I2D GPU Installation Guide

For Rhino 8 on Windows

Version 1.0.4 English edition 14 October 2026

This guide takes you from a new Windows installation to a working I2D GPU depth engine and a first relief mesh in Rhino. Complete the Python installation before preparing the engine inside Rhino. Use the Quick Start User Manual for your first project and the full User Manual for detailed controls.

Python must be installed separately. The I2D GPU installer installs the Rhino plugin and accompanying files. The panel's **Install / repair GPU engine** button uses an existing compatible Python installation to create the managed environment, install its packages, and prepare models. It does not install the base Python application for you.

What you need

Requirement	What to prepare
Windows and Rhino	A Windows x64 computer with Rhino 8 installed and working. Run Rhino at least once before installing the plugin.
Python	Full 64-bit Python 3.12 or 3.11 with pip and the Python launcher. Page 2 explains the version choice and installation.
GPU acceleration	An NVIDIA GPU supported by the supplied CUDA PyTorch runtime, with a compatible NVIDIA driver. CPU operation is available for smaller models when selected explicitly.
Internet and storage	Internet access for packages, selected models, and online licensing. Allow space for the managed environment, downloaded models, and caches. Larger models require more storage and memory.
Plugin access	A valid trial or purchased license for the operations you want to use. Runtime setup and license activation are separate steps.

Installation order

1. Install compatible Python and confirm it starts on page 2.
2. Close Rhino and install the I2D GPU package on page 3.
3. Open the panel and activate access on page 3.
4. Prepare the engine and generate a test mesh on page 4.
5. Use page 5 if any step fails. Page 6 covers updates and support.

Use the same Windows account throughout. The normal plugin installation and managed runtime are stored in that user's profile. Start with a small test before loading demanding settings.

Install Python first

Choose a compatible version

Use full 64-bit **Python 3.12 or 3.11**. Start with the **Python 3.12 64-bit Windows installer** for the current package. The Python 3.12.10 release page linked below provides that full installer. The current I2D engine setup checks for these two Python version families.

Package specific requirement. Some compiled pipeline packages require Python 3.11. If your supplied package explicitly requires 3.11, follow its version requirement instead. Python 3.11.9 has an official 64-bit Windows installer. A version mismatch reported by the engine needs the matching package and runtime.

Run the full Windows installer

1. Open the Python 3.12 release page below. Under **Files**, choose **Windows installer (64-bit)**. Choose the full installer rather than the embeddable package.
2. Run it. Select **Add python.exe to PATH** and keep the **Python launcher** and **pip** enabled. Use the normal default installation location.
3. Choose **Install Now**, or customize the installation while retaining the interpreter, standard library, pip, and launcher. Finish the setup.
4. Close any open Rhino sessions and command windows. Open a fresh Command Prompt so it receives the updated environment.

Verify before continuing

Run **py -3.12 --version**. It should print a Python 3.12 version. Run **py -3.12 -m pip --version** to check pip.

To check the interpreter's architecture, run:

py -3.12 -c "import struct; print(struct.calcsize('P') * 8)"

The result should be **64**. For a package using Python 3.11, replace **-3.12** with **-3.11** in these checks. If the launcher is unavailable, run the installed python.exe using its full path to check the same version, pip, and architecture.

Keep Python installed. The managed environment is created from it and may depend on that base installation. Rhino's script editor having Python support is not this separate prerequisite. Let I2D install its own tested AI packages in the managed environment.

Official Python 3.12 Windows installer release page

https://www.python.org/downloads/release/python-31210/

Official Python 3.11 Windows installer release page

https://www.python.org/downloads/release/python-3119/

Install and activate the plugin

Install the Rhino package

1. Open Rhino 8 once if you have just installed it, then close all Rhino windows.
2. Obtain the current **I2D GPU for Rhino 8** installer from the product page or the supplied release package. Use the GPU edition. Installer filenames normally begin **I2D_GPU_Plugin_Rhino8_Setup**; the version suffix can change.
3. Run the installer under the Windows account you use for Rhino. Choose its language, read the license and third party notices, and complete installation.
4. Keep the default Rhino 8 plugin location unless your supplied package instructs otherwise. The normal installer uses a per-user location and does not require an administrator account.
5. Restart Rhino 8, open or create a document, and run **I2DGPUOpenPanel** in the Rhino command line.

The default installed folder is:

%APPDATA%\McNeel\Rhinoceros\8.0\plug-ins\I2D_GPU_Plugin

Paste that path into the Windows Explorer address bar to open your own user folder. Keep I2D_GPU_Plugin.rhp, its supporting DLLs, and the pipeline folder together. Copying only the RHP does not provide the complete depth engine.

If Rhino does not recognize the command

Run **PlugInManager**, or open **Tools > Options > Plug-ins**. Find I2D GPU and check its load status. Use **Load plug-in** when available. If it is absent, use **Install…** and choose **I2D_GPU_Plugin.rhp** from the installed folder, then restart Rhino if requested.

If Rhino reports a loading error, retain its exact text and check the complete installer package. Fix plugin loading before attempting Python engine setup. A missing toolbar alone does not prevent opening the panel with the command.

Activate trial or purchased access

Expand **Licensing** and read the current status. Choose **Get trial license** when available, then **Refresh license**. For a purchased license, follow the supplied activation instructions. **Buy a license** opens the configured purchase route; use **Complete purchase** if the panel shows it, then refresh the status.

The available buttons depend on the build and license state. Use **Copy** beside **Hardware ID** if support asks for it. Initial trial issuance and other online licensing operations require connectivity. Follow the validity and check requirements shown by your license.

Product page and current downloads

https://www.i2dplugin.org/i2dgpu.html

Prepare the engine and test it

Prepare the managed environment

1. In **GPU depth engine**, enable **Show advanced GPU depth settings** if needed. For the initial check, choose **Small - vits** or **Auto** in the model list. Auto resolves to Small in the current plugin.
2. Choose **Device: CUDA / GPU** for NVIDIA acceleration. If you are testing CPU operation, choose **CPU** or **Auto** with Small. CPU processing can be much slower.
3. Choose **Install / repair GPU engine**. Allow package installation and model preparation to finish. The panel shows progress and status; the first setup can take longer because files must be downloaded.
4. Choose **Refresh GPU status**. **Ready (CUDA)** confirms the managed runtime can see CUDA. **Ready (CPU only)** reports a CPU-only runtime. If CUDA was requested, a CUDA error must be resolved before that device can run.
5. Use **Download model** if the selected model needs preparation. A different depth model or background model can need additional downloads later.

The engine installer manages its own Python environment and tested packages. Use the NVIDIA driver for your GPU; a separate CUDA Toolkit installation is not part of the normal I2D setup. Engine readiness confirms runtime availability; generating a depth map also checks the selected model and image workflow.

Generate a small test relief

1. In **File & actions**, leave **File is pre-made depth / height map** off. Choose **Browse…** and open a clear photo or artwork.
2. In **Preset**, choose **Open settings…** and load **BR_Universal**. The current set has six BR presets; their selection is explained in both user manuals.
3. After loading the preset, set the model back to **Small - vits** for this initial test. Recheck the device: presets save CUDA and can change the model. On a CPU test, explicitly select CPU or Auto again.
4. Choose **GPU depth quality: Draft - 518** and **Output quality: Draft - 256**. Keep Live mesh off. If testing without background extraction, leave Remove background off.
5. Choose **Update depth map**. Wait for a depth image. Read any download or error status. A successful preview shows the inference stage has run.
6. Check Rhino document units, set a suitable **Target depth**, and choose **Add mesh**. Inspect the resulting geometry in Rhino and save the document.

Installation success. The panel opens, license status permits the test, the engine is ready, a depth image is generated, and Add mesh creates geometry. Use the Quick Start User Manual to refine that first result.

Resolve installation problems

Python cannot be found

Confirm **py -3.12 --version** or **py -3.11 --version** works in a new command window. Install the full 64-bit interpreter with pip and the launcher if it does not. Restart Rhino after installing Python. A newer unrelated Python version alone does not satisfy the current setup's 3.11 or 3.12 check.

If the **python** command opens the Microsoft Store, use the version-specific **py** command to verify your real installation. Keep the default Python location and launcher so the plugin can discover it. If discovery still fails, export diagnostics and provide the version and installed path to support.

Python is present but the runtime is missing or broken

The base Python application and the I2D managed environment are separate. Run **Install / repair GPU engine**, allow it to finish, then refresh the status. Errors mentioning PyTorch or tested package versions need managed-runtime repair. If a custom Python environment is reported, use support to restore the managed configuration; installing random packages into another Python environment will not repair it.

CUDA is unavailable

Confirm the computer has an NVIDIA GPU supported by the supplied runtime. Update its NVIDIA driver, restart Windows if the driver installer requests it, and refresh the GPU status in Rhino. If needed, run engine repair to restore the supplied CUDA PyTorch packages.

For a CPU check, choose Small with CPU or Auto. Recheck the device after loading a preset. An integrated GPU or an AMD GPU does not provide NVIDIA CUDA. Use the exact CUDA error and diagnostics to investigate hardware or driver compatibility.

Package or model downloads fail

Read the status for the failed item. Check internet connectivity, free storage, and any proxy or network restrictions. Ask your network administrator to permit the required Python package and model download services when a managed network blocks them. Retry engine repair for package setup or Download model for the selected model. Use **Cancel GPU** to cancel the current operation before retrying.

The depth step runs out of memory

Choose Small and Draft GPU depth quality. For the first test, keep background extraction off and reduce tiled detail in advanced settings if needed. Final mesh output quality is a separate control: lowering it reduces mesh build cost, but does not by itself lower the AI inference size.

The panel opens but an operation is denied

Read **Licensing**, check connectivity if an online check is required, and use **Refresh license**. Record the exact message if access still fails. Reinstalling Python or the depth engine does not activate or renew a license.

Support evidence. Choose **Export GPU diagnostics…** after a failure and retain the error text. Include the step that failed, Rhino version, plugin version, Python version, GPU and driver, and the chosen model and device.

Updates files and support

Update or repair an installation

Version 1.0.4 adds Custom depth curve, CPU Fast QEM and shared action progress. Keep I2DFastQem.dll with the plugin. Save your work and close Rhino before installing. Back up your personal presets first. Install the complete current package, restart Rhino, refresh licensing and GPU status, and run a small depth-and-mesh test. Check for **Depth map updated** and **Mesh added** in the shared progress bar. Use **Install / repair GPU engine** if the runtime needs preparation after the update.

The normal installer preserves preset files that already have the same filename. An update can therefore leave an older copy of a supplied preset in place. Back up custom changes and compare supplied presets with the new release when you need its updated settings. The current manuals document six presets; an older installation may still contain additional names.

Useful locations

Paste these paths into the Windows Explorer address bar. The percent variables expand to your Windows user's folders. These are default locations; a customized installation may differ.

Item	Default location
Rhino plugin files	%APPDATA%\McNeel\Rhinoceros\8.0\plug-ins\I2D_GPU_Plugin
Presets	%APPDATA%\McNeel\I2D_GPU\presets
Managed runtime	%LOCALAPPDATA%\I2D_GPU\Runtime
Processing cache	%LOCALAPPDATA%\I2D_GPU\Cache

Let the panel prepare its managed runtime on each computer. Back up source images, masks and Rhino documents separately.

Remove the plugin

Close Rhino and use the installed I2D GPU uninstall entry in Windows Installed apps or Apps and features. Python is a separate application. Removing it may affect I2D's environment and other programs that use it. Presets, downloaded runtime files, caches, and project files can remain outside the plugin folder; retain anything you need before cleaning up stored data.

Official references

The Python release links are on page 2; the product download page is on page 3. The references below explain Python installation, Rhino plugin loading, and NVIDIA driver downloads.

Python Windows installation documentation

https://docs.python.org/3.12/using/windows.html

Rhino 8 plugin options

https://docs.mcneel.com/rhino/8/help/en-us/options/plug-ins.htm

NVIDIA driver downloads

https://www.nvidia.com/en-us/drivers/