

Windows Drivers Workshop

Installation Guide

To conduct the Windows Drivers Workshop, a series of tools will be required to execute the exercises. This detailed installation guide will be provided for each of these tools. The process will outline the necessary steps and requirements for installing each tool, along with corresponding download links.

BIOS Settings

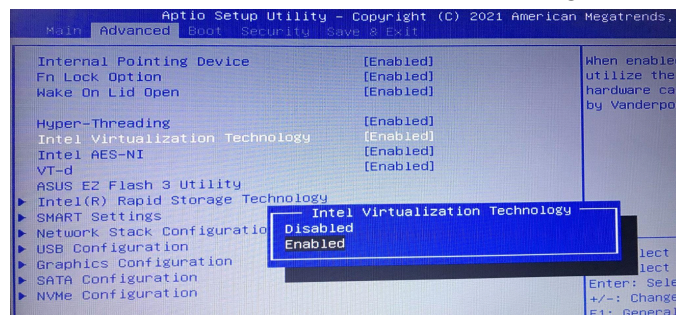
The following BIOS settings on our system will be necessary to enable virtualization and kernel debugging.

On most computers, to access the BIOS, you typically need to do the following:

- At startup, before Windows begins to load, you need to press (often repeatedly) one of the following keys: **F2**, **F10**, **DEL**, or **ESC**.
- If you have pressed the correct key designated by the motherboard manufacturer, you should see the BIOS setup screen.

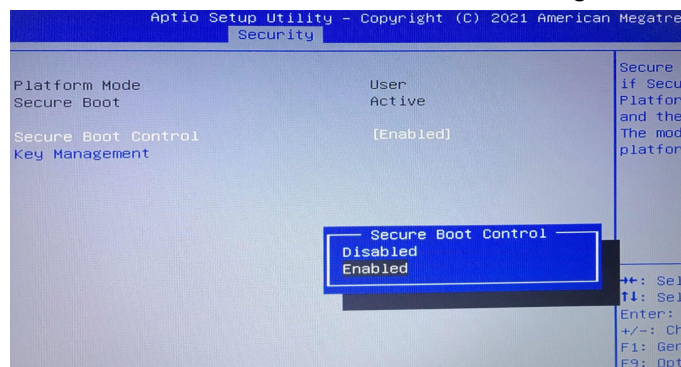
Virtualization

We need to enable virtualization in the BIOS and save the changes:



Secure Boot

We need to disable Secure Boot in the BIOS and save the changes:



Hyper-V on Windows 10/11

Hyper-V is used to create virtual machines in Windows 10/11. By default, Hyper-V is disabled in Windows, so you need to enable it for it to appear in our programs.

Requirements

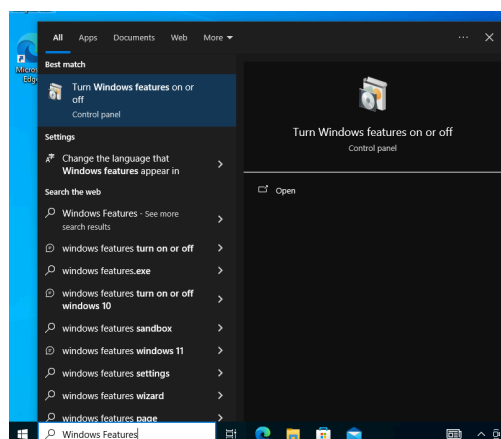
Below are the requirements to run Hyper-V:

- Windows 10 Enterprise, Pro or Education.
- 64-bit processor with Second Level Address Translation (SLAT) capability..
- CPU compatibility for Virtual Machine Monitor Mode Extension (VT-c on Intel CPUs).
- At least 4 GB of main memory (RAM) is required.

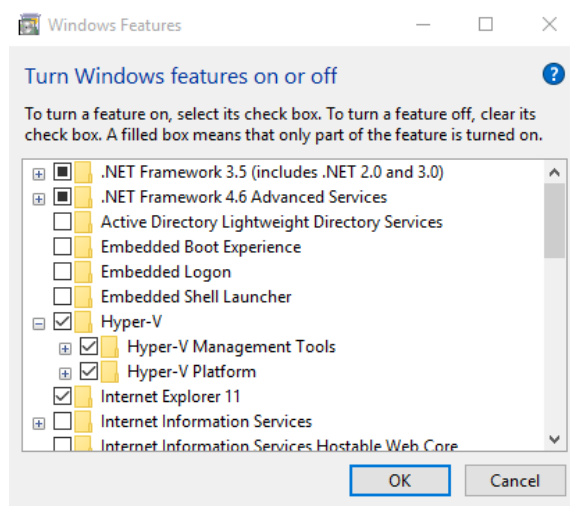
There are several ways to enable the optional Hyper-V feature. This guide will demonstrate how to enable it using the Settings interface:

Enabling the Hyper-V role in Settings

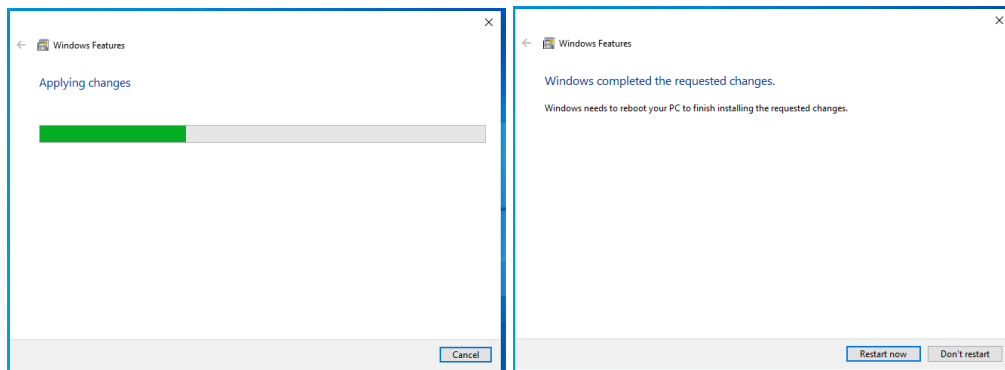
1. In the search bar, type 'Windows Features' and select [Turn Windows features on or off](#).



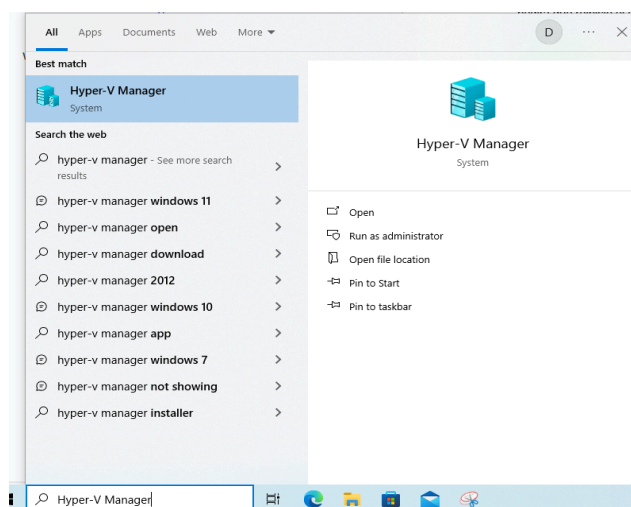
2. You will see the Windows Features window. Inside it, look for the feature called 'Hyper-V'. You need to enable it by clicking the checkbox next to its name. Confirm by clicking OK or Apply.



- Now, the changes will begin to apply. Please wait until the installation process completes. Once the installation is finished, you will be prompted to restart your computer to see the applied changes.



After restarting your computer, you'll be able to access Hyper-V. To do this, you can type 'Hyper-V Manager' in the search bar and select the program to open it.



Other ways to enable Hyper-V

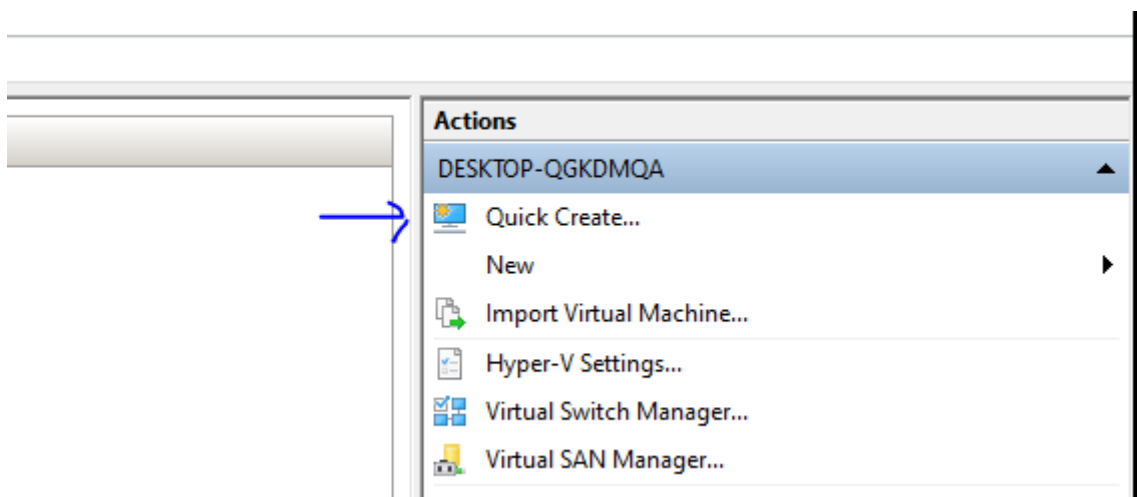
If you want to enable Hyper-V through PowerShell or CMD using DISM, you can find detailed instructions in the official Microsoft documentation at the following link:

<https://learn.microsoft.com/es-es/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>

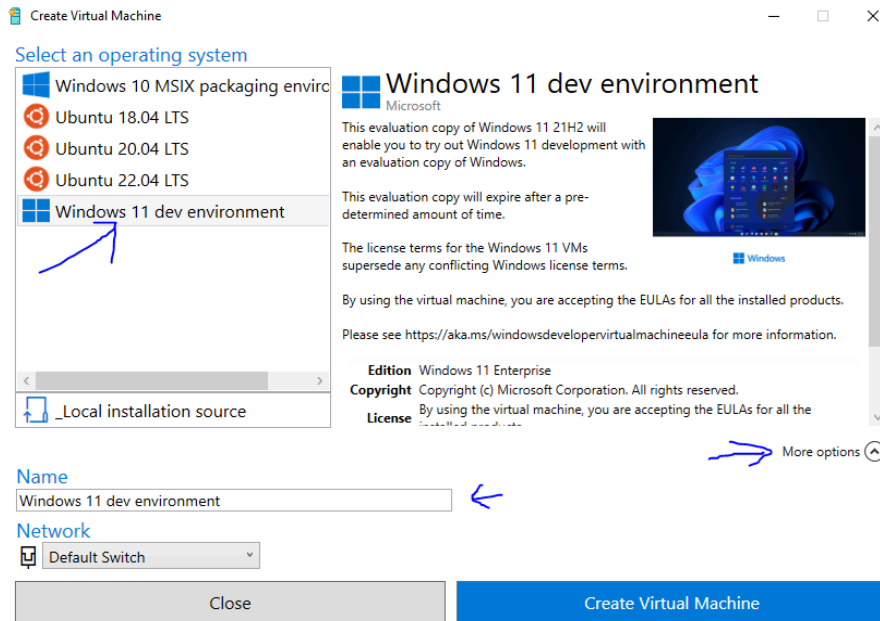
Option 1: Create a virtual machine using Hyper-V quick options (Windows 11)

Hyper-V offers several ways to create a virtual machine, including one called Quick Create. To do this, open Hyper-V Manager and on the right-hand side, you'll see a column titled **Actions**.

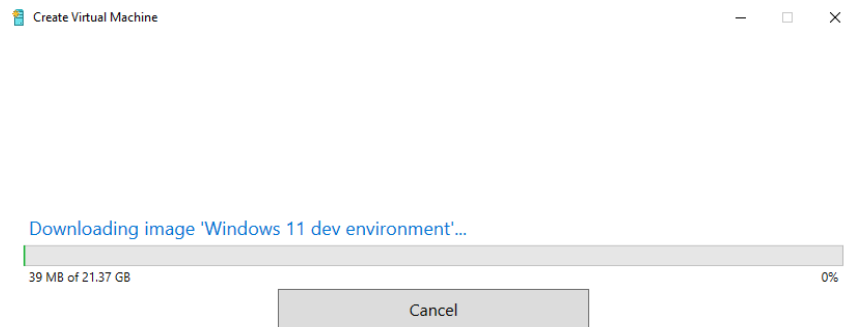
Click on the Action called 'Quick Create...':



Next, in the Create Virtual Machine window, select '**Windows 11 dev environment**'. Then, click on 'More options' to assign a name to your virtual machine. Finally, to start the download, click the '**Create Virtual Machine**' button:



Download process:



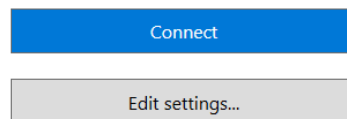
When the download process is complete, we will see the option to Connect to the virtual machine:



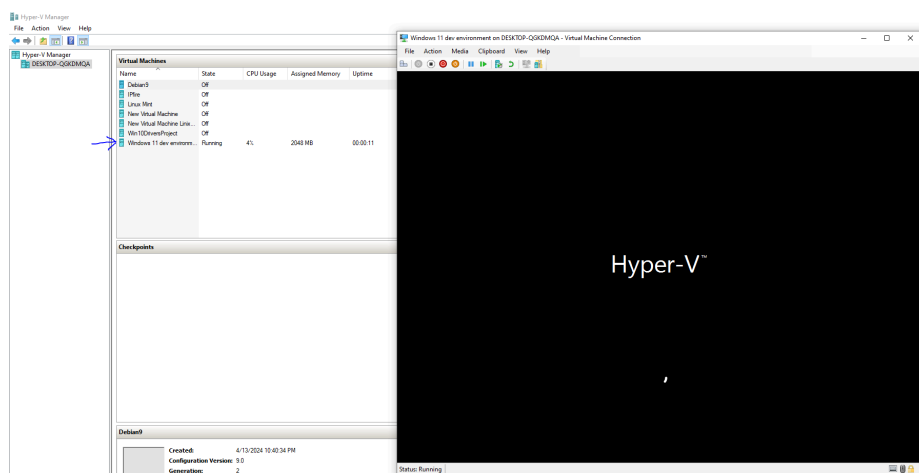
Virtual machine created successfully



Windows 11 dev environment



When you click Connect, the virtual machine will start booting into Windows 11.



Option 2: Create a virtual machine with a Windows 10 ISO image

For the workshop, we will use Windows 10/11 as the operating system to install and test drivers. Therefore, we will need an ISO image of the operating system, which will be used within the virtual machine.

The following link contains a YouTube video demonstrating the creation of a virtual machine in Hyper-V: <https://youtu.be/fyJRVjOv2wM>

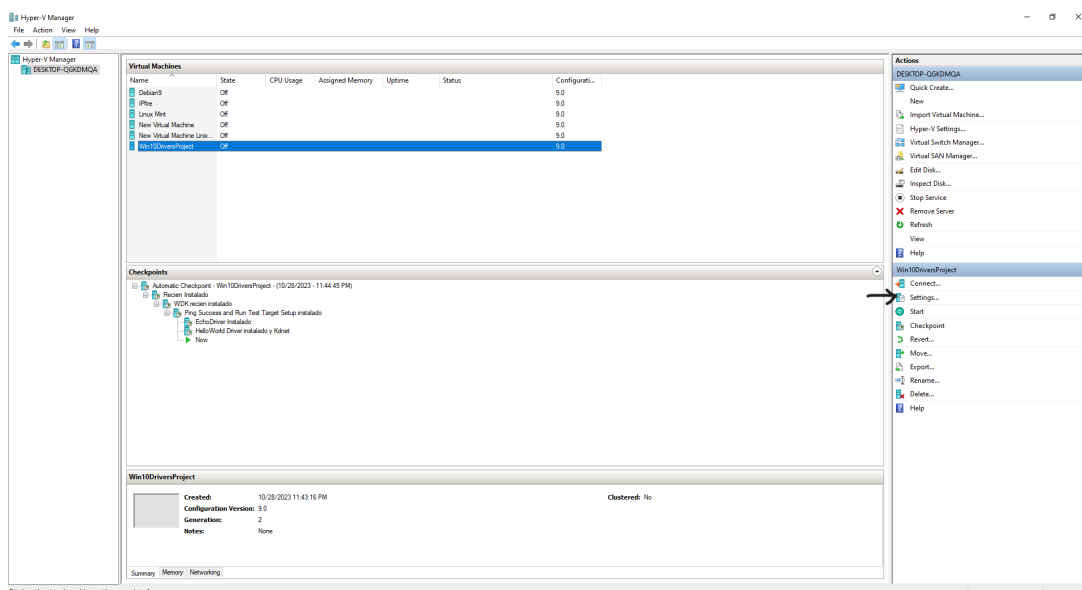
Microsoft provides documentation on creating the virtual machine, which you can consult at the following link:

<https://learn.microsoft.com/es-es/virtualization/hyper-v-on-windows/quick-start/create-virtual-machine>

Enable/Disable Secure Boot in the Virtual Machine

To disable Secure Boot on our virtual machine, we first need to ensure that the virtual machine is powered off (state is 'off') so that it allows us to change the configuration.

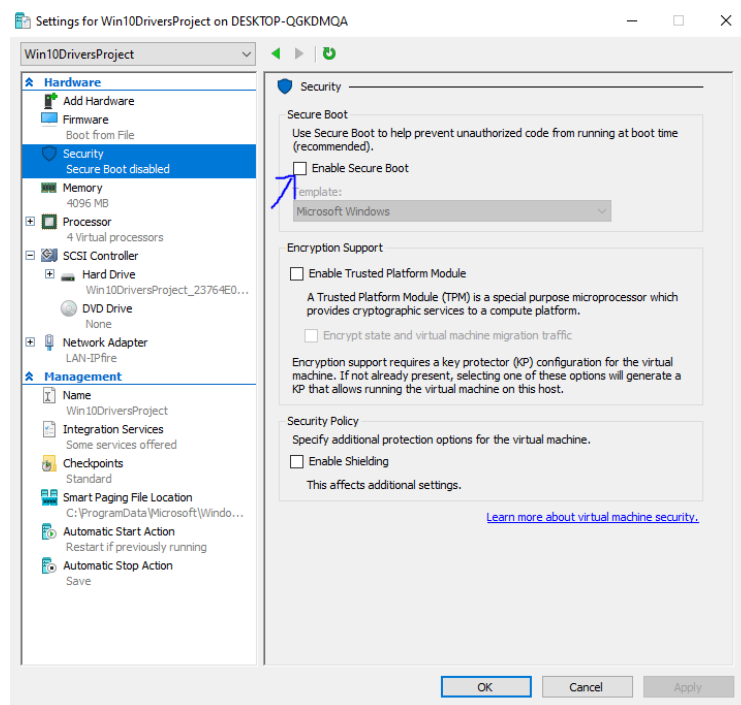
Next, select the virtual machine you want to modify. Once selected, in the right-hand column of Hyper-V Manager, you'll find a section titled with the name of the virtual machine you selected, followed by a series of options to click. Among the options displayed, click on 'Settings...':



You will see the window for configuring your virtual machine settings. Within the settings, on the left-hand side, there is a column divided into **Hardware** and **Management** sections.

In the Hardware section, you will find an option called Security. Select it, and on the right-hand side, you'll see a checkbox labeled Enable Secure Boot.

Select it to enable or disable Secure Boot. Then, apply and save the changes.

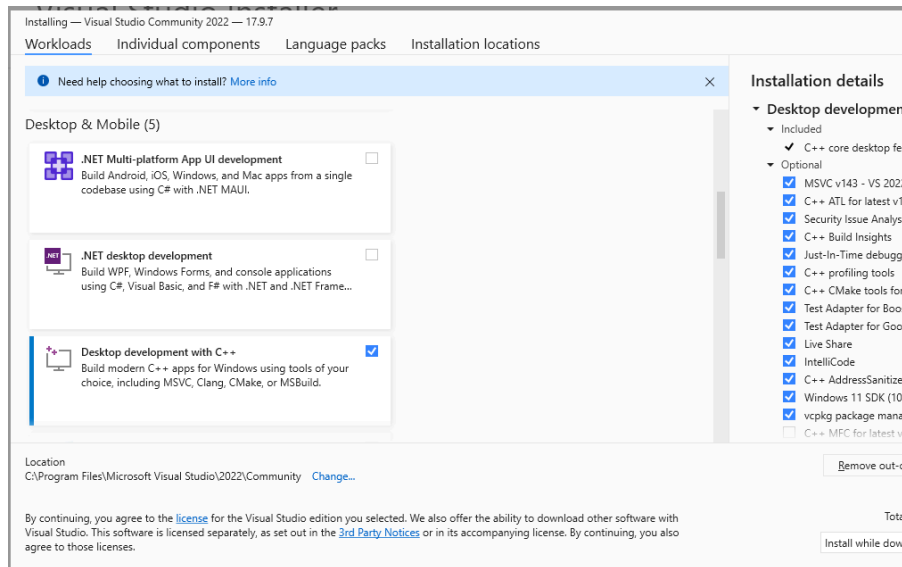


Visual Studio

We will use the Visual Studio 2019 or later development environment. The Community, Enterprise, or Professional editions of Visual Studio support driver development, so you can choose any of these. Below is the official link to download Visual Studio: <https://visualstudio.microsoft.com/es/>

Installation of Visual Studio and Required Components

After downloading the executable, ensure that during the installation process, under the Workloads tab, you have selected the following: **Desktop development with C++**.



Next, in the Individual components tab, we need to search for and select the following components:

- MSVC v143 - VS 2022 C++ ARM64/ARM64EC Spectre-mitigated libs (Latest)
- MSVC v143 - VS 2022 C++ x64/x86 Spectre-mitigated libs (Latest)
- C++ ATL for latest v143 build tools with Spectre Mitigations (ARM64/ARM64EC)
- C++ ATL for latest v143 build tools with Spectre Mitigations (x86 & x64)
- C++ MFC for latest v143 build tools with Spectre Mitigations (ARM64/ARM64EC)
- C++ MFC for latest v143 build tools with Spectre Mitigations (x86 & x64)

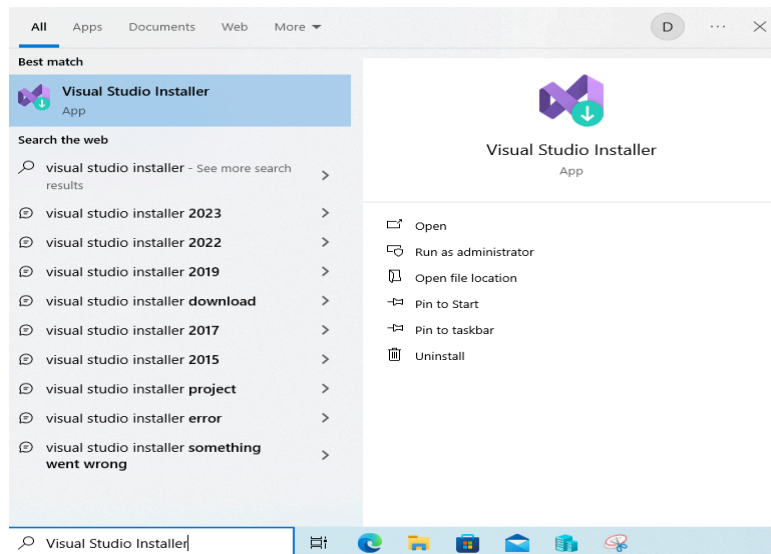


After selecting the components, click the Install button. Once Visual Studio installation is complete, restart your computer.

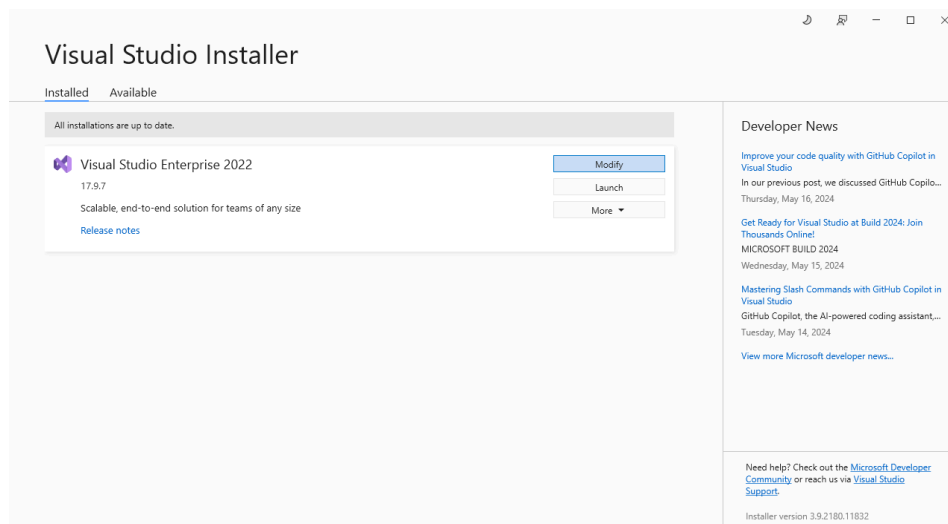
Components Installation

This section only applies to those who already had Visual Studio installed on their computers. The steps are very similar to those shown in the '[Installation of Visual Studio and Required Components](#)' section.

In this case, in the Windows Start menu search bar, type '**Visual Studio Installer**' and open the Visual Studio Installer.



You will see the Visual Studio Installer window, displaying the version of Visual Studio installed on your computer along with other details. Click the 'Modify' button to access the section shown in the previous point. Once there, you can check if the components listed in the '**Installation of Visual Studio and Required Components**' section are installed or not. If they are not installed, follow the same steps detailed in this section to install the required components for the workshop.



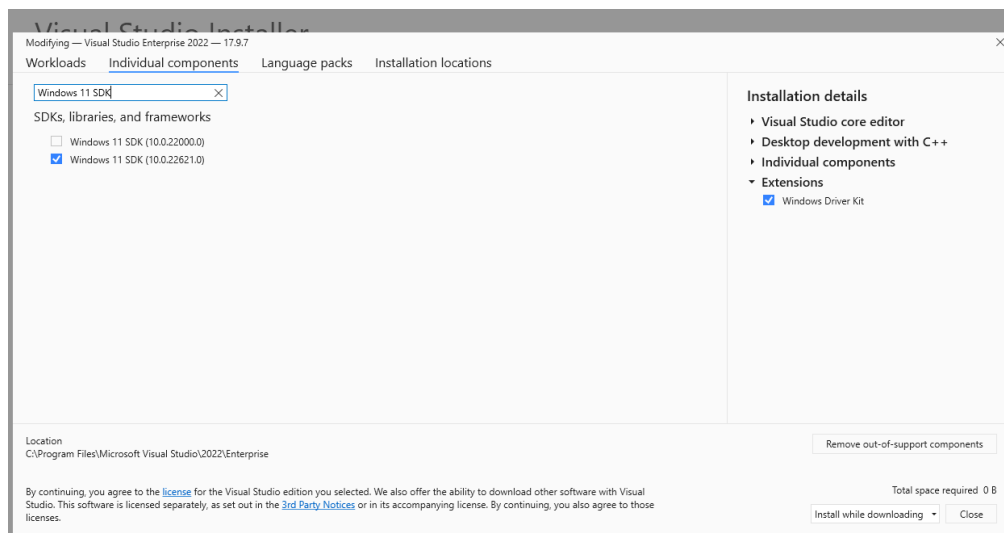
SDK and WDK Installation

Important: The Build numbers of the SDK and WDK must match for them to function correctly.

SDK (Software Development Kit)

If you have installed Visual Studio with the '**Desktop development with C++**' option, then the SDK should already be installed. This option internally includes the installation of the SDK.

To verify the installation of the SDK on your computer, you can open the Visual Studio Installer and click on the 'Modify' button. Next, in the 'Individual components' tab, search for 'Windows 11 SDK'.



If it is installed, you should see the installed version of the SDK with a checkmark next to it, along with its Build number.

If it is not installed, you can install it directly from Visual Studio within the Individual components tab, or separately by downloading it from the official Microsoft documentation at the following link:

<https://developer.microsoft.com/en-us/windows/downloads/windows-sdk/>

WDK (Windows Driver Kit)

To download the WDK, you can follow the documentation provided by Microsoft:

<https://learn.microsoft.com/en-us/windows-hardware/drivers/download-the-wdk>

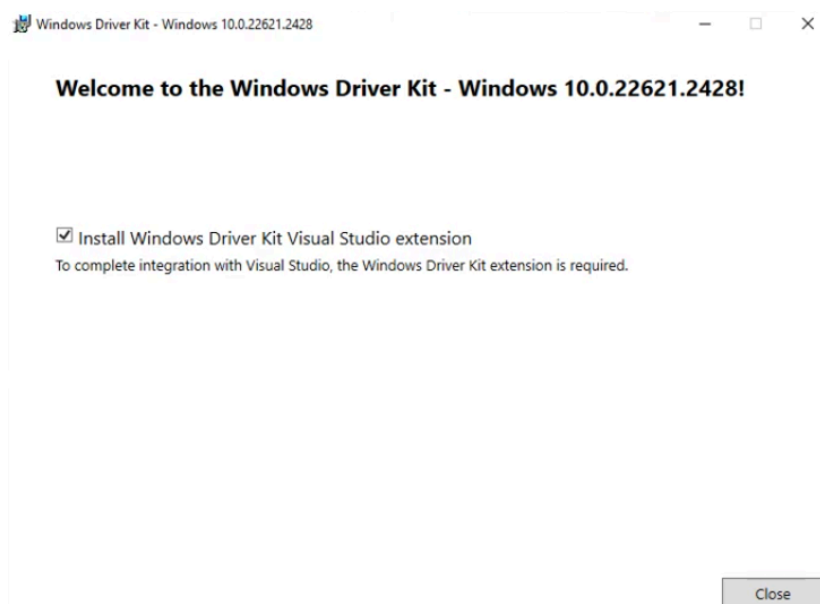
In the documentation, step 3 contains the download link. Remember to verify that the Build version matches the one you have for the SDK.



Step 3: Install WDK

- [Download WDK 10.0.22621](#)

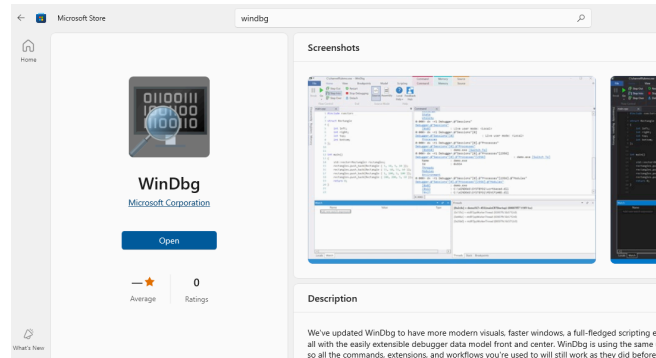
During the installation process, check the box that indicates to install the WDK extension for Visual Studio:



WinDBG

You can download WinDBG from the Microsoft Store:

<https://apps.microsoft.com/detail/9pgjgd53tn86?hl=es-cr&gl=CR>



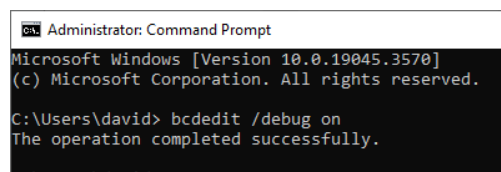
Optional: Local Kernel Debugging

To set up Local Kernel Debugging, Secure Boot must be disabled (this is changed directly from the BIOS).

Note: When disabling Secure Boot, whether on a virtual or physical machine, remember that if it's disabled on your physical machine for the workshop, you should re-enable it from the BIOS after completing the necessary activities.

With Secure Boot disabled, restart the computer and access a command line interface, such as Command Prompt (CMD) with administrator privileges. Enter the following command needed to configure Local Kernel Debugging:

`bcdedit /debug on`



Next, restart the computer for the changes to take effect. Now, you can connect to Local Kernel Debugging in WinDbg:

