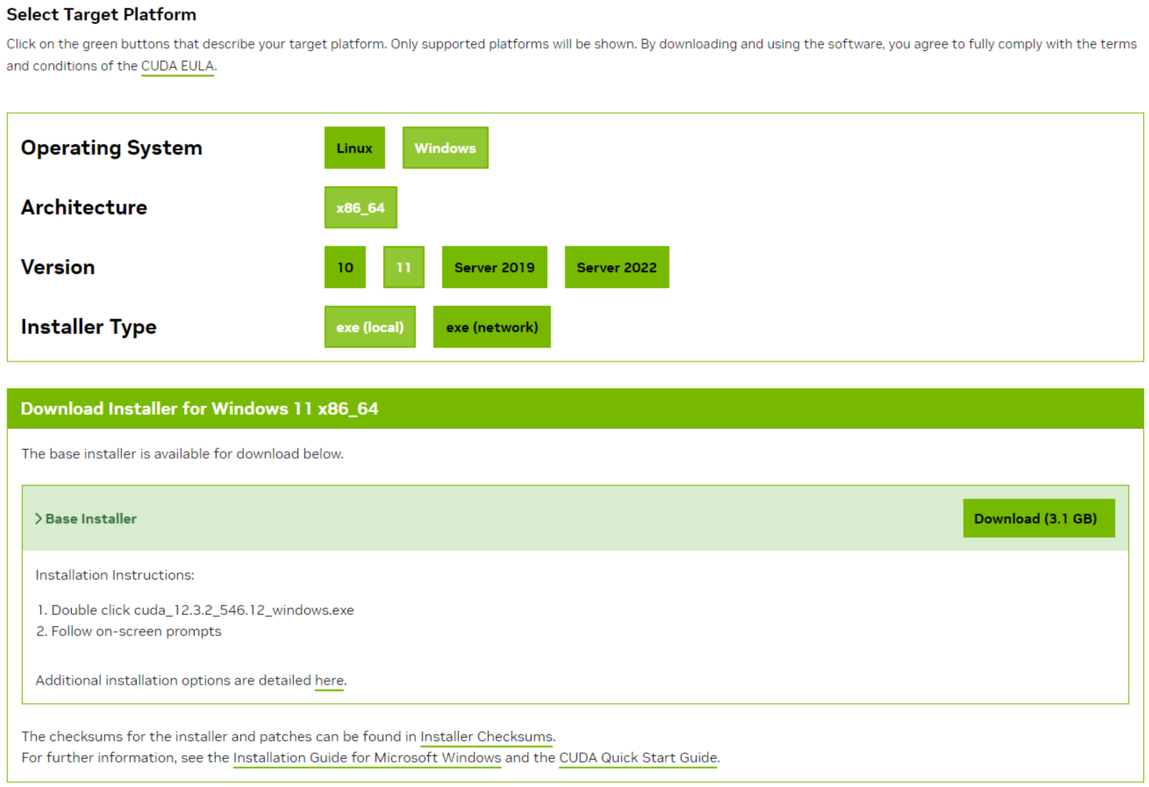
CUDA Installation Steps

1. Go to the below link and download CUDA toolkit 12.3.

[CUDA Toolkit 12.3 Update 2 Downloads | NVIDIA Developer]

2.



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the CUDA EULA.

Operating System: Linux, Windows

Architecture: x86_64

Version: 10, 11, Server 2019, Server 2022

Installer Type: exe (local), exe (network)

Download Installer for Windows 11 x86_64

The base installer is available for download below.

> Base Installer — Download (3.1 GB)

Installation Instructions:

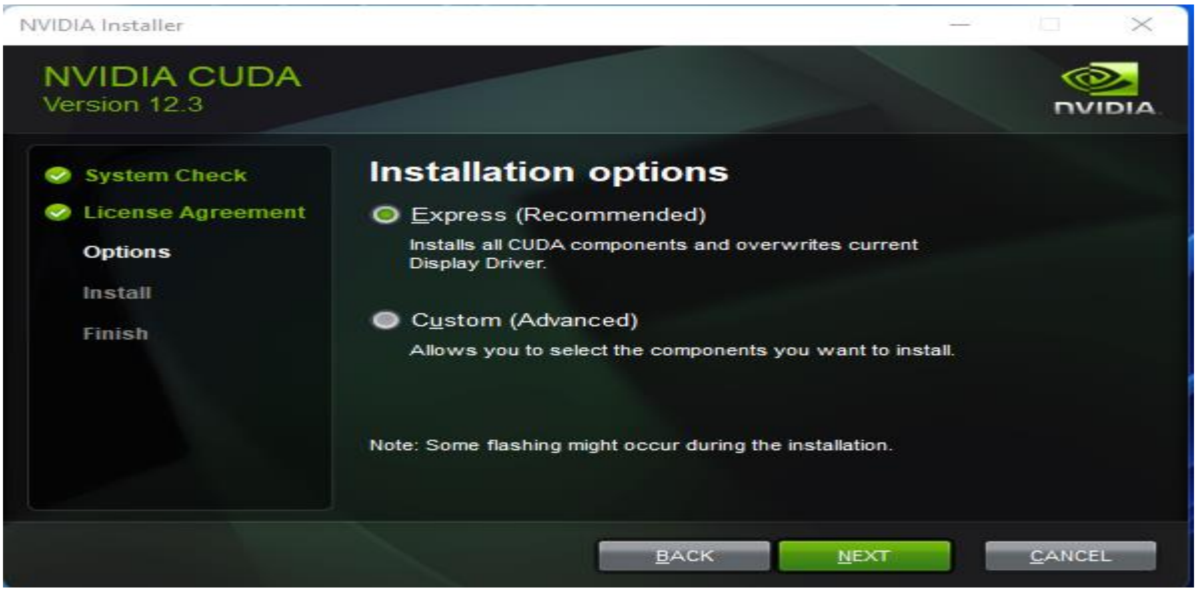
1. Double click cuda_12.3.2_546.12_windows.exe
2. Follow on-screen prompts

Additional installation options are detailed here.

The checksums for the installer and patches can be found in Installer Checksums.
For further information, see the Installation Guide for Microsoft Windows and the CUDA Quick Start Guide.

3. Once downloaded run the exe file

4.



Installation options

● Express (Recommended)

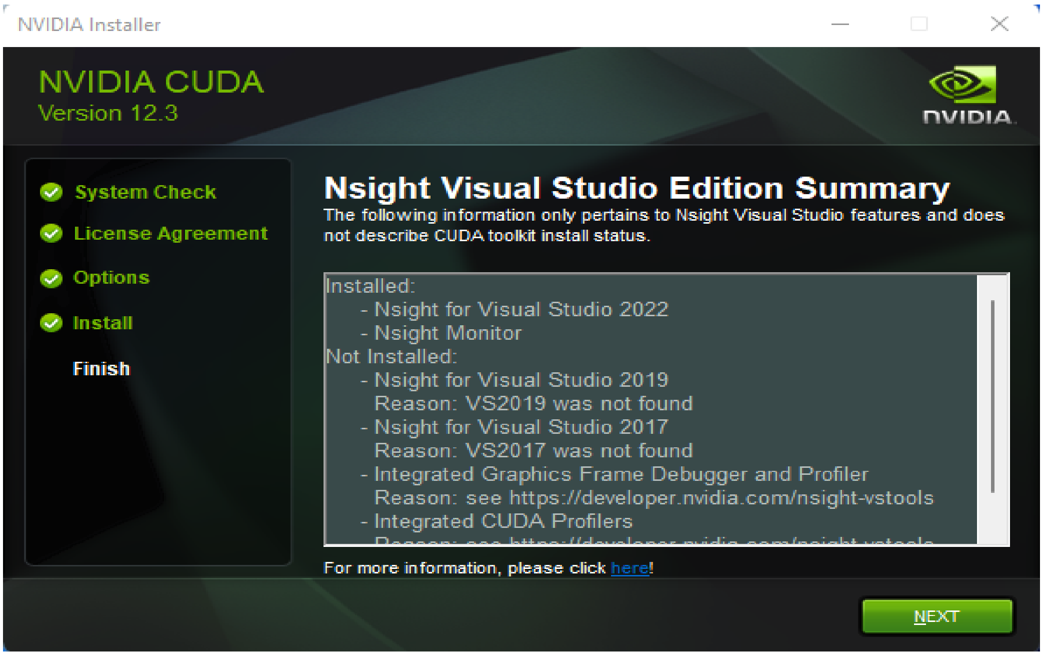
Installs all CUDA components and overwrites current Display Driver.

○ Custom (Advanced)

Allows you to select the components you want to install.

Note: Some flashing might occur during the installation.

5.



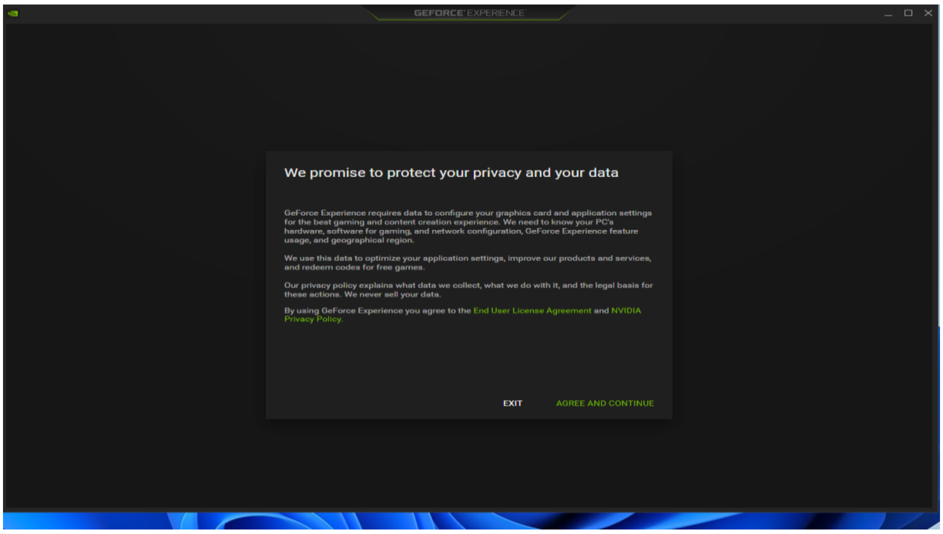
Nsight Visual Studio Edition Summary

The following information only pertains to Nsight Visual Studio features and does not describe CUDA toolkit install status.

```
Installed:
    - Nsight for Visual Studio 2022
    - Nsight Monitor
Not Installed:
    - Nsight for Visual Studio 2019
      Reason: VS2019 was not found
    - Nsight for Visual Studio 2017
      Reason: VS2017 was not found
    - Integrated Graphics Frame Debugger and Profiler
      Reason: see https://developer.nvidia.com/nsight-vstools
    - Integrated CUDA Profilers
      Reason: see https://developer.nvidia.com/nsight-vstools
```

For more information, please click here!

6.



We promise to protect your privacy and your data

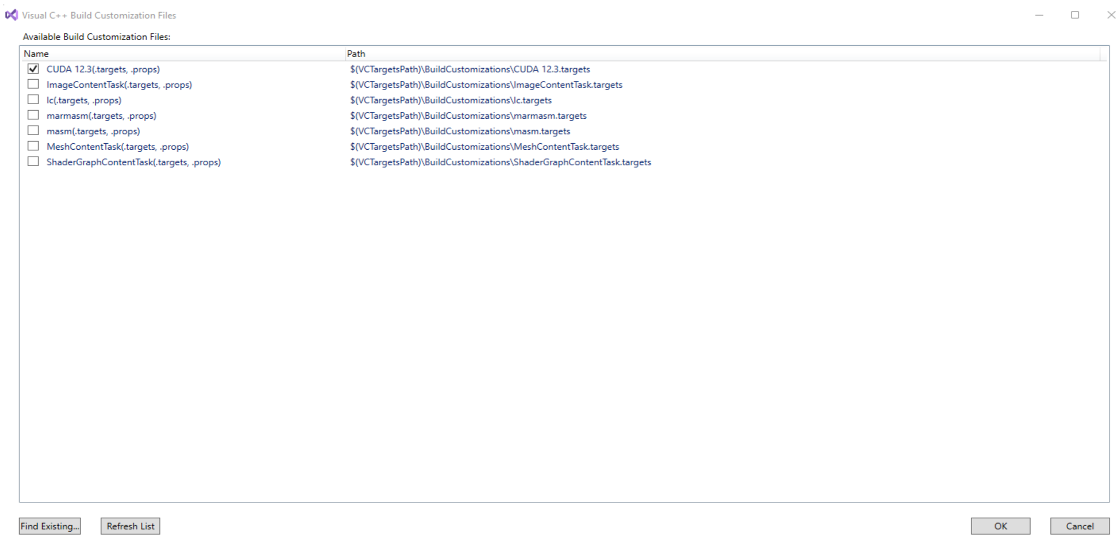
GeForce Experience requires data to configure your graphics card and application settings for the best gaming and content creation experience. We need to know your PC's hardware, software for gaming, and network configuration, GeForce Experience feature usage, and geographical region.

We use this data to optimize your application settings, improve our products and services, and redeem codes for free games.

Our privacy policy explains what data we collect, what we do with it, and the legal basis for these actions. We never sell your data.

By using GeForce Experience you agree to the End User License Agreement and NVIDIA Privacy Policy.

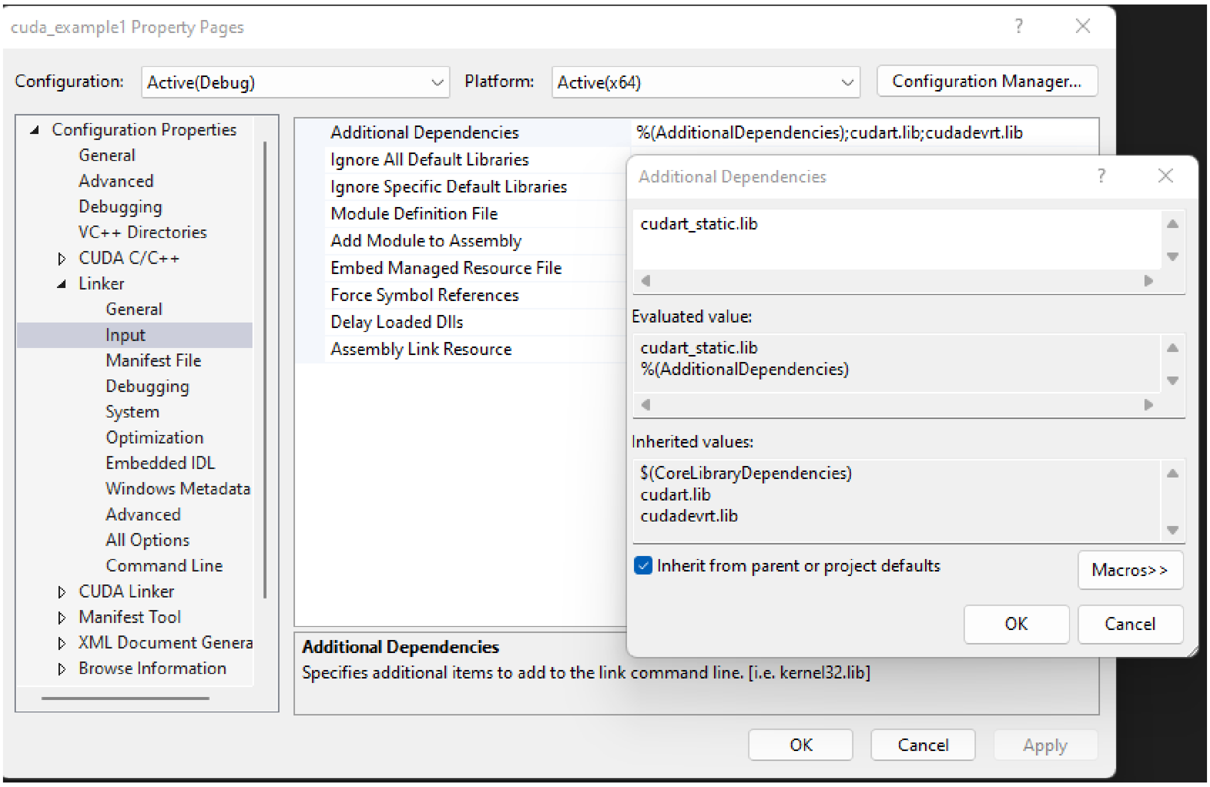
7. Once installed, open Visual studio 2022
8. Create a new project -> Empty project -> Project_name
9. Right click on Source File -> Add item -> New item -> CUDA 12.3 C/C++ File
10. Right click on Project -> Build dependencies -> build customizations
11.



Available Build Customization Files:

Name	Path
☑ CUDA 12.3(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\CUDA 12.3.targets
□ ImageContentTask(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\ImageContentTask.targets
□ lc(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\lc.targets
□ marmasm(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\marmasm.targets
□ masm(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\masm.targets
□ MeshContentTask(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\MeshContentTask.targets
□ ShaderGraphContentTask(.targets, .props)	$(VCTargetsPath)\BuildCustomizations\ShaderGraphContentTask.targets

12. Select properties-> linker-> input-> additional Dependencies-> add cudart_static.lib



Configuration: Active(Debug)

Platform: Active(x64)

Additional Dependencies: %(AdditionalDependencies);cudart.lib;cudadevrt.lib

Additional Dependencies

Specifies additional items to add to the link command line. [i.e. kernel32.lib]

Additional Dependencies

```
cudart_static.lib
```

Evaluated value:

```
cudart_static.lib
%(AdditionalDependencies)
```

Inherited values:

```
$(CoreLibraryDependencies)
cudart.lib
cudadevrt.lib
```

☑ Inherit from parent or project defaults

13. Sample Test Program

```
#include "cuda_runtime.h"
#include "device_launch_parameters.h"

#include <stdio.h>

__global__ void add_arrays_gpu(int* a, int* b, int* c)
{
        c[threadIdx.x] = a[threadIdx.x] + b[threadIdx.x];
}

int main()
{
        const auto count = 5;
        int host_a[] = { 1, 2, 3, 4, 5 };
        int host_b[] = { 10, 20, 30, 40, 60 };
        int host_c[count];

        int* device_a, * device_b, * device_c;
        const int size = count * sizeof(int);
        cudaMalloc(&device_a, size);
        cudaMalloc(&device_b, size);
        cudaMalloc(&device_c, size);

        cudaMemcpy(
                device_a, host_a,
                size,
                cudaMemcpyHostToDevice
        );

        cudaMemcpy(
        device_b, host_b,
        size,
        cudaMemcpyHostToDevice
    );

    add_arrays_gpu << <1, count >> > (device_a, device_b, device_c);

    cudaMemcpy(
        host_c, device_c,
        size,
        cudaMemcpyDeviceToHost
    );


    for (auto i = 0; i < count; i++)
    {
        printf("%d ", host_c[i]);
    }

    getchar();

    return 0;
}
```