Relay: Blender to Unreal user guide

Relay sends what you select in Blender straight into the Unreal project you have open. Static meshes, skeletal meshes, animations, materials, textures and cameras arrive with the right scale, pivots and names, and in the right place in your level. Sending again updates the same assets and actors instead of making copies.

Requirements

	Tested	Notes
Blender	4.2 LTS, 4.5 LTS, 5.2 LTS	Needs 4.2 or newer; older Blender won't install it
Unreal Engine	5.5 and 5.8 (Interchange and legacy FBX importers)	Other versions untested so far
OS	Windows 11	macOS and Linux untested so far

Blender and Unreal must run on the same computer. No Unreal plugin is needed; Relay uses the Python scripting that ships with Unreal.

Install

1. In Blender, open **Edit > Preferences > Get Extensions**, click the **v** menu at the top right, choose **Install from Disk** and pick `relay_unreal-1.1.0.zip`.
2. Press **N** in the 3D viewport and open the **Relay** tab.

Set up Unreal (once per project)

1. **Edit > Plugins**: make sure **Python Editor Script Plugin** is enabled. Restart if Unreal asks.
2. **Edit > Project Settings > Plugins > Python**: tick **Enable Remote Execution**.

Or paste these lines into `<YourProject>/Config/DefaultEngine.ini` (the **Copy Unreal Settings** button in Relay's preferences copies them):

```
[/Script/PythonScriptPlugin.PythonScriptPluginSettings]
bRemoteExecution=True
```

Back in Blender, press **N** in the 3D viewport and open the **Relay** tab. At the top of the **Relay to Unreal** panel, click the button with two circular arrows next to the editor list (its tooltip says *Check Connection*). You should see `Connected: <YourProject> | UE 5.x`.

Sending

- **Send Selected** sends the selected objects.
- **Send Active Collection** sends everything in the collection highlighted in the Outliner.
- **Object > Send to Unreal** does the same as Send Selected.

Relay checks the selection first. Errors stop the send and show up under **Check**. Warnings are listed but don't block it. Blender stays usable while Unreal imports.

Save your .blend before your first send. Relay remembers which actors in the level belong to which Blender objects by file name.

What gets sent

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, one actor per duplicate
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 Camera actors in a Level Sequence `LS_<file name>`

Object scale is baked into the mesh. Location and rotation go onto the placed actor.

Naming conventions for extras

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

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

`UCX_` collision must be convex. Relay warns you if it isn't.

Materials

Relay rebuilds each material as an instance of one of two parent materials it creates in `<Content Folder>/_Relay`: `M_Relay_Opaque` and `M_Relay_Masked`. It reads the Principled BSDF feeding 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 tells you when it falls back to a constant value. For complex shaders, bake them to image textures first.

Materials whose names contain spaces or dots (`Old Wood.001`) are matched to their mesh slots even after Unreal renames them.

Characters and animation

- Select the armature, or any mesh it deforms. Relay sends the rig with every mesh it deforms.
- **Animations**: *Active Action* sends the current action. *All Matching* sends every action that animates the rig's bones. *None* sends the mesh only.
- **Deform Bones Only** leaves control bones out of the Unreal skeleton.
- **Target Skeleton**: set this to an existing skeleton asset, such as the UE5 mannequin's, to import onto it. Bone names must match.

Relay exports a temporary centimeter-scale copy of your rig, so the skeleton arrives with no x100 scale on the root bone and no extra "Armature" root bone. Your own objects, meshes and actions are never changed.

Skinned meshes should be parented to their armature. Shape keys are kept unless the mesh also has non-armature modifiers such as Bevel or Subdivision. Relay warns you in that case.

Cameras and Sequencer

Cameras in the selection go into one Level Sequence, `<Content Folder>/Cinematics/LS_<file name>`, covering the scene's frame range at the scene's frame rate.

- Each camera becomes a **Cine Camera** placed in the level. Its movement, rotation and focal length are keyed on every frame, so what you see through the Blender camera is what Sequencer plays back.
- With **depth of field** on, the focus distance is keyed too, including a focus object followed frame by frame, and the f-stop is copied.
- **Timeline markers bound to cameras** (Ctrl+B in the timeline) become camera cuts, the same way Blender switches cameras. Without markers, the scene camera is used for the whole range.
- The sensor size follows your render aspect ratio and Blender's sensor fit.
- Sending again updates the same sequence and cameras. Tracks and cameras you added yourself in Sequencer are left alone.

Not transferred: lens shift, and orthographic cameras (Unreal's Cine Camera is perspective only). Relay warns about both.

Settings

Setting	What it does
Content Folder	Where assets go in the Unreal project. Must start with `/Game`
Folders	*By Type* (Meshes, Materials, Textures, Animations), *By Collection* (mirrors your collections) or *Flat*
Naming Prefixes	Adds `SM_`, `SK_`, `A_`, `MI_`, `T_` (Epic's convention)
Materials and Textures	Turn off to send geometry only
Place in Level	Adds actors at the Blender positions. Sending again moves the same actors
Outliner Folder	Folder for placed actors in Unreal's Outliner
Collision	*Custom Only*, or *Custom or Box* (adds a simple box when there is no `UCX_`)
Generate Lightmap UVs	Unreal's lightmap UV generation for static meshes
Nanite	Builds static meshes with Nanite
Cameras to Sequencer	Sends cameras into a Level Sequence (see Cameras and Sequencer)
Exchange Folder	Where FBX files are written for Unreal to read. The last three sends are kept

Network settings are in **Preferences > Add-ons > Relay** and match Unreal's defaults. If port 6776 is busy, for example because Send to Unreal is also open, Relay picks the next free port by itself.

Troubleshooting

"No Unreal Editor found" Unreal must be open with a project loaded, and **Enable Remote Execution** must be ticked (see Set up Unreal). If you changed Unreal's *Multicast Group Endpoint* or *Multicast Bind Address*, make Relay's preferences match.

Half the object is under the floor Relay places objects exactly where they are in Blender, around their origin. Blender's default cube is centered on the origin, so half of it sits below the grid in Blender too, and below the floor in Unreal. Move the object up in Blender, or set its origin to the bottom (**Object > Set Origin**), and send again.

Several editors are open Pick the one you want in the editor list at the top of the Relay tab.

The send finished but a material is grey The material isn't built on a Principled BSDF, or its texture file is missing. Run **Check** to see which.

Animations are missing Set **Animations** to *All Matching*, or make the action active on the armature.

"Unreal needs one root bone" Unreal skeletons have exactly one root. Parent your top-level deform bones to a single root bone.

Where are the exported files? Advanced > Open Exchange Folder. Attach the newest `send_*` folder when you ask for support.

Support

Email **support@szmc.co.za** with your Blender and Unreal versions, what you sent, and what went wrong. A zip of the newest `send_*` folder from the exchange folder helps a lot.

Known limitations (1.1)

- Blender and Unreal must be on the same machine.
- No UDIMs, procedural materials or Blender Bump nodes.
- Unreal to Blender (the reverse direction) isn't supported.
- Sending doesn't save your Unreal level. Imported assets are saved.

Changelog

1.1.0: Cameras into Unreal's Sequencer: Cine Cameras with movement, focal length and focus keyed per frame, and marker-based camera cuts. Fixes Unreal 5.5 skeletal import.

1.0.0: First release.