, by checking corresponding boxes. In "Project Name" put the name you want or more suitable for you. After this, we go to "Network Options" by pressing the corresponding button:



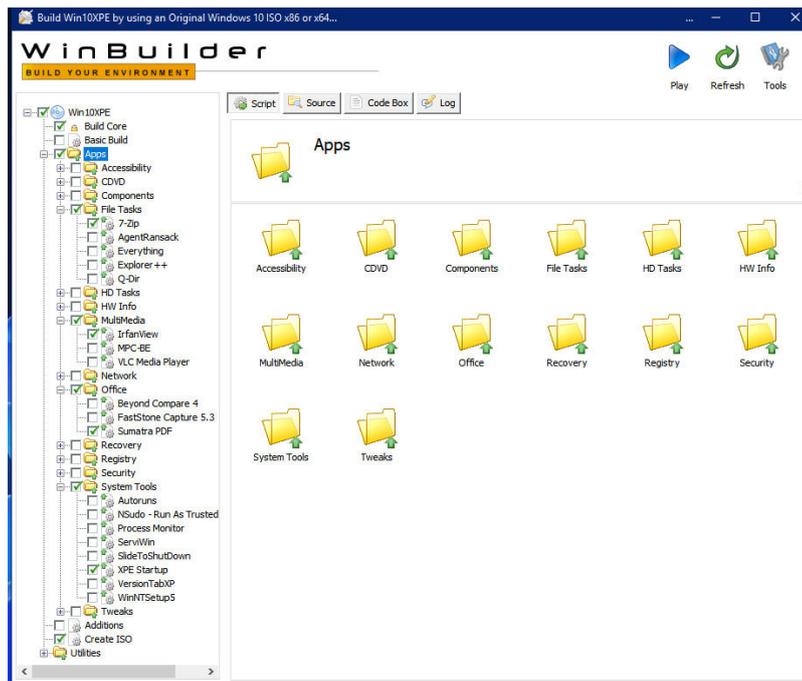
I didn't check anything, because I choose not to have network connection, I don't need it, and I think it would add "noise" to the audio, but that's my opinion.

Anyway, the network is installed by default, so this is only to customize the "PENetwork" and "Windows IExplorer" settings.

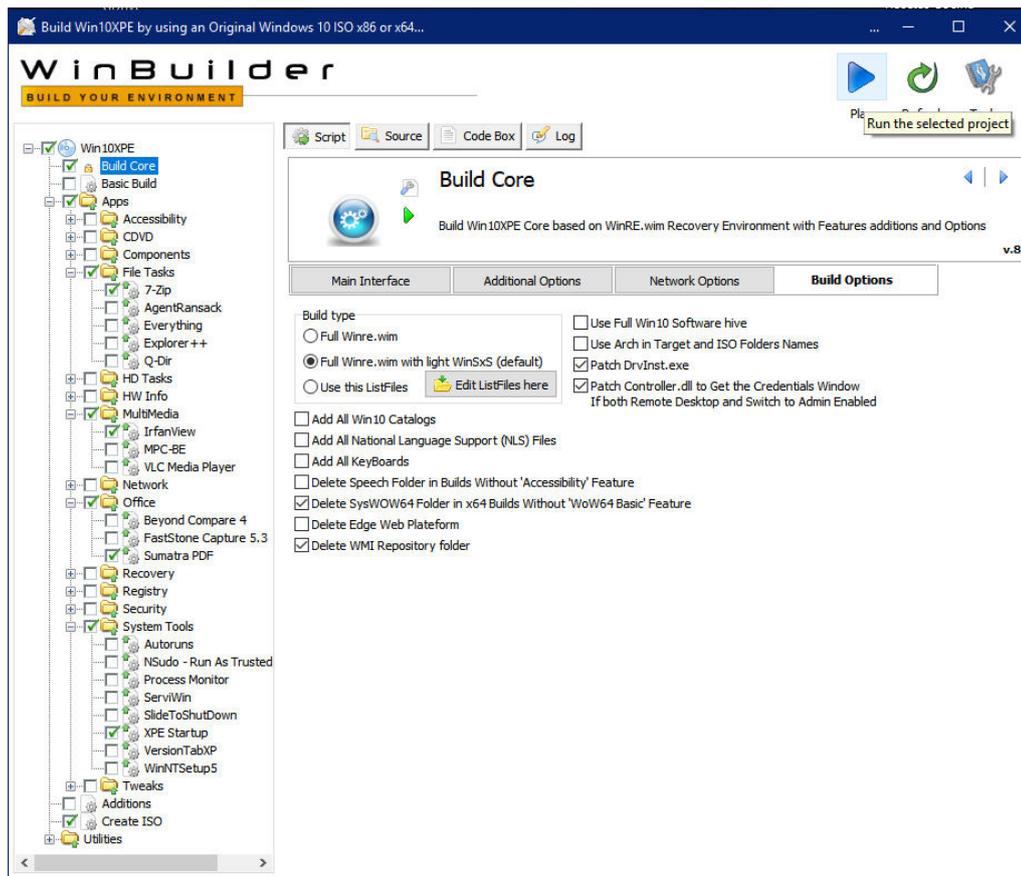


I left the checked default options. To obtain more information, you can check the pdf with the Guide in program folder.

Now is the time to highlight the "Apps" box on the left panel and display/expand them by clicking on the + sign on the left. This allows us to browse the apps and choose the ones we want to install.



Due to my minimalist endeavor, I only installed XPE Startup, which adds XPEStartup.cmd and XPEStartup.ini to the root of the USB unit, allowing you to customize some tasks without having to rebuild the project. After install your chosen apps, we only have to check the "Create ISO" box on the left panel. I will not talk about "Tweaks" and "Utilities" boxes as they are not necessary for my purpose at hand.



And finally, we only have to press the "play" button, blue arrow head, and see how the application is building our Win11 or 10, basic XPE, which takes several minutes. Created iso is saved in Win10XPE folder.

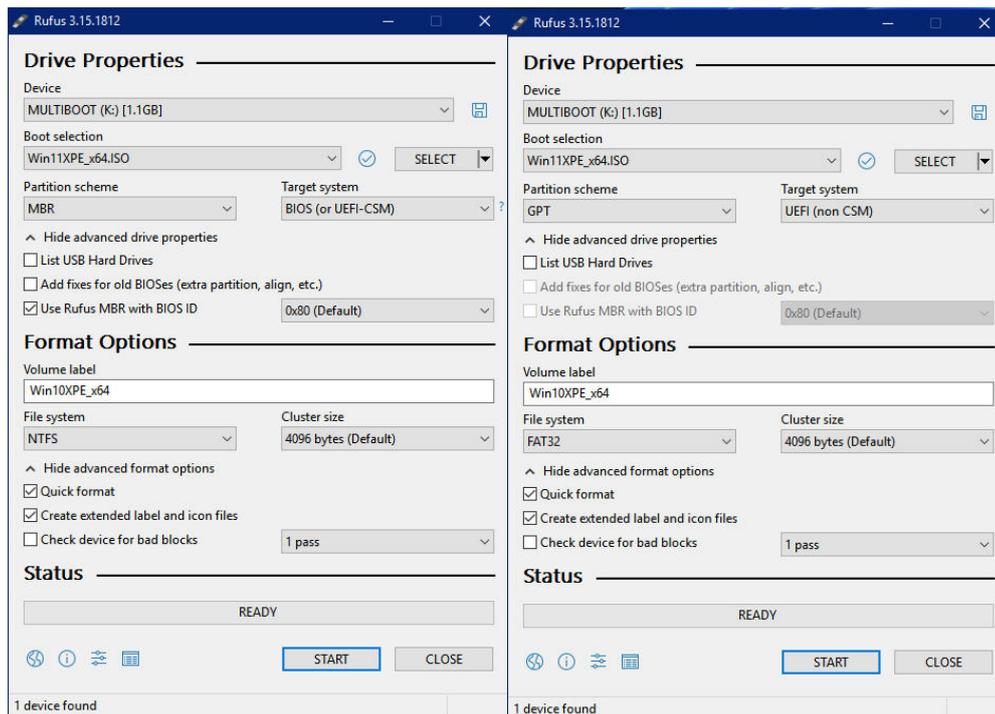
Now we have created a basic Win11PE.iso that will boot both in MBR and UEFI, it's time to take the next step, create a bootable USB pendrive to verify that the project works.

## 2. Creation of the Bootable USB Pendrive Win11XPE Basic

Objective is to verify that the startup and operation of our Win11XPE are correct, before continuing with the tasks that we have pending.

I have used Rufus 3.15, there are other tools, but this is the one I know best.

We have 2 options: MBR / BIOS UEFI (CSM), which will work on most PCs, or GPT UEFI, which will only boot on UEFI systems.



Make your choice, then press "SELECT" button and select the .iso file corresponding to our newly created Win11XPE, then press "START" and wait for the program to copy the files contained in the Win11XPE.iso to the USB Pendrive.

Once this is done, boot your PC from USB Pendrive.

If boot works, we are ready to create the VHD, mandatory for the next step.

### 3. Win11 XPE Flatboot Installation

Windows PE usually boots from a boot.sdi file that loads boot.wim and creates an **X: drive in RAM** and works from there.

This means that every modification we make to the system, adding or changing files, installing applications, drivers, etc., will be lost when it is turned off; as well as changes made to the registry or configuration.

To avoid this, this is why the flatboot installation is necessary

A **flatboot installation** consists of Windows located on the X: drive, running from HD, that is, it is not loaded into RAM.

In fact, any Windows PE needs at least 2 GB of RAM to work, while in flatboot mode it only needs 512 MB.

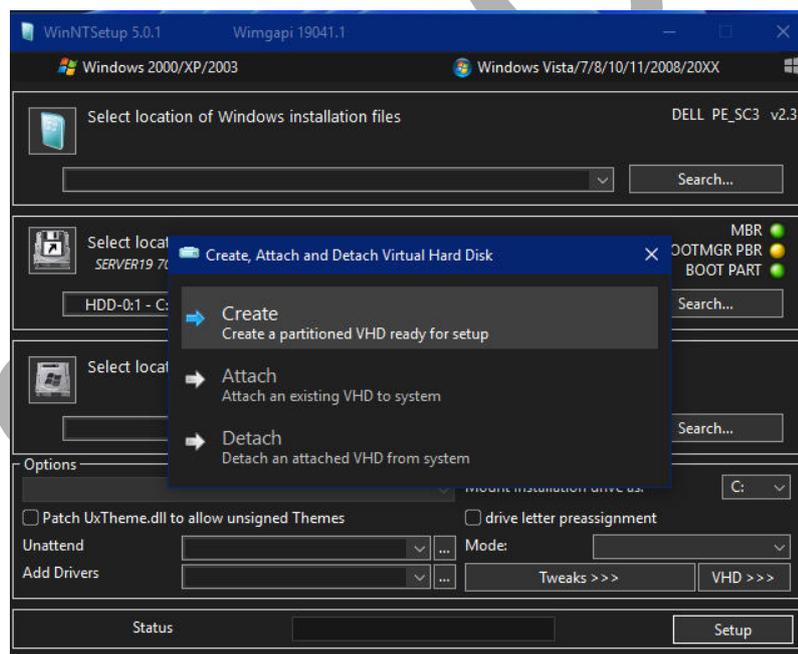
This installation mode preserves the new files introduced into the system, which are not lost on the next restart, and allows the installation of new drivers and applications. The only downside is that the Windows PE registry is volatile by Microsoft's design, and any changes made to it ARE LOST on the next reboot.

In any case, this problem will be easily solved as we will explain later.

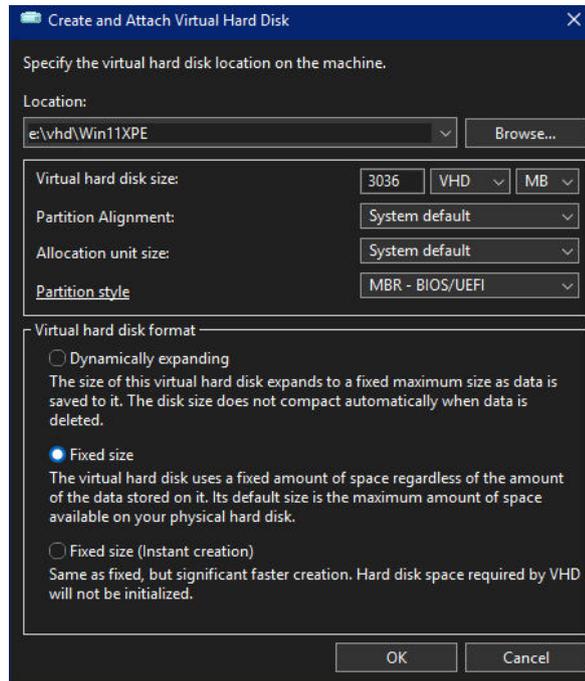
#### a. VHD Creation:

The first step is to create a VHD where to install the Win11XPE, this can be done through the bootpart commandline, diskmgr.msc, or through an application, **WinNTSetup**, (<https://msfn.org/board/topic/149612-winntsetup-v512/>) which is an easier and faster way to do it.

We start by executing WinNTSetup, appearing the following screen:



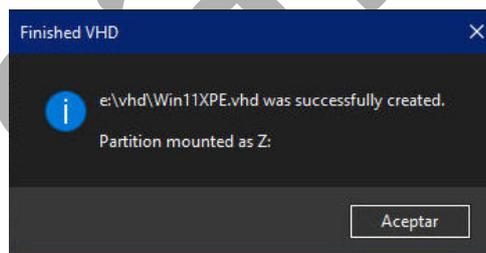
We click on "VHD >>>" and select "Create":  
Now we will see the following screen:



Now we select the size of the VHD, location, and partition style (MBR, MBR-BIOS / UEFI or only UEFI) as it suits us.

A fixed size of 3GB is enough for our purpose.

We click "OK" and then this will appear:



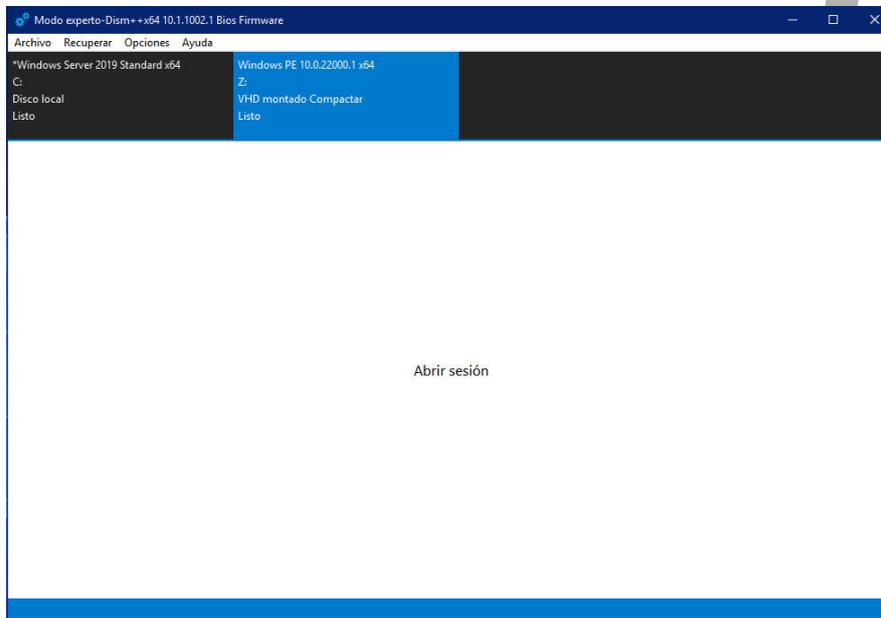
We already have the VHD created and mounted, ready to use and take the next step.

## b. Application boot.wim image from ISO file to VHD

We previously mounted the Win11XPE.iso image, boot.wim localized into folder sources. We can do it through the command "**dism.exe**", typing "Dism /apply-image /imagefile:boot.wim /index: 1 /ApplyDir: Z:\" putting the full path to boot.wim file, and where Z: is the letter corresponding to the drive of the VHD created earlier.

Another easier way is to use the **Dism ++** app, contained in Windows PE Strelec, or it can be downloaded from here: <https://www.chuyu.me/es/index.html>

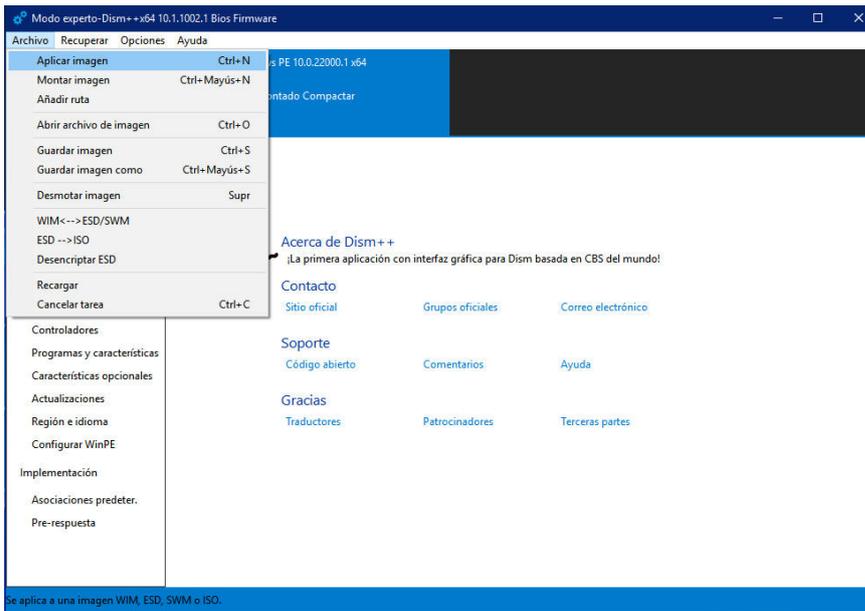
Run DISM ++, you will see the following:



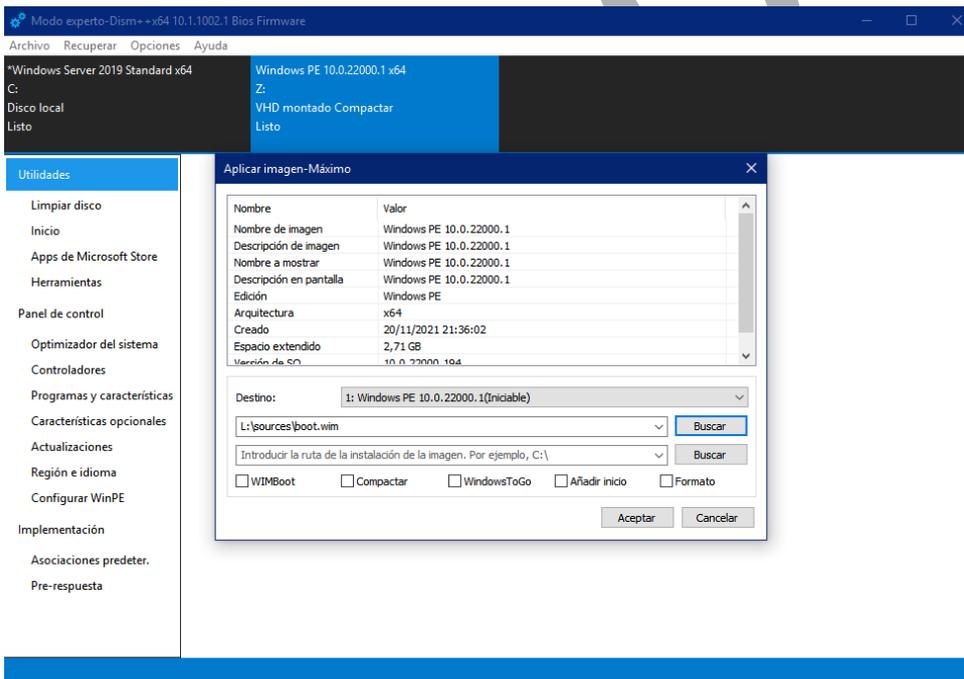
IMPORTANT:

Activate "expert mode" (options >> more options >> expert mode)

Click on **open session** and then go to menu> file> apply image, it should look like this:

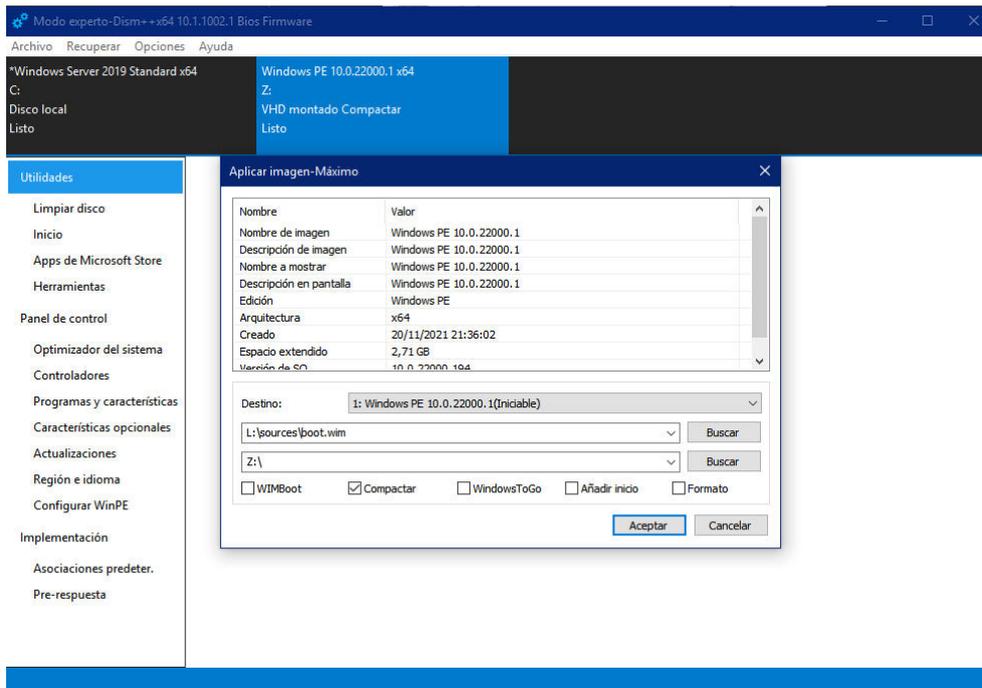


Click on **apply image**, this will appear:



Now you should enter the **full path** to the boot.wim file of the mounted iso, in the case of the screenshot "L:\sources\boot.wim".

Then you must find the target where this image is going to be applied, the letter of the mounted VHD drive (in this case Z:\)



Finally check the **compact** checkbox and press Accept button.

### c. Copy boot files ISO to VHD

Copy boot and efi folders, and the BOOTMGR, bootmgr.efi, bootmgr.exe files.

**DO NOT COPY, CDUsb.y, XPEStartup.cmd, or XPEStartup.ini**

The boot.sdi file, contained in the boot folder, does not need to be copied either.

### d. Add boot VHD to system BCD

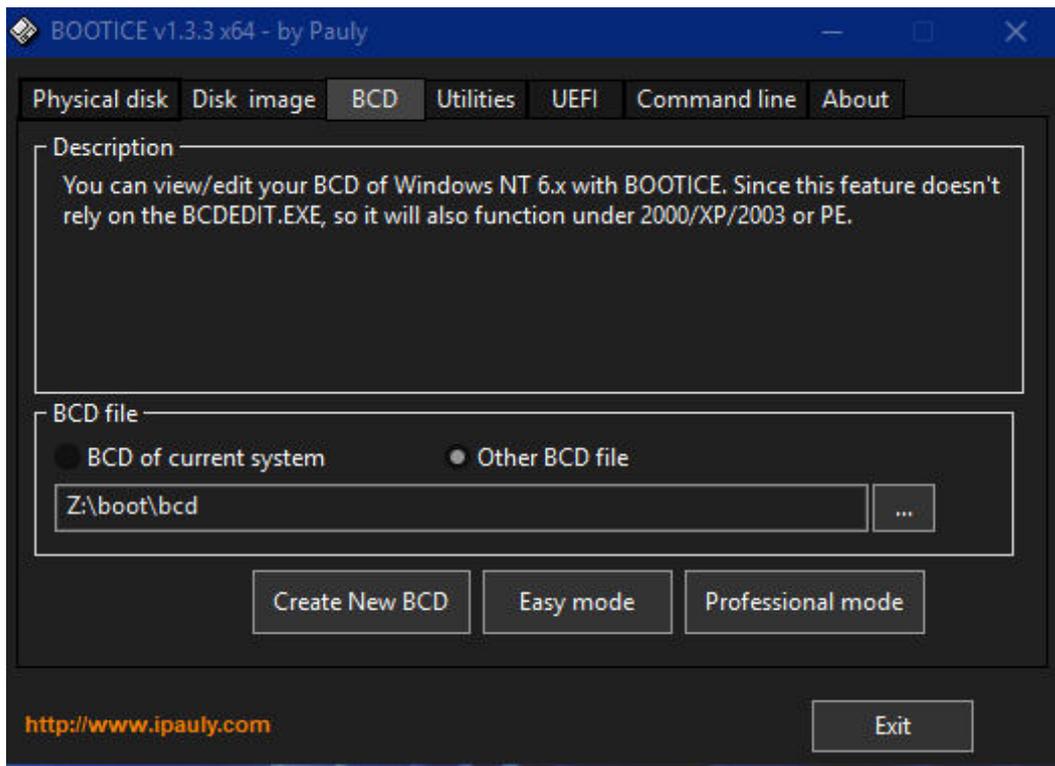
The objective is to be able to start the VHD to verify that it works and continue with the steps that remain. For that we need a small program, **Bootice**, which is included in WinPE Strelec, or it can be downloaded from here:

<https://reboot.pro/files/file/592-bootice-v1332/>

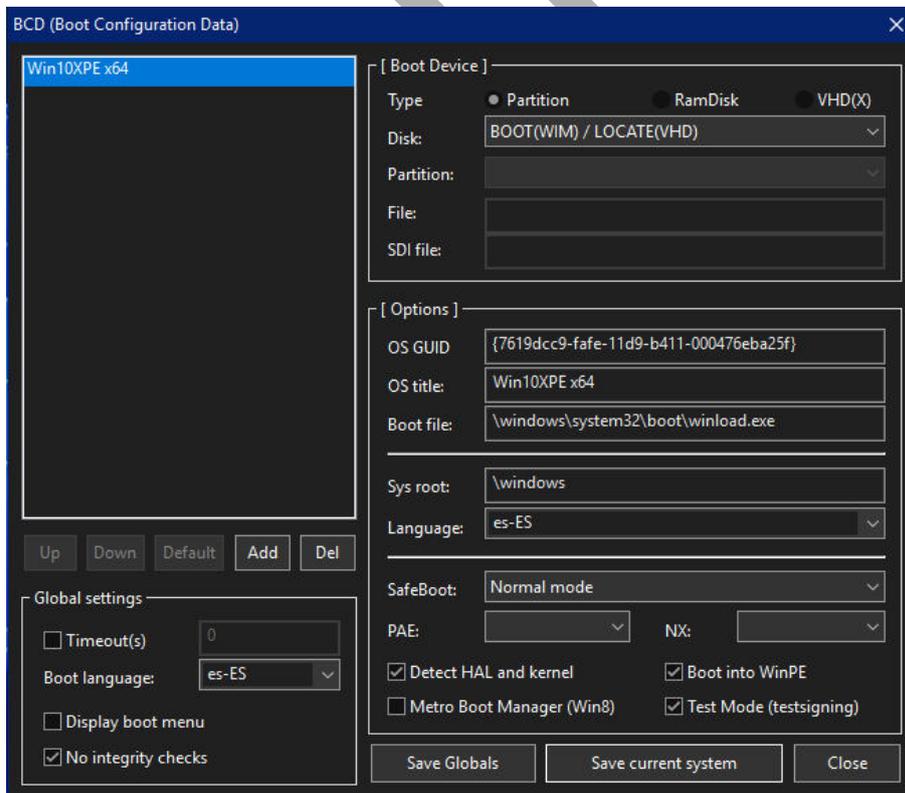
Run Bootice, click on the BCD tab select "another BCD file"; go to the boot folder of the VHD and select the BCD file.

Edit or create a new VHD entry in both BCDs located on boot partition of your HD or USB device, in accordance with attached pictures:

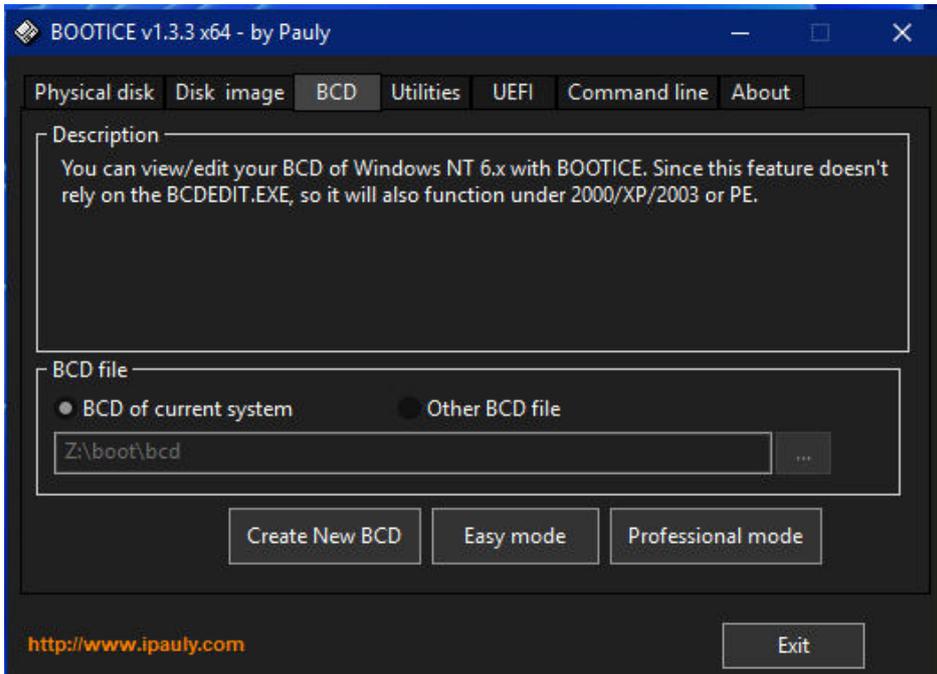
## BCD VHD:



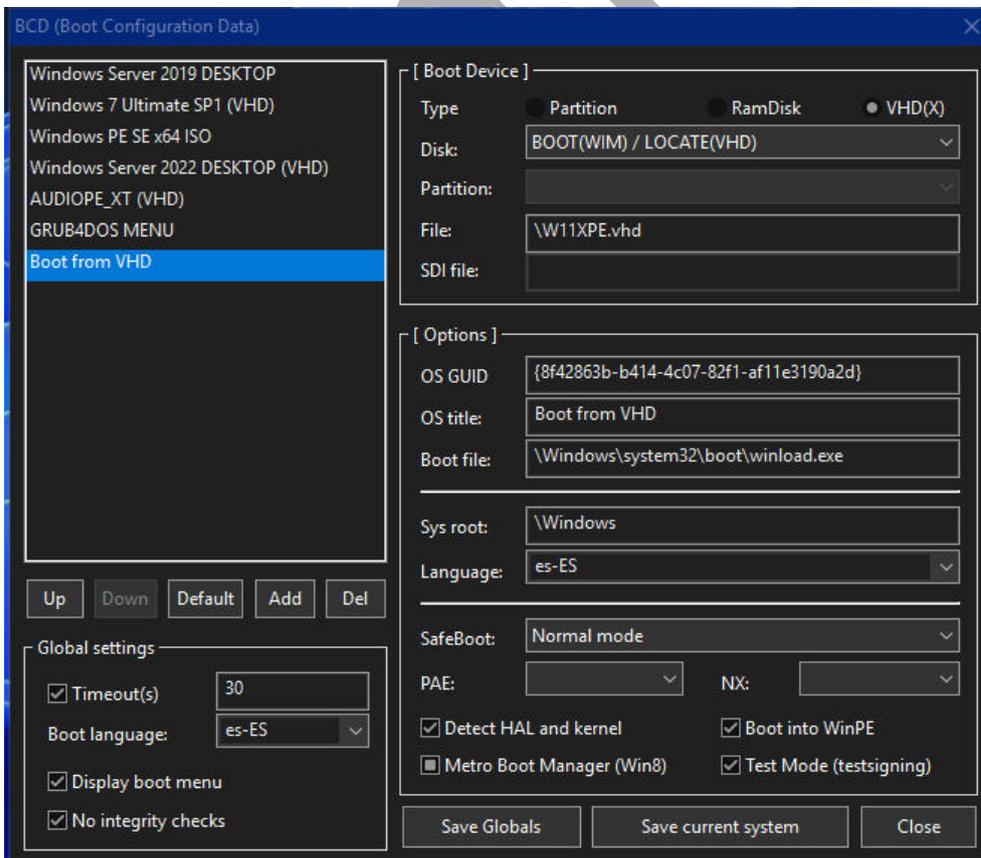
Select “**Easy mode**” and click the button, then the following screen will appear, it has to look exactly like this (except boot language):



Now we go to the **BCD** of the **system**:



We select "Easy mode", it has to look exactly like this (except boot language):



**Look carefully, this is very important:**

**Edit on the Boot\BCD the bootfile to point to:**

**\windows\system32\boot\winload.exe for MBR / CSM booting.**

**Edit on the EFI\Microsoft\Boot\BCD the bootfile to point to:**

**\windows\system32\boot\winload.efi for UEFI booting.**

With this, when we restart the PC the start of the VHD will have been added to our start menu.

If we do **not want to touch the system's BCD**, we have another possibility, safer if we do not have experience in these matters, it consists in **applying the BOOT.WIM** image on a **USB Pen** formatted in NTFS, or FAT32, being the PRIMARY AND ACTIVE partition, and boot from USB Pen. It must be 3.0 or 3.1, USB 2.0 is very slow.

The steps for this are the same as for the VHD, **don't forget to modify the BCD of the USB Pen** as described for the BCD of the VHD. The **system BCD does not need to be modified**. Now we start the VHD or the USB Pen and verify that it is indeed the case, and it works.

Now we start the VHD or the USB Pen and verify that it is indeed the case, and it works.

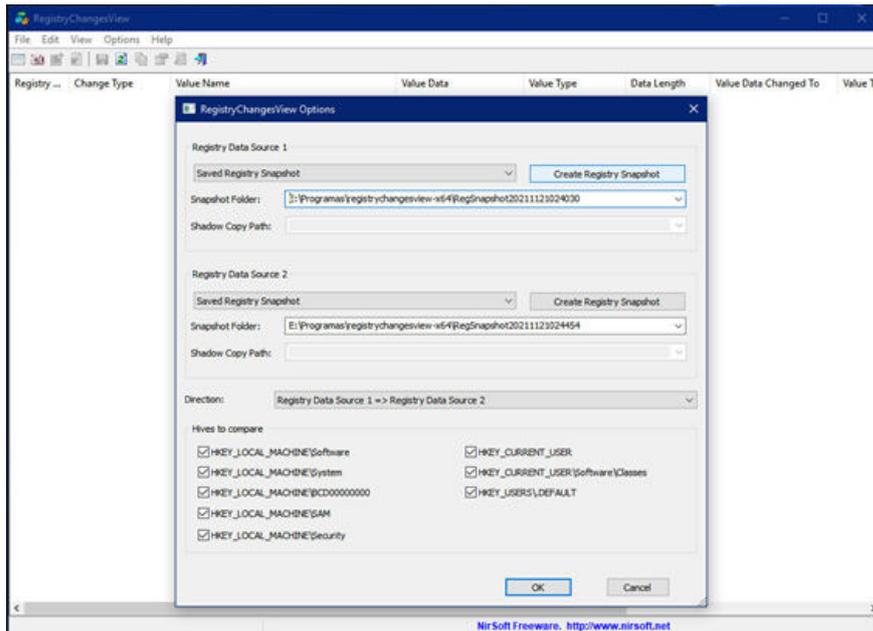
Well, whether in VHD or USB Pen, we are ready for the next step, which will be the installation of all the drivers and audio applications that we want or need.

**e. Installation of drivers and applications:**

Before installing anything, we must run the "RegistryChangesView v1.28" program that can be downloaded from here:

[https://www.nirsoft.net/utils/registry\\_changes\\_view.html](https://www.nirsoft.net/utils/registry_changes_view.html)

The first thing you see is this:



You must have the executable and its folder outside the VHD or USB Flash Drive.

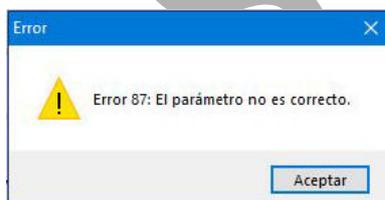
Now in the heading "**Registry Data Source 1**", press the button "**Create Registry Snapshot**", this will capture the status of the pre-install registration.

Now you can minimize the application, and begin with the installation of all the applications and drivers that you need.

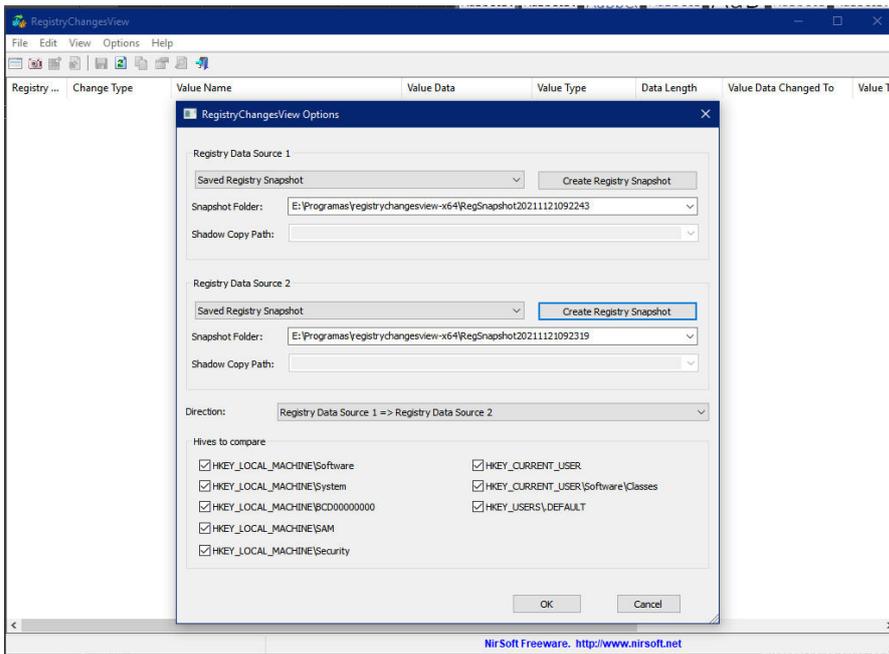
Finished the process of all the installations **DO NOT RESTART THE PC**, now you must take the **second snapshot**, in the section corresponding to "**Registry Data Source 2**".

This will capture the registry status after all installs.

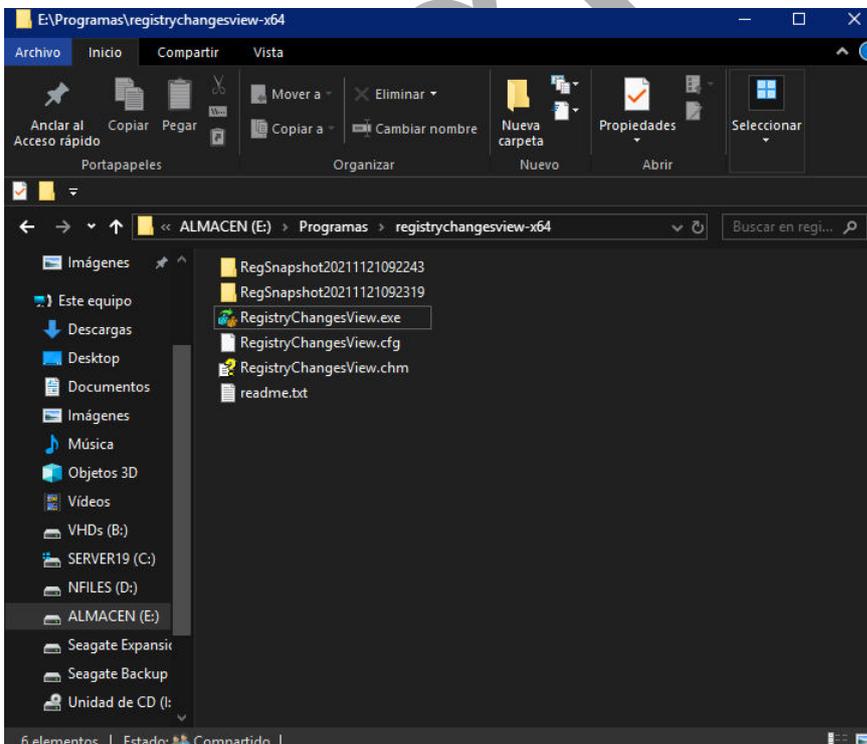
**NOTE.**- If an error appears after capturing, **ignore it and proceed**, the snapshot has been taken and saved. You can check this by seeing that a regsnapshotxxxxx folder, with content, has been created in the program folder.



This would look like this:

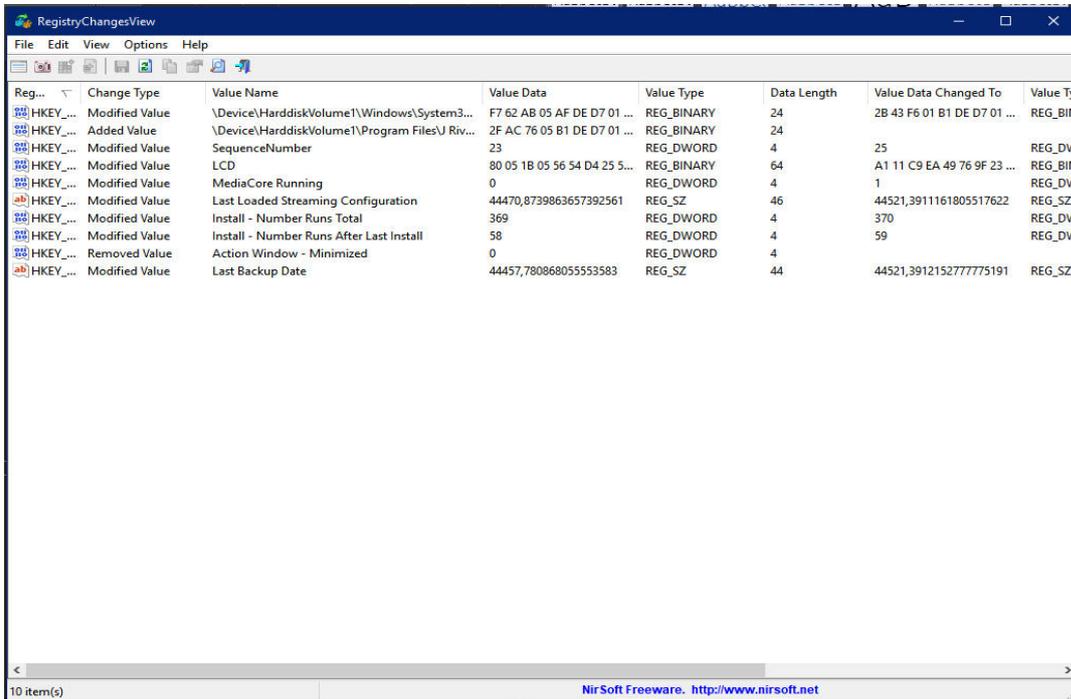


After the second snapshot, before pressing OK, check that the numbers of the regsnapshot folder of "Registry Data Source 1" and "Registry Data Source 2" that appear in the boxes where the path is seen, correspond to the folders "regsnapshotxxxxxx" contained in the program folder.



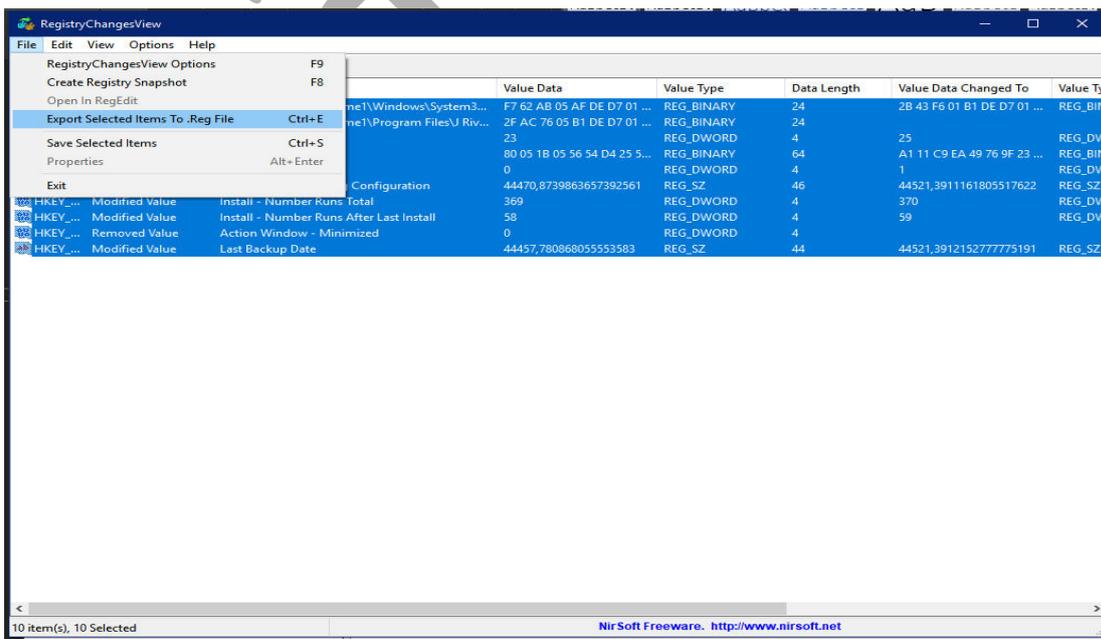
If not, you must copy the lowest number (corresponding to the first capture) in "Registry Data Source 1" and the highest in "Registry Data Source 2".

Once this is done, you can now press the **OK** key, the program now compares the 2 snapshots and will detect the differences of the pre and post installation record, which will be displayed on a screen:



We can save these changes in a file that can later be incorporated into the Win11XPE offline registry, where they no longer be lost, and when the system starts up again, that information is already in the registry permanently. We will select **all the values**, and we will go to menu> file> **export selected items to .Reg file**.

We will save this ".reg file" in a folder external to the VHD or to the USB Pendrive, for later OFFLINE import.



## f. Offline import to the VHD or USB Pendrive Registry

We restart the PC with Windows PE Strelec and then mount the VHD or insert the USB Pen that contains our Win11XPE.

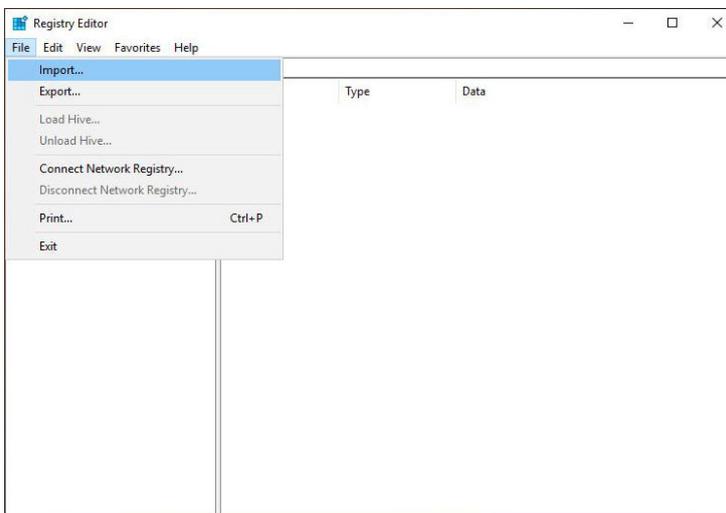
Win PE Strelec can be downloaded from here:

[https://mega.nz/file/E9d1gALR#87OpVpOgZuBQ8dri-c-NTcgliBkgexgEK7oxVpH\\_Hbl](https://mega.nz/file/E9d1gALR#87OpVpOgZuBQ8dri-c-NTcgliBkgexgEK7oxVpH_Hbl)

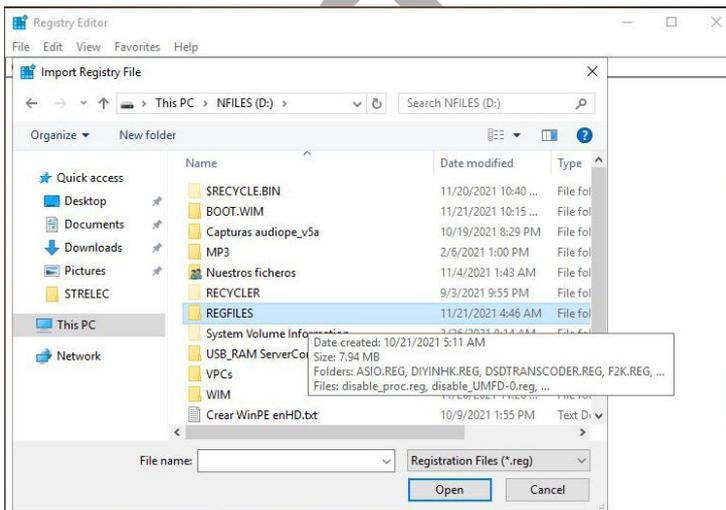
[https://www.majorgeeks.com/files/details/sergei\\_strelecs\\_winpe.html](https://www.majorgeeks.com/files/details/sergei_strelecs_winpe.html)

Once started, we go to start menu> programs> utilities> **remote registry** and we open.

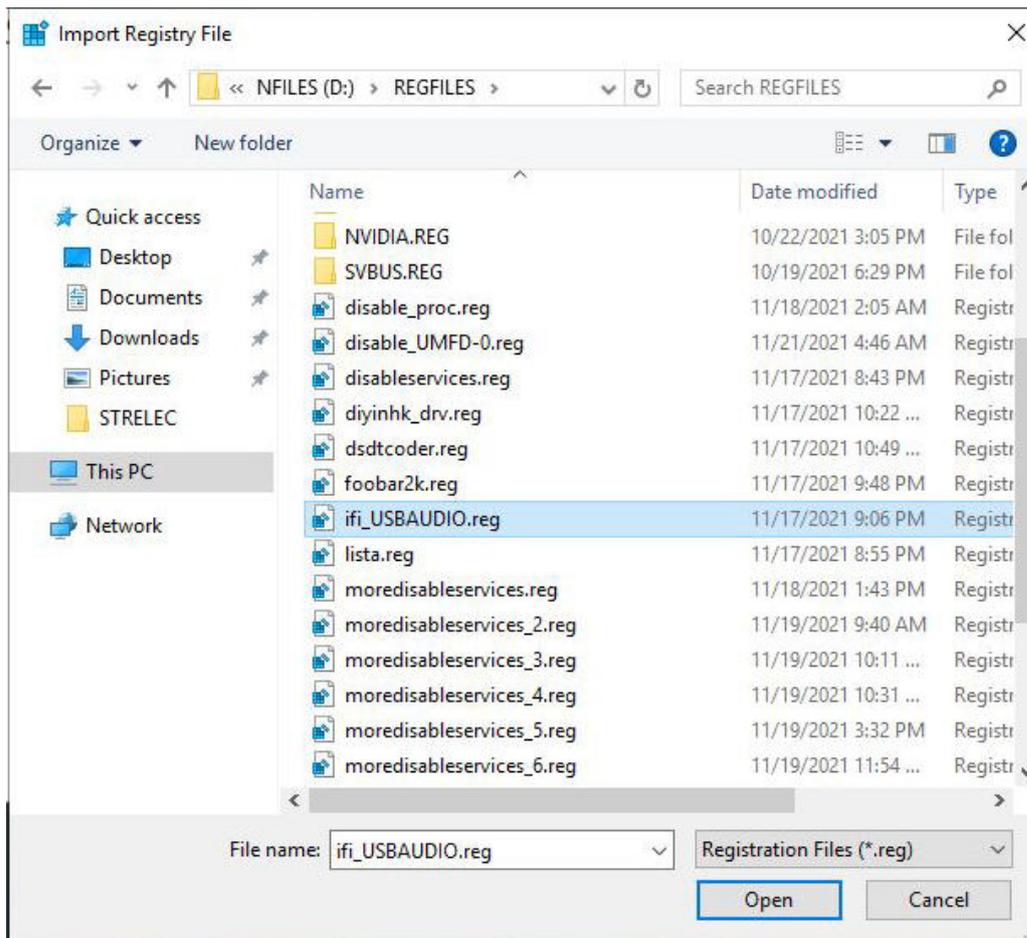
Then we go to the drive corresponding to the mounted VHD, or to the USB Pen, we open it and look for the Windows folder, select it and open it. This screen will appear:



We select menu> file> **import registry file** and we look for the location where we save the .reg file that we had exported before exiting Win11XPE.



We select the .reg file that we want to import



we press OPEN, and the changes we made to the registry with the installation of drivers and applications are already imported.

Another alternative is to do the individualized export-import process for each driver and application that we install, it is slower, but if something goes wrong, it will be easier to locate the fault.

It also has the advantage that if we create another future Win11XPE, and we are not interested in any of the drivers or installed applications, we may not install it.

This is the method that I have applied, and it is the one that I advise, although it takes a few minutes to complete the process.

Well, now we have a VHD or a USB Pen with a fully operational "Flatboot" installation of Win11XPE Audiophile, but we must check that it works, and we will restart the PC from it.

### g. Tweaks:

Now that we have the system operational, we can tune it a little more if we want, and change some settings, disable some processes and services, etc.

For this there are 2 ways, edit offline "pecmd.ini" located in Windows\System32\ and execute "taskmgr.exe" and/or "services.msc"

A very complete information can be seen here:

[https://www.isunshare.com/blog/detailed-explanation-for-pecmd-in-winpe-pecmd-  
ini-file-configuration-1/](https://www.isunshare.com/blog/detailed-explanation-for-pecmd-in-winpe-pecmd-ini-file-configuration-1/)

Pecmd.ini determines what pecmd.exe will load when Windows PE starts, drivers, network, desktop, start menu, etc.

We can edit it offline with notepad and make modifications that will persist on all Win11XPE boots.

#### 1. DISABLING THE NETWORK

I must thank seeteeyou, partner of the forum, who told me how to do it, and a link where there is more information:

[https://serverfault.com/questions/866479/disable-network-connection-in-windows-  
pe](https://serverfault.com/questions/866479/disable-network-connection-in-windows-pe)

In outline, it consists of creating an unattend.xml file, modifying the **winpeshl.ini** file by editing it with notepad, offline of course.

```
[LaunchApps]
wpeinit /unattend="%SystemRoot%\system32\Unattend.xml"
```

This is how it should look

This is how the **unattend.xml** file looks

Copy both files to \windows\system32 of the VHD or USB Pendrive and in the next reboot the network will be disabled.

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      processorArchitecture="AMD64">
```

## 2. EDIT PECMD.INI

With this we can prevent unwanted programs from starting, shortcuts on the desktop that we do not want, etc, we do it with notepad, obviously offline.

An advantage is that if the result of the changes we do not like, it is easy to reverse

Pecmd.ini sample:

```
_SUB NetInit
//=Load Network drivers at startup

//=EXEC =@!%WinDir%\System32\Drvload.exe %WinDir%\inf\athw8x.inf
%WinDir%\inf\netathr10x.inf %WinDir%\inf\netathrx.inf %WinDir%\inf\netbc63a.inf
%WinDir%\inf\netbc64.inf %WinDir%\inf\netrtwlane.inf %WinDir%\inf\netrtwlane_13.inf
%WinDir%\inf\netrtwlanu.inf %WinDir%\inf\netwbw02.inf %WinDir%\inf\netwew00.inf
%WinDir%\inf\netwew01.inf %WinDir%\inf\netwlan92de.inf %WinDir%\inf\netwlv64.inf
%WinDir%\inf\netwns64.inf %WinDir%\inf\netwsw00.inf %WinDir%\inf\netwtw02.inf
%WinDir%\inf\netwtw04.inf %WinDir%\inf\netwtw06.inf

//=EXEC !=SetWG.exe WORKGROUP

//=EXEC @!%WinDir%\System32\Net.exe Start Wlansvc

//=EXEC @!%WinDir%\System32\Net.exe Start WinHttpAutoProxySvc

//= Launch PENetwork

//= EXEC X:\Program Files\PENetwork_x64\PENetwork.exe

// Remote Desktop Server Connect (Uncommented By RDC Plugin)

//=EXEC @!%WinDir%\System32\mstsc.exe /v:
```

To cancel lines insert at the beginning of the same the pair of characters "// =", without quotation marks, as this example shows.

I will not go into more detail, as this would take up many pages, I only show what is essential to change the most important settings. Whoever wants to go deeper I advise you to go to the page cited at the beginning of section g.

### h. Adjust processes and services

It is always better for the audio to run as few processes and services as possible in the background, the limit is in the extra functions of the system that we want to keep.

The only method that I know of, and have used, is to disable processes and services (seeing their dependencies), and see if this affects the functionality, first of the system, and after the audio, if that is the case, I do not disable.

To make these changes permanent, if they are not harmful, we can only use "RegistryChangesView", and take snapshot prior to the change, post change, export changefile.reg, restart with Win PE Strelec> remote registry> offline import of changefile .reg,

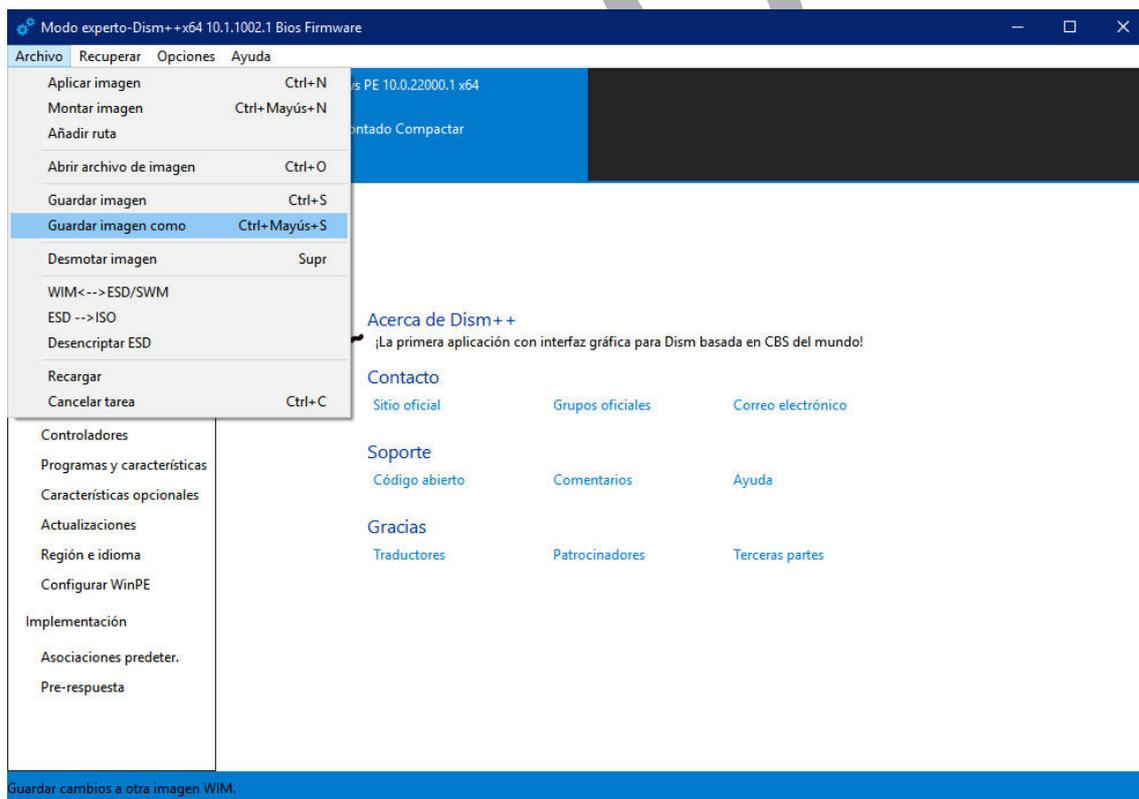
That is, the same as to install drivers and applications. I advise that each change we make has its corresponding changefile.reg, even if it is slower.

When we have already achieved the minimum of processes and services in the background that we want, or can, we are ready to restart the PC from the Win11XPE flatboot VHD or USB Pendrive. If everything works fine, we are ready for the last two steps.

### i. Create new boot.wim image

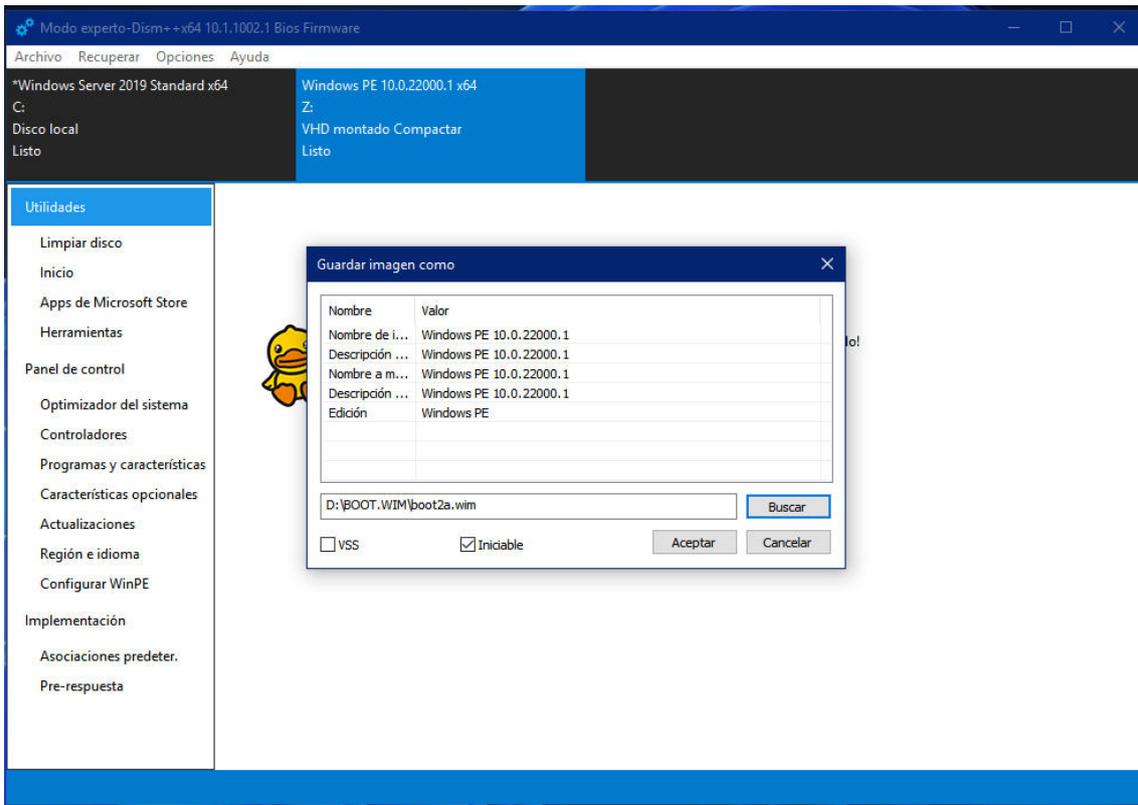
With the Win11XPE already updated with the drivers and applications, processes and services that we want, we must go from Win11XPE "flatboot" to "normal" WinXPE.

We start the PC with our host OS, mount the VHD or insert the USB Pen where the Win11XPE flatboot installations are and execute the **DISM ++** app



Open the program, in the upper ribbon it will show the windows detected, among them will be the Win11 XPE flatboot, which we will select, it will be highlighted in blue.

Then go to menu> file> **"Save image as"**, indicate the path where we want to save the image.wim, give it a name, and save.

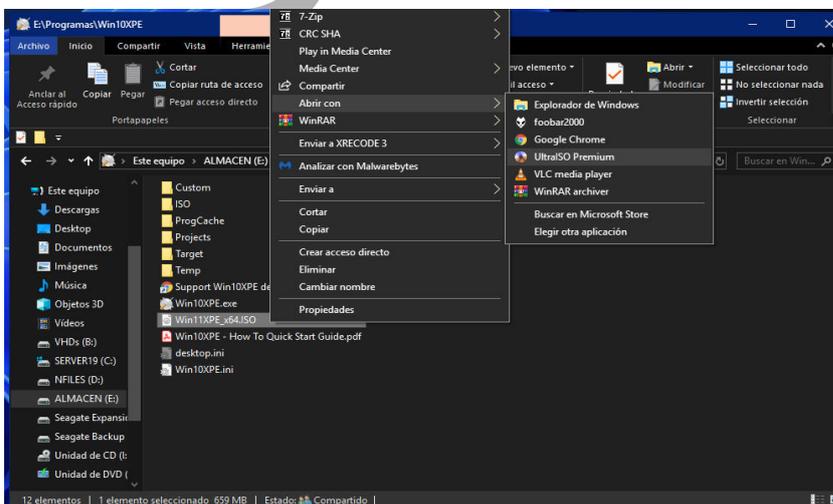


Once this is done, we only have one last step, to replace the raw ISO Win11XPE boot.wim file that we created at the beginning, with the new one, completely to our liking or convenience.

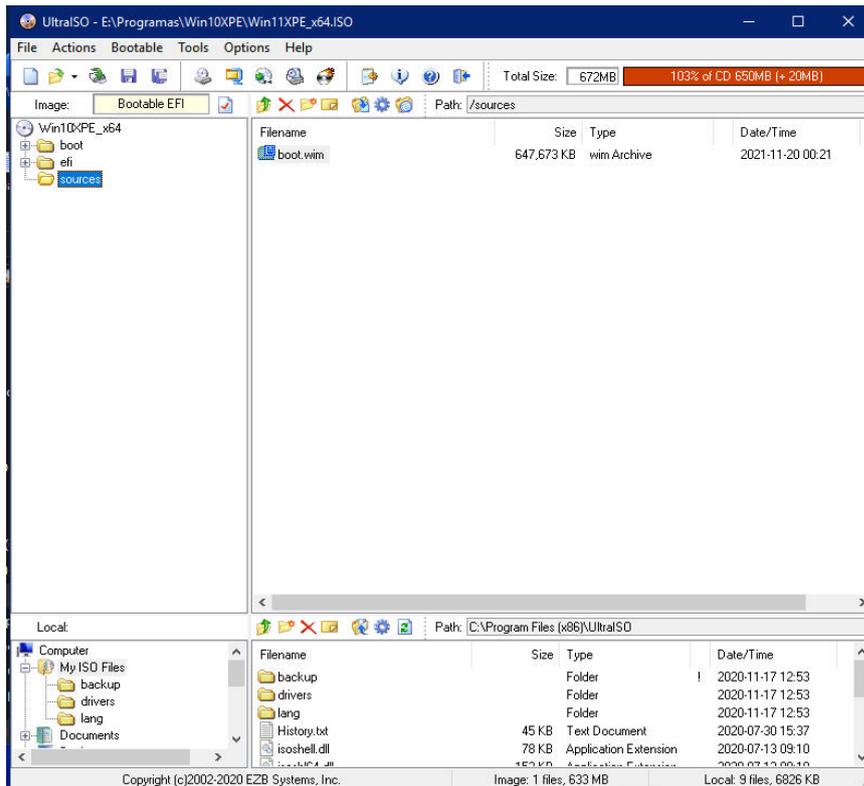
#### j. Change initial boot.wim to the new one created with Dism ++

It can be done in several ways, I do it with **Ultraiso** because it verifies the changes and creates .md5 files, for greater security. If we have Ultraiso installed in our host OS we open it and start, if we do not have it we can use Win10 PE Strelec.

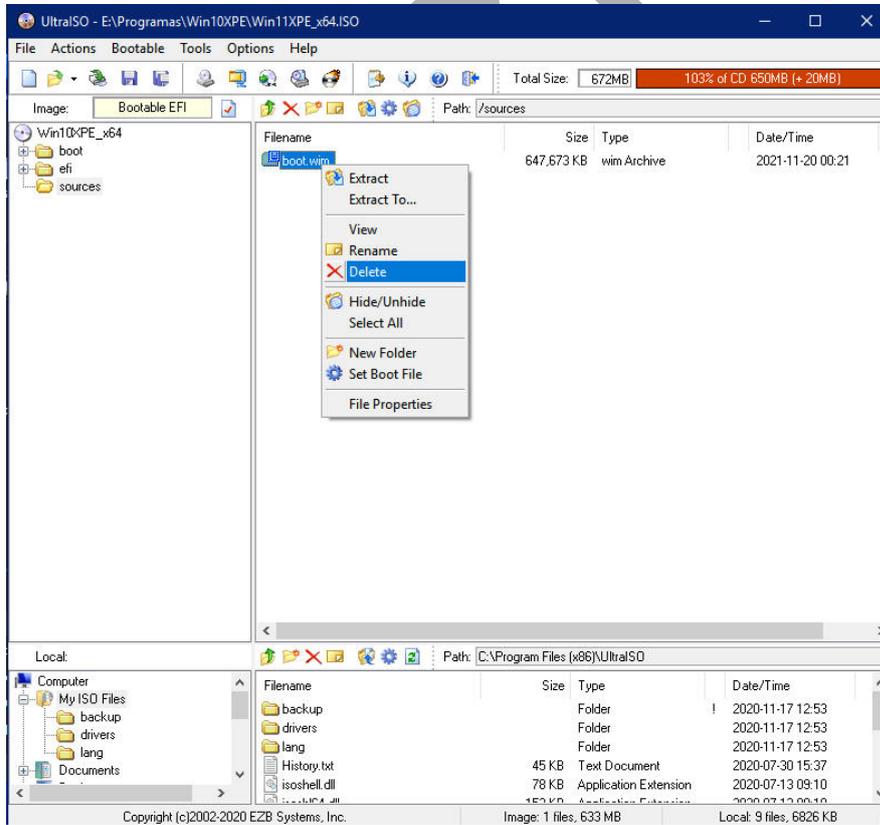
We look for the ISO image of the initial Win11XPE and open it:



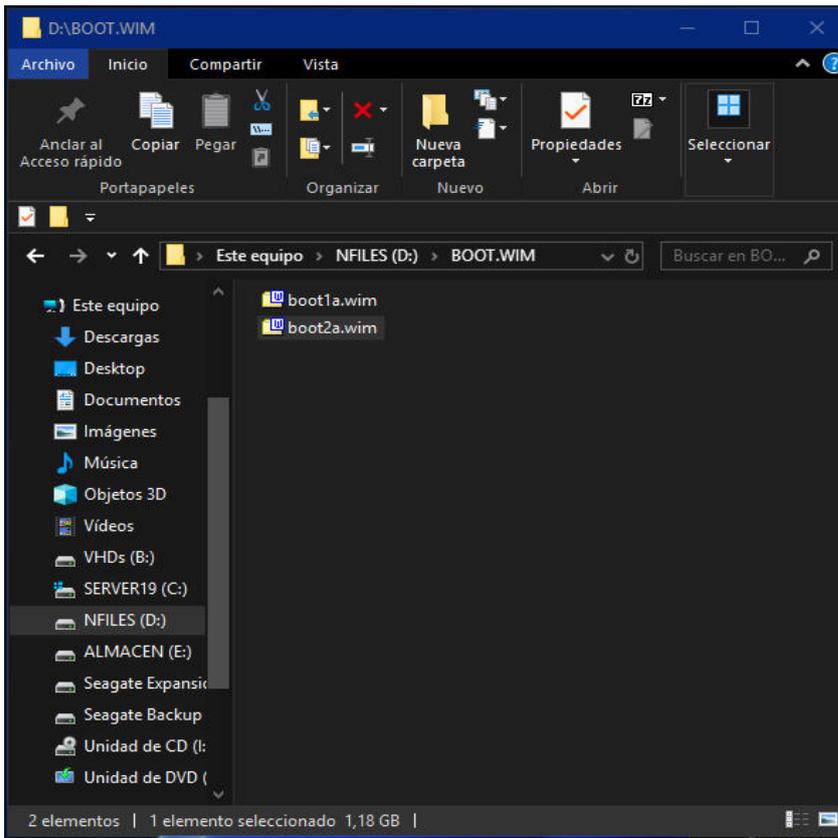
We go to the **sources** folder, and select **boot.wim**



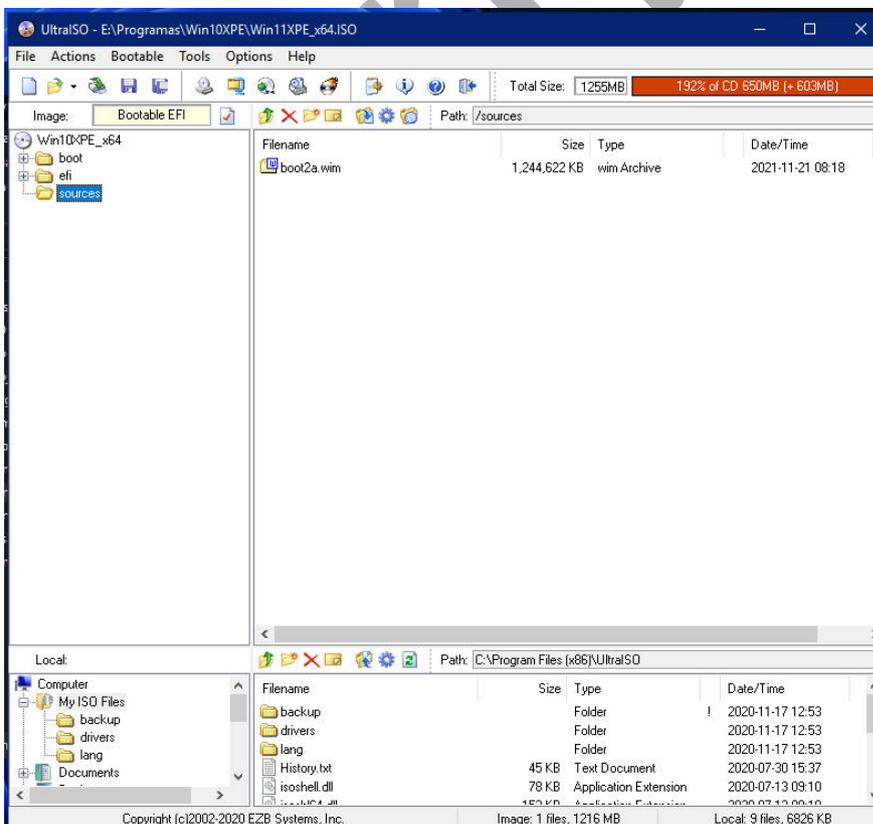
we delete the old file



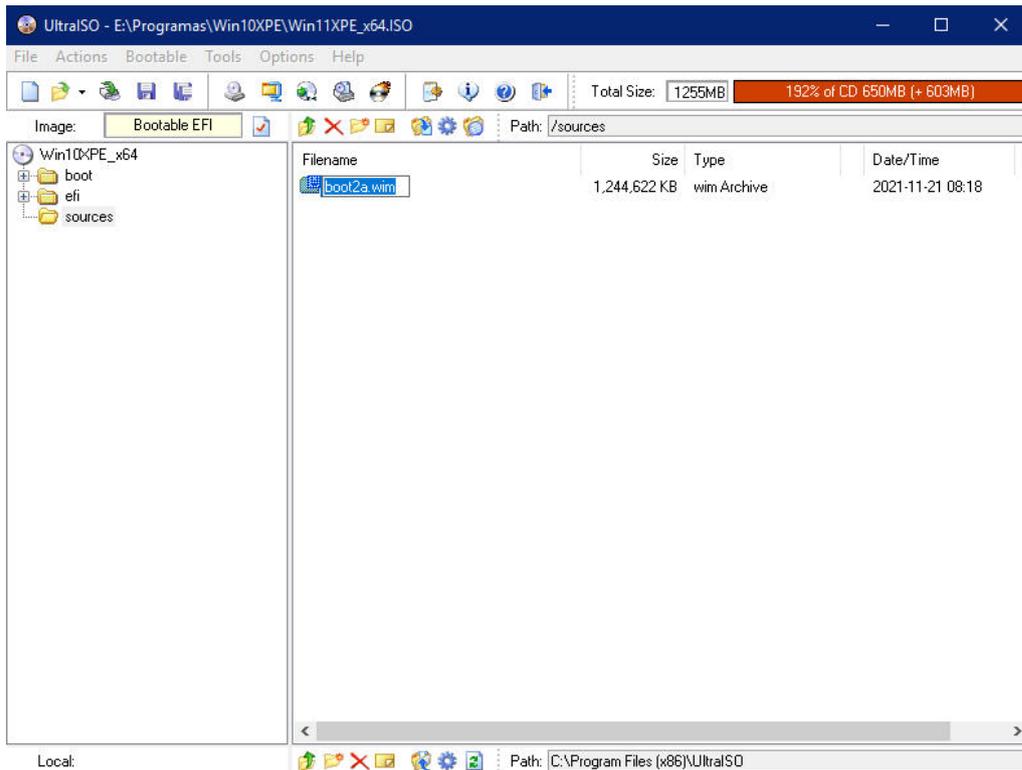
Now we go to the folder where we have saved the new boot.wim and select it



and we **copy** it into the sources folder of the open ISO



To finish we **rename** the new copied file.wim as boot.wim



Finally we rename the ISO label and save the newly modified file, with the name we want.

We already have our brand new Win11XPE.iso just created and functional with all the applications and drivers installed and working.

Now it only remains to transfer the ISO to a USB Pen, with Rufus or similar, restart the PC with it and check that everything works as expected.

Enjoy the music!!

## Acknowledgments:

To Wonko the Sane, and Alacrán from the reboot.pro forum for their help and answers.

To BernieK of this forum, for his wonderful guide to WS2019 Optimization ...

To seeteeyou and Internethandler, of this forum, for their ideas and suggestions to decrease the number of active services and processes.

To ChriSR for making the Win10XPE program.

To my nephew Alberto for reviewing and correcting the grammatical errors in this guide.

And finally, to my wife, for putting up with me all day glued to the PC with my gadgets.

SamotC