Relay for Unreal user guide

Relay for Unreal imports Blender `.blend` files into the Content Browser. Drop a `.blend` in, and its meshes, skeletons, animations, materials, textures and cameras come in as Unreal assets, with the right scale, pivots and names. Save the `.blend` in Blender later, and Unreal re-imports it by itself, updating the same assets instead of making copies.

Requirements

	Tested	Notes
Unreal Engine	5.5 and 5.8	Other versions untested so far
Blender	4.2 LTS and 5.2 LTS	Use 4.2 or newer; a newer Blender is needed for files saved by one
OS	Windows 11 (64-bit)	The plugin is Windows only

Blender must be installed on the same computer. Blender is free and is not included with the plugin: download it from blender.org. Relay for Unreal runs it in the background, without opening a window, to read the `.blend`. You don't need to open Blender yourself or install anything in it.

The plugin turns on Unreal's **Python Editor Script Plugin** and **Editor Scripting Utilities**, which ship with the engine.

Install

1. Install the plugin from Fab into your engine version (Fab > Library > Install to Engine), then open your project and enable **Relay for Unreal** under **Edit > Plugins > Importers**. Restart the editor when asked.
2. Relay looks for Blender by itself: first the program Windows opens `.blend` files with, then `C:\Program Files\Blender Foundation`. If Blender is somewhere else, or you have several versions, set **Blender Executable** under **Edit > Editor Preferences > Plugins > Relay for Unreal**.

Importing

Any of these imports a `.blend`:

- Drag the `.blend` from Explorer into a Content Browser folder.
- **Import** in the Content Browser, then pick the `.blend`.
- **Tools > Relay > Import .blend File...** imports into the **Default Import Folder** from the settings (`/Game/Blender` unless you change it).

A progress bar shows while Blender reads the file; **Cancel** stops it. A notification says how many assets were created and updated, and any warnings go to the **Output Log** under `LogRelay`.

Assets go into the folder you imported into, sorted into `Meshes`, `Materials`, `Textures`, `Animations` and `Cinematics`. Next to them is a **Blender Source** asset (orange) named after the file. It remembers where the `.blend` lives and what it produced, and it is how re-import and auto-sync find their way back.

What gets imported

Everything visible in the `.blend`'s view layer. To import part of a file, put those objects in a collection and type the collection's name into the source asset's **Collection** field, then re-import.

In Blender	In Unreal
Mesh, curve, text or surface object	Static mesh `SM_<name>`, pivot at the object origin
Linked duplicates (same mesh, same scale)	One static mesh
Armature and the meshes it deforms	Skeletal mesh `SK_<name>` with skeleton and physics asset
Actions on that armature	Animation sequences `A_<armature>_<action>`
Principled BSDF material	Material instance `MI_<name>`
Image textures used by those materials	Textures `T_<name>` with the right color space and compression
Cameras	Cine Cameras in a Level Sequence `LS_<file name>`

Object scale is baked into the mesh. To also place actors in the open level at their Blender positions, turn on **Place in Level** in the settings.

Naming conventions for extras

These are the same conventions Unreal's FBX importer uses. Make the helpers **children** of the object they belong to in Blender.

Child object	Becomes
Mesh named `UCX_anything` (also `UBX_`, `USP_`, `UCP_`)	Custom collision: convex, box, sphere, capsule
Empty named `SOCKET_Name`	Static mesh socket called `Name`
Mesh named `anything_LOD1`, `_LOD2` ...	Extra LODs, in number order

Re-import and auto-sync

- **Auto-sync** (on by default): when the `.blend` is saved, Unreal re-imports it about a second and a half later. It waits while you are playing in the editor, and it only re-imports files that actually changed. Turn it off for one file with the source asset's **Auto Sync** box, or for everything in the settings.
- **Re-import by hand**: right-click the Blender Source asset and choose **Reimport**.

Re-importing updates the same assets in place, so materials you assigned to them, Blueprints that use them and actors placed in levels keep working.

Materials

Each material is rebuilt as an instance of one of two parent materials in `<folder>/_Relay`: `M_Relay_Opaque` and `M_Relay_Masked`. Relay reads the Principled BSDF that feeds the material output.

Principled input	Transferred as
Base Color	color or image texture
Metallic, Roughness	value, image texture, or one channel of a packed texture through **Separate Color**
Normal	image texture through a **Normal Map** node, with the green channel flipped for Unreal
Emission Color, Emission Strength	color or image texture, times strength
Alpha	value, texture, or the base color texture's alpha. Any alpha makes the material masked

Procedural textures, math nodes between the texture and the BSDF, bump nodes and UDIM tiles are not transferred; Relay warns when it falls back to a constant. Bake complex shaders to image textures in Blender first. Images packed into the `.blend` are supported.

Characters and animation

- Every armature comes in with all the meshes it deforms. Skinned meshes should be parented to their armature.
- **Animations**: *All* imports every action that animates the rig's bones, *Active* only the current one, *None* the mesh only.
- **Deform Bones Only** leaves control bones out of the Unreal skeleton.
- **Target Skeleton**: an existing skeleton asset, such as the UE5 mannequin's, to import onto. Bone names must match.
- Unreal skeletons have exactly one root bone. Parent your top-level deform bones to a single root.

Cameras and Sequencer

Cameras go into one Level Sequence, `<folder>/Cinematics/LS_<file name>`, covering the scene's frame range at its frame rate. Position, rotation and focal length are keyed on every frame, and so are the focus distance and f-stop when depth of field is on. Timeline markers bound to cameras become camera cuts. Lens shift and orthographic cameras are not transferred.

Settings

Edit > Editor Preferences > Plugins > Relay for Unreal, or **Tools > Relay > Relay Settings**. They are saved for you in this project, so each person on a team can point at their own Blender.

Setting	What it does
Blender Executable	`blender.exe` to use. Empty means find it automatically
Blender Timeout	Stop Blender if one import takes longer than this
Default Import Folder	Folder used by **Tools > Relay > Import .blend File...**
Folders	*By Type*, *By Collection* (mirrors your collections) or *Flat*
Naming Prefixes	Adds `SM_`, `SK_`, `A_`, `MI_`, `T_` (Epic's convention)
Materials and Textures	Turn off to import geometry only
Cameras to Sequencer	Imports cameras into a Level Sequence
Place in Level	Also places actors in the open level at their Blender positions
Collision	*Custom Only*, or *Custom or Box* (adds a simple box when there is no `UCX_`)
Generate Lightmap UVs, Nanite	Unreal's options for the static meshes
Animations, Deform Bones Only, Target Skeleton	See Characters and animation
Auto Sync, Auto Sync Delay	Re-import when the `.blend` is saved, and how long to wait after the save

Python and Blueprints

The import is available to editor scripts:

```
import unreal
source, summary = unreal.RelayImporterLibrary.import_blend_file(
    r"D:/Art/props.blend", "/Game/Props", '{"place_in_level": true}')
print(summary)                      # e.g. "23 new, 0 updated in 3.1s"
summary = unreal.RelayImporterLibrary.reimport_blend_source(source)   # None if it failed
```

The third argument overrides settings for this import only. A regular `AssetImportTask` with a `.blend` file name works too.

Troubleshooting

"Blender was not found" Install Blender from blender.org, or set **Blender Executable** in the settings.

"Nothing to import" The `.blend` has no visible meshes, armatures or cameras, or the **Collection** on the source asset doesn't exist in the file. Hidden objects and objects excluded from the view layer are skipped.

"... could not open this .blend" Blender can't open files saved by a newer version of Blender. Set **Blender Executable** to the same or a newer Blender than the one the file was saved with.

A material is grey The material isn't built on a Principled BSDF, or its texture file is missing. The warnings in the Output Log (`LogRelay`) name the material.

Auto-sync didn't run Check **Auto Sync** in the settings and on the source asset. Auto-sync waits while Play in Editor is running. It watches the file at the path shown on the source asset, so if you moved the `.blend`, re-import it from its new location.

Support

Email **support@szmc.co.za** with your Unreal and Blender versions, what you imported, and what went wrong. Attach the newest folder from `<YourProject>/Saved/Relay/` and the `LogRelay` lines from the Output Log.

Known limitations (1.0)

- Windows only.
- Blender must be installed on the same computer.
- No UDIMs, procedural materials or Blender Bump nodes.
- Unreal to Blender (the reverse direction) isn't supported.

Licences

The plugin is sold under the Fab Standard License. The scripts it runs inside Blender (`Source/ThirdParty/RelayBlender`) are licensed under the GNU GPL v3 or later, as Blender requires for code that uses its Python API, and their full source is included. See `ThirdPartyNotices.txt`.

Changelog

1.0.0: First release.