

# FPS Arms Modular Pack

Unreal Engine documentation

---

## Overview

The FPS Arms Modular Pack is a set of first-person arms built to be mixed. It includes eight skeletal meshes:

- **Three gloves:** standard, fingerless and semi-fingerless, each in 3 variants.
- **Two sleeve pieces:** a long sleeve and a tactical jacket, each in 3 variants.
- **Five bare skin pieces:** the full arms, the arms, the hands, a full set of fingers and a three-finger set, each in 2 skin tones.

**Every variant comes in a clean and a dirty version.** Each piece is a separate **skinned mesh** on the UE5 Mannequin skeleton. Assemble a loadout from the pieces and it animates with anything your Mannequin already runs, no retargeting needed.

The pack also ships the full source: the Blender rig, the smart materials, the high poly meshes and every baked map. You can build your own variants on top of ours.

## What's included

### Skeletal meshes

| Piece                 | Asset                  | Texture variants        | Triangles |
|-----------------------|------------------------|-------------------------|-----------|
| Bare arms             | SK_Arms                | 2 skin tones + 2(dirty) | 2841      |
| Full Arms             | SK_FullArms            | 2 skin tones + 2(dirty) | 9121      |
| Fingers               | SK_Fingers             | 2 skin tones + 2(dirty) | 3824      |
| Hands                 | SK_Hands               | 2 skin tones + 2(dirty) | 6560      |
| Three fingers         | SK_ThreeFingers        | 2 skin tones + 2(dirty) | 2132      |
| Standard glove        | SK_GloveStandard       | 3 + 3(dirty)            | 8682      |
| Fingerless glove      | SK_GloveFingerless     | 3 + 3(dirty)            | 8779      |
| Semi-fingerless glove | SK_GloveSemifingerless | 3 + 3(dirty)            | 6808      |
| Long sleeve           | SK_LongSleeve          | 3 + 3(dirty)            | 1939      |
| Tactical jacket       | SK_TacticalJacket      | 3 + 3(dirty)            | 3462      |

Every texture variant comes in a clean and a dirty version. **All textures are 4096 × 4096.**

The meshes have no LODs.

## Materials

Two master materials for cloth and gloves, M\_Simple and M\_Blending, same apply for skin. Every texture variant has its own material instances.

## Source files

The Blender rig file, the smart materials, the low and high poly FBX files, the baked maps, and the extra materials and alphas used for texturing. See the Source files section. Requirements

**Unreal Engine:** 5.3 or later.

**Skeleton:** the UE5 SK\_Mannequin skeleton, shared by Manny and Quinn.

**Source files:** Blender 4.5\* and smart materials made in Adobe Substance 3D Painter 9.1.2.

\*For weight painting adjustments use the “**Rigging\_scene**” file included in the source files

For rigging and skinning compatible with Unreal Skeleton we relied on [Game Rig Tool](#) we suggest referring to the original tutorial guide found at [Blender to Game Engines \(Unreal, Unity, Godot etc.\) - YouTube](#)

## Skeleton and animation

All pieces are skinned to the full-body SK\_Mannequin skeleton. Any animation, Animation Blueprint or asset built for the SK\_Mannequin skeleton works on them directly, with no retargeting.

Tested with the First and Third Person Template.

# Building a loadout

## Which pieces go together

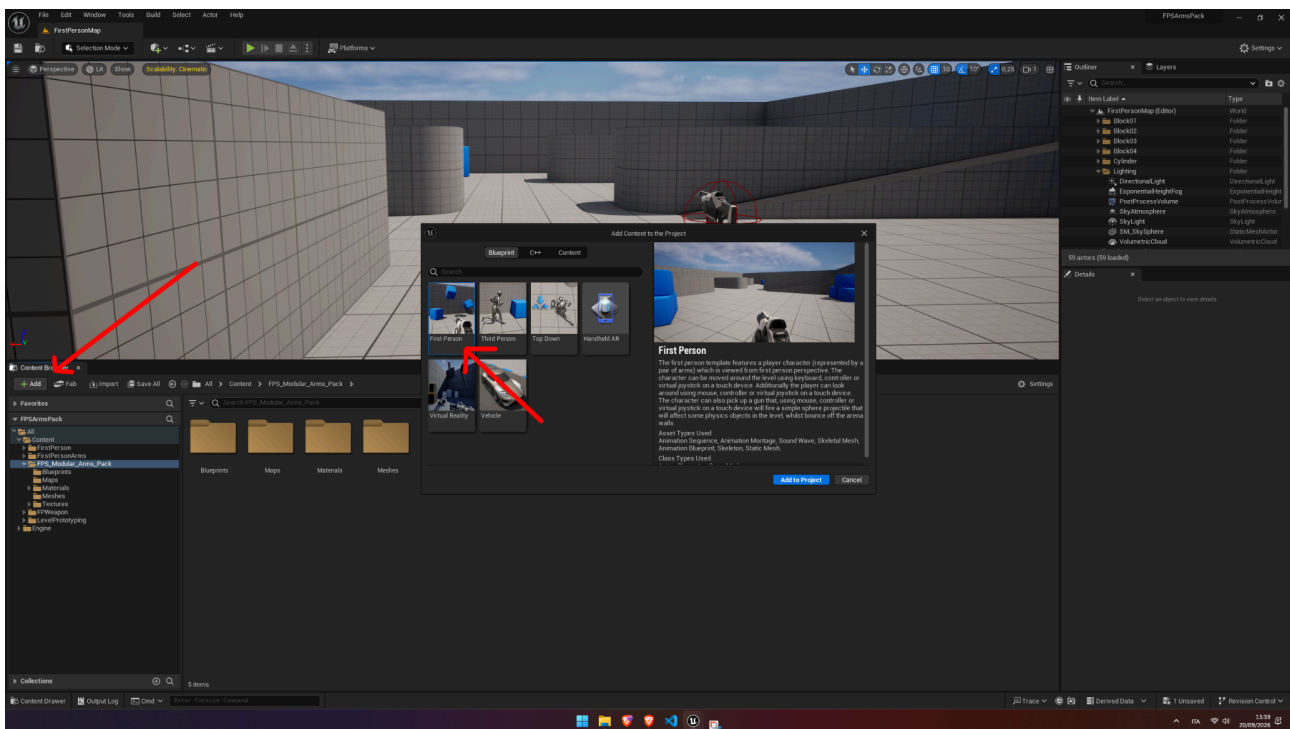
Each glove has a matching fingers mesh:

| Glove           | Add                                      |
|-----------------|------------------------------------------|
| Standard        | Nothing, the glove covers the whole hand |
| Fingerless      | SK_Fingers                               |
| Semi-fingerless | SK_ThreeFingers                          |

## Setting it up (5.3 to 5.5)

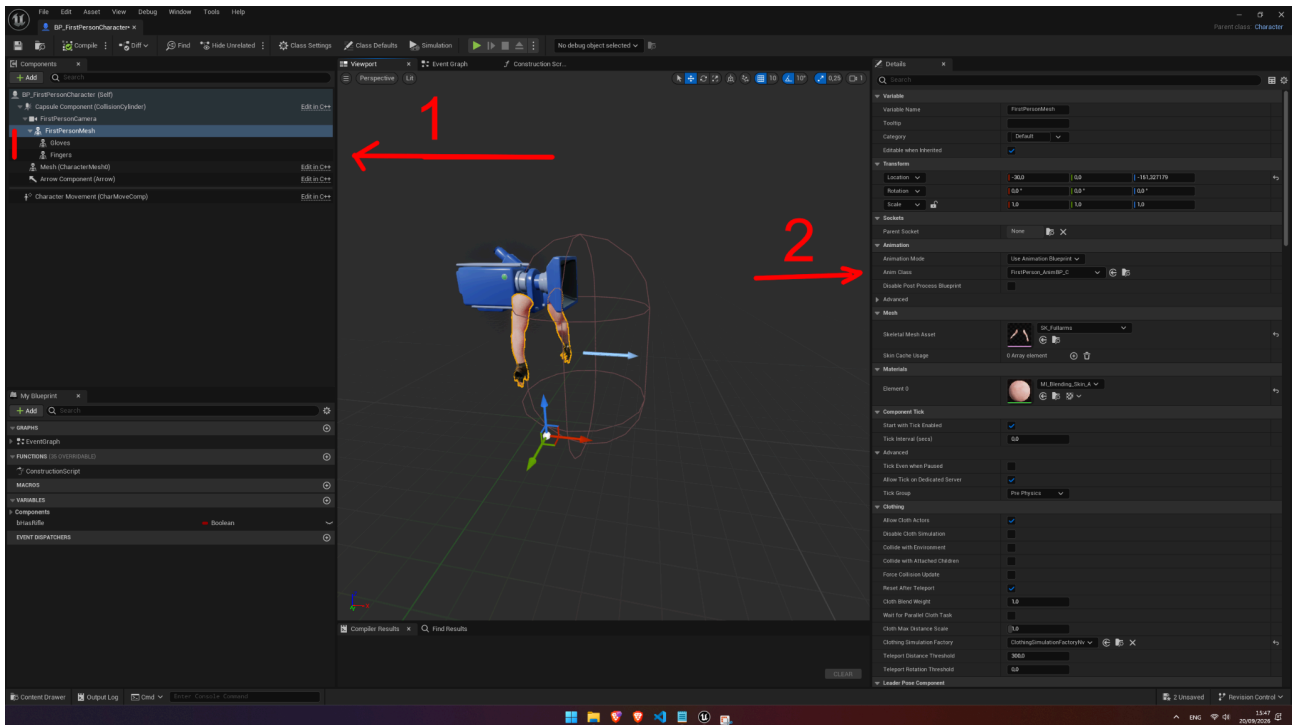
This example builds a loadout on Epic's First Person content, using a glove and its matching fingers mesh.

1. Add the First Person content to your project. In the Content Browser, click **Add**, choose **Add Feature or Content Pack**, pick **First Person** and click **Add to Project**.

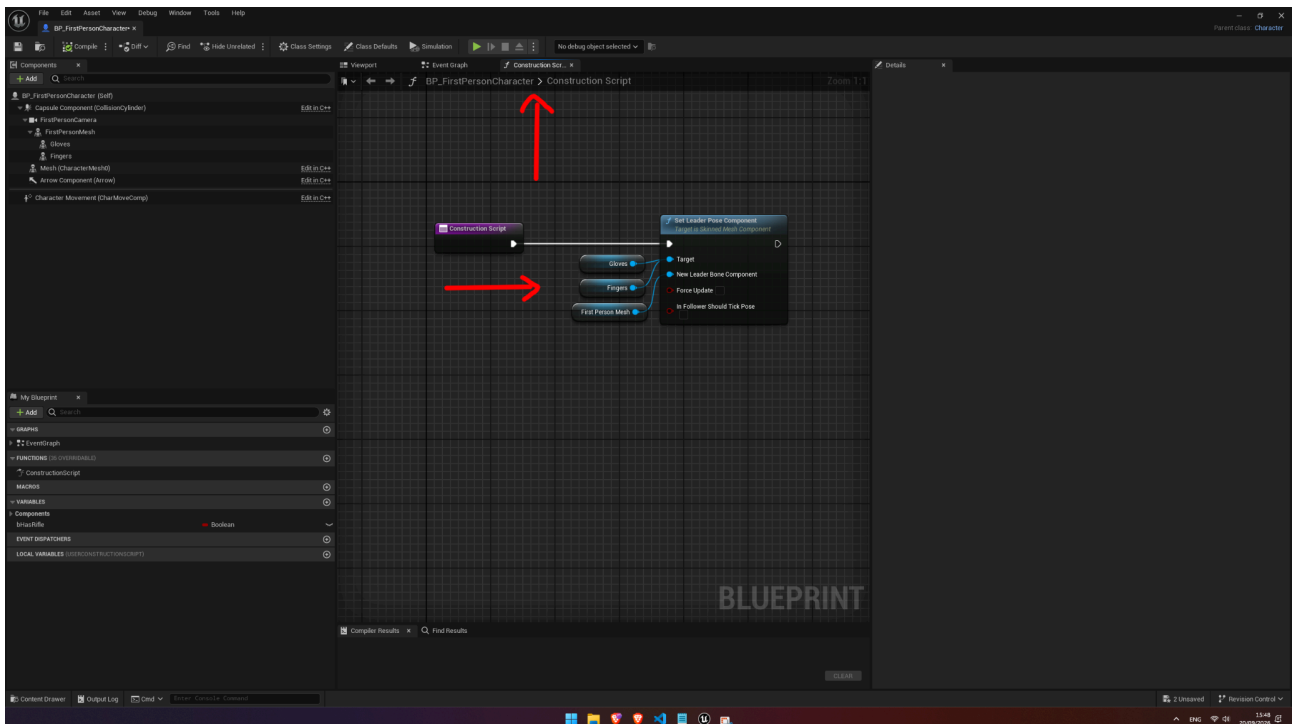


2. Open **BP\_FirstPersonCharacter** and select the first person mesh in the Components panel. This component is the leader.

- Click **Add** in the Components panel and choose **Skeletal Mesh**. The new component is created as a child of the first person mesh. Rename it after the piece it holds, for example Gloves.
- In the Details panel, set **Skeletal Mesh Asset** to the piece, for example SK\_GloveFingerless.
- Repeat steps 3 and 4 for every piece in your loadout. For a fingerless glove, that's one component for the glove and one for SK\_Fingers.



- In the Construction Script, add a **Set Leader Pose Component** node. Connect the piece components to **Target** and the first person mesh to **New Leader Bone Component**.

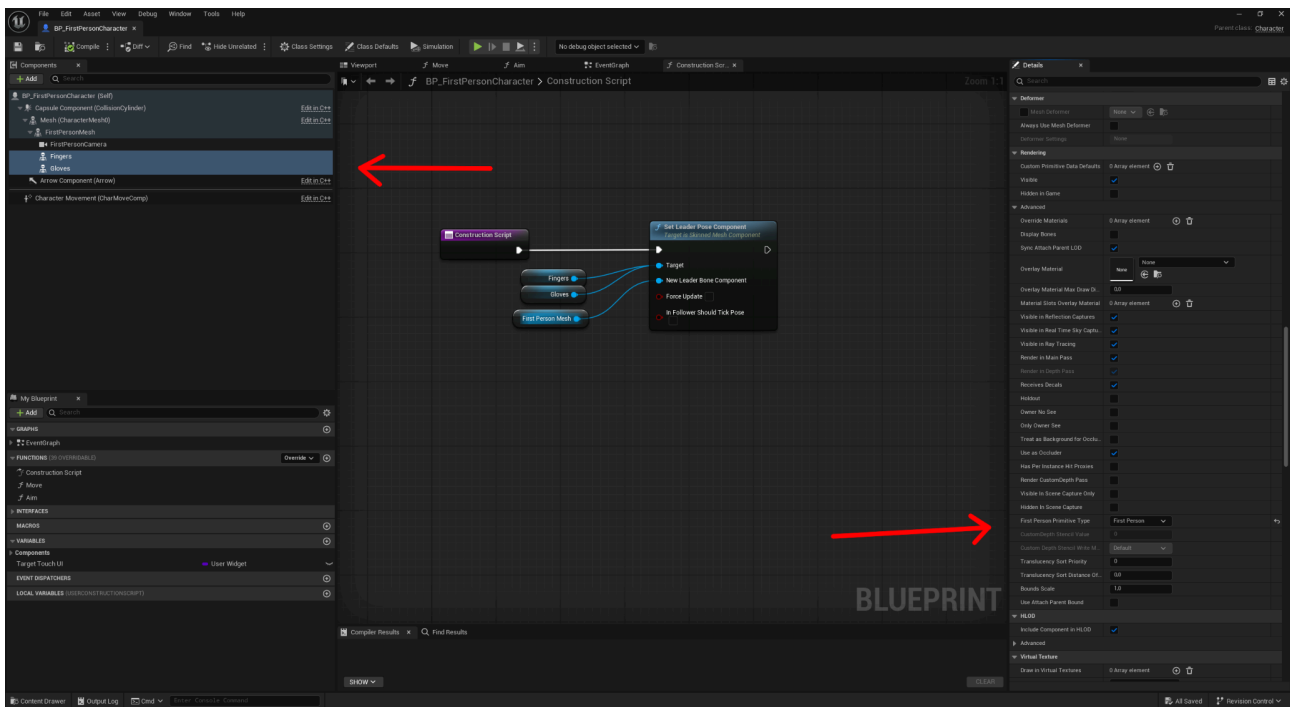


- Setup is done!

## Setting it up (5.6 & later)

This example builds a loadout on Epic's First Person content, using a glove and its matching fingers mesh.

1. Add the First Person content to your project. In the Content Browser, click **Add**, choose **Add Feature or Content Pack**, pick **First Person** and click **Add to Project**.
2. Open **BP\_FirstPersonCharacter** and select the first person mesh in the Components panel. This component is the leader.
3. Click **Add** in the Components panel and choose **Skeletal Mesh**. The new component is created as a child of the first person mesh. Rename it after the piece it holds, for example Gloves.
4. In the Details panel, set **Skeletal Mesh Asset** to the piece, for example SK\_GloveFingerless.
5. Repeat steps 3 and 4 for every piece in your loadout. For a fingerless glove, that's one component for the glove and one for SK\_Fingers.



6. Select each piece component. In the Details panel, under **Rendering > Advanced**, set **First Person Primitive Type** to **First Person**. **This step is required.** The first person camera renders the arms with its own field of view and scale, so any piece left on None is drawn differently and won't line up with the arms.
7. In the Construction Script, add a **Set Leader Pose Component** node. Connect the piece components to **Target** and the first person mesh to **New Leader Bone Component**.
8. Press Play to check the loadout.

## Setting it up (Third Person 5.3+)

For true first person, with the camera on a full-body character, use Epic's Third Person content instead. The steps are the same as above, with three differences:

- In Add Feature or Content Pack, pick **Third Person** instead of First Person.
- Open **BP\_ThirdPersonCharacter** and use its **Mesh** component as the leader.
- Skip the First Person Primitive Type step. It only applies to the First Person setup on 5.6 and later.

# Materials

## Master materials

**M\_Simple:** Default Lit, Two Sided. It samples two textures, BCO and NRM, and wires them straight to the outputs.

**M\_Blending:** Default Lit, Two Sided. It samples a clean set and a dirty set, four textures in total, and blends between them with a per-instance parameter.

Two Sided is on to cover small gaps between pieces and the small single-sided cloth pieces.

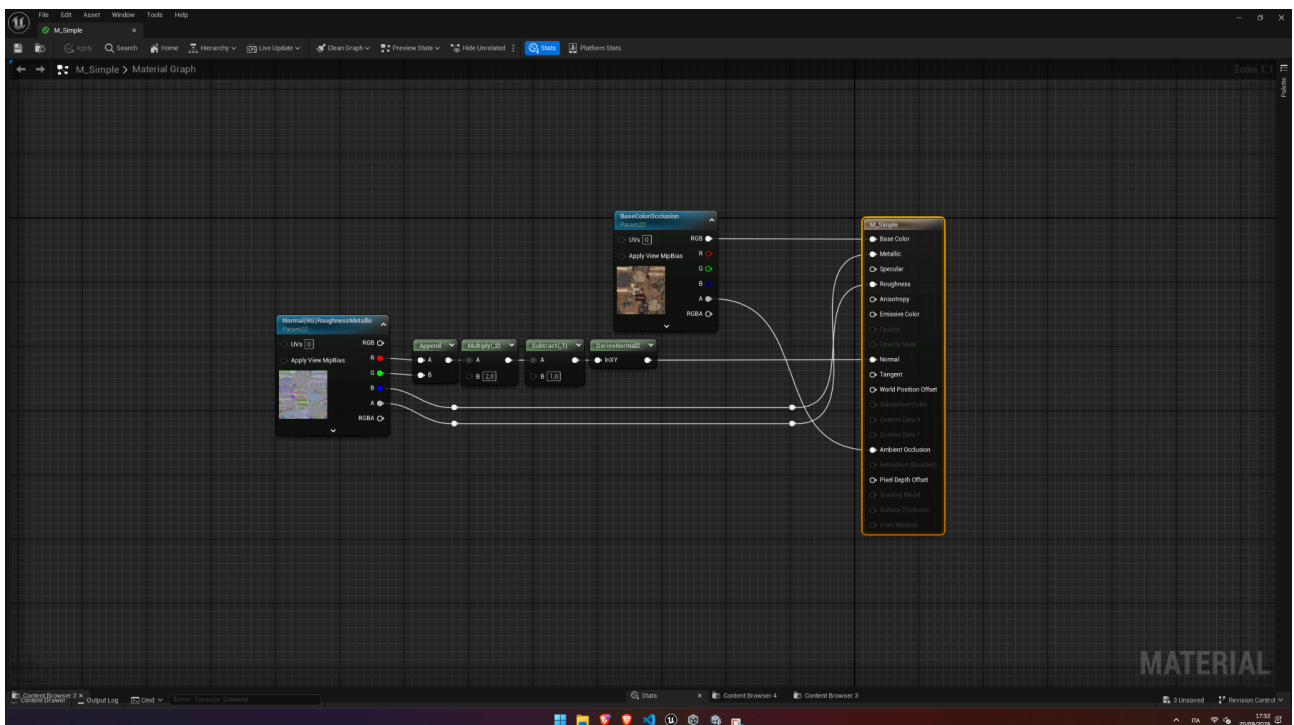
**M\_Skin\_Simple / M\_Skin\_Blending:** Preintegrated Skin shading model. They use the same texture layout, with the NRM alpha driving subsurface instead of metallic.

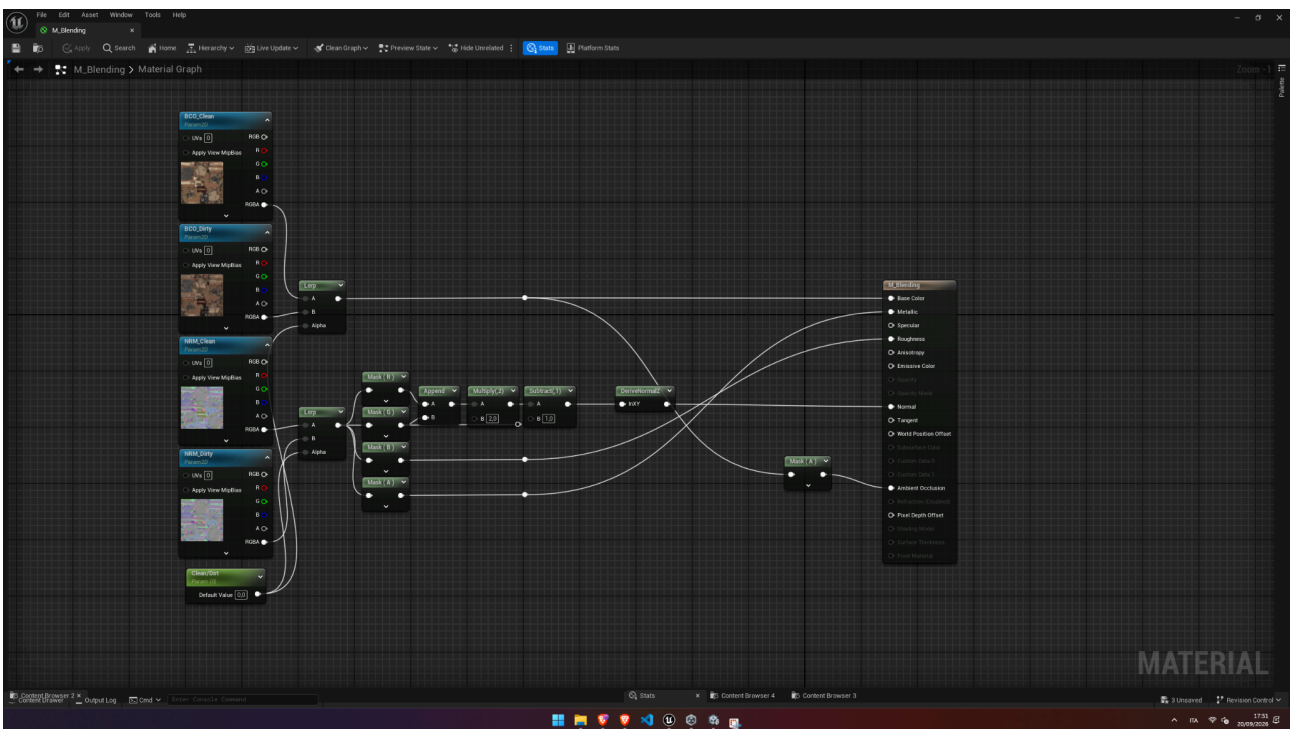
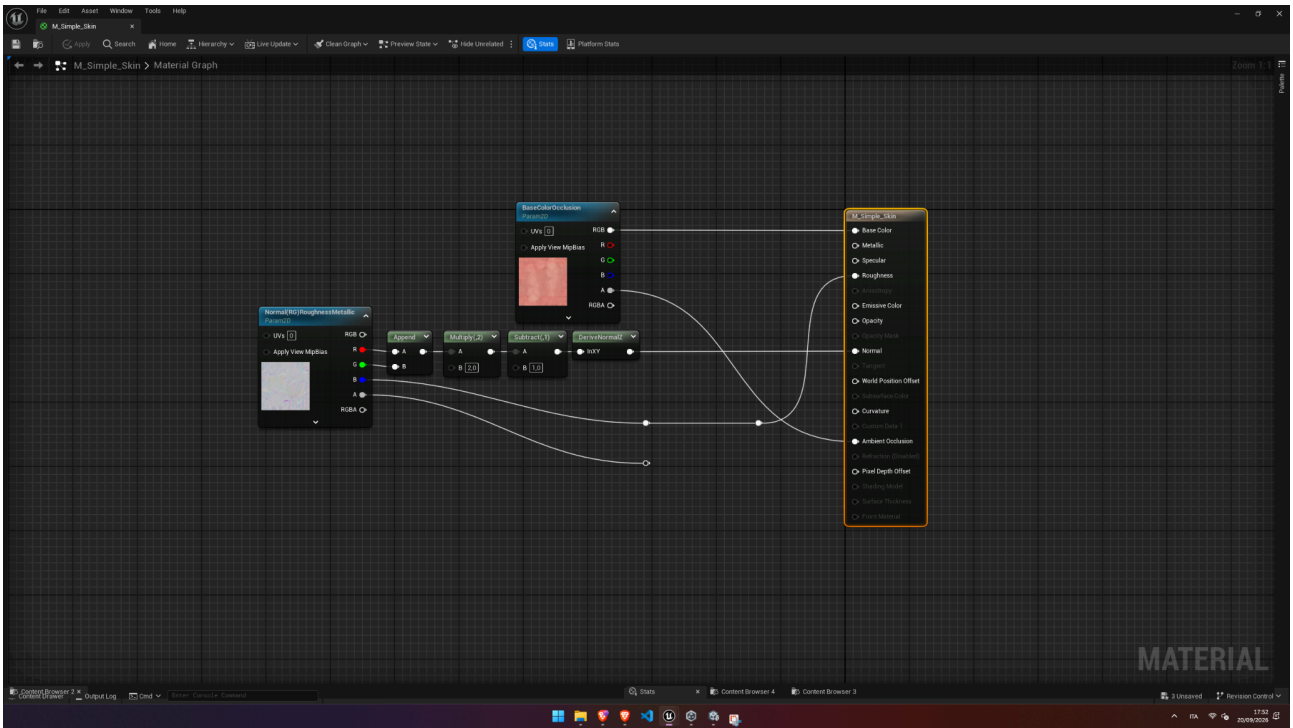
## Simple or Blending

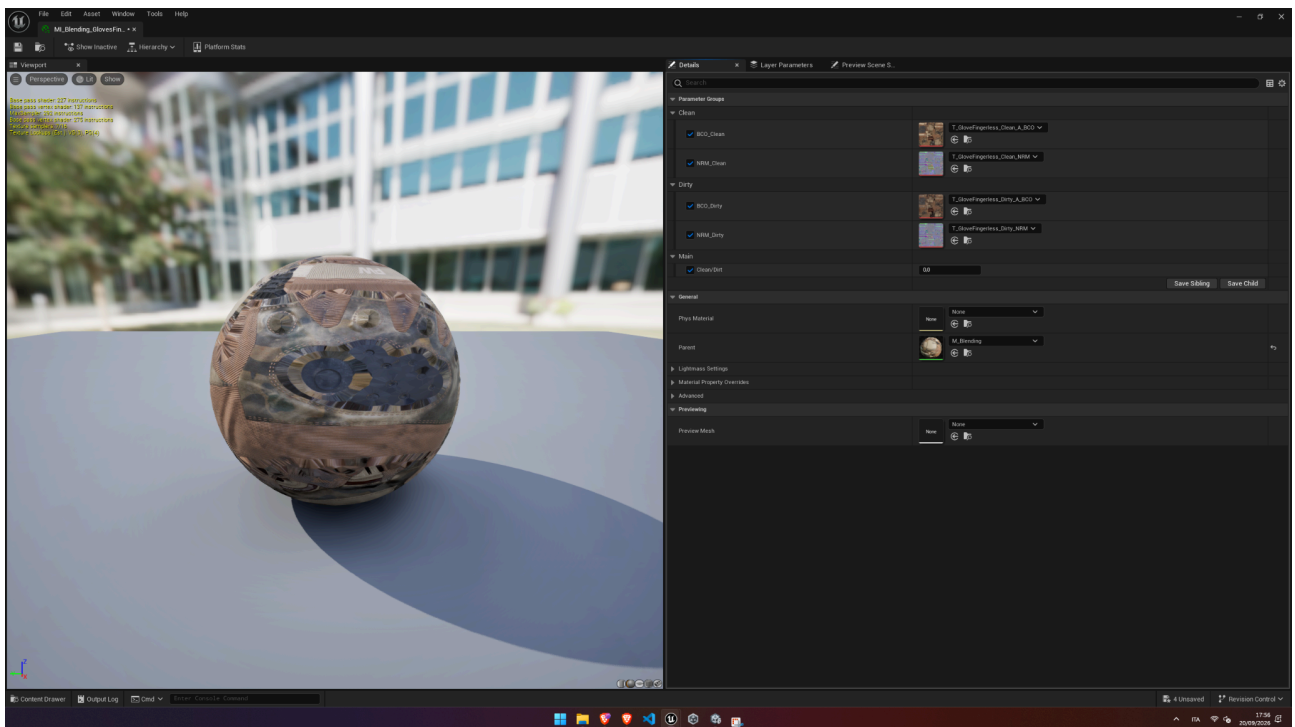
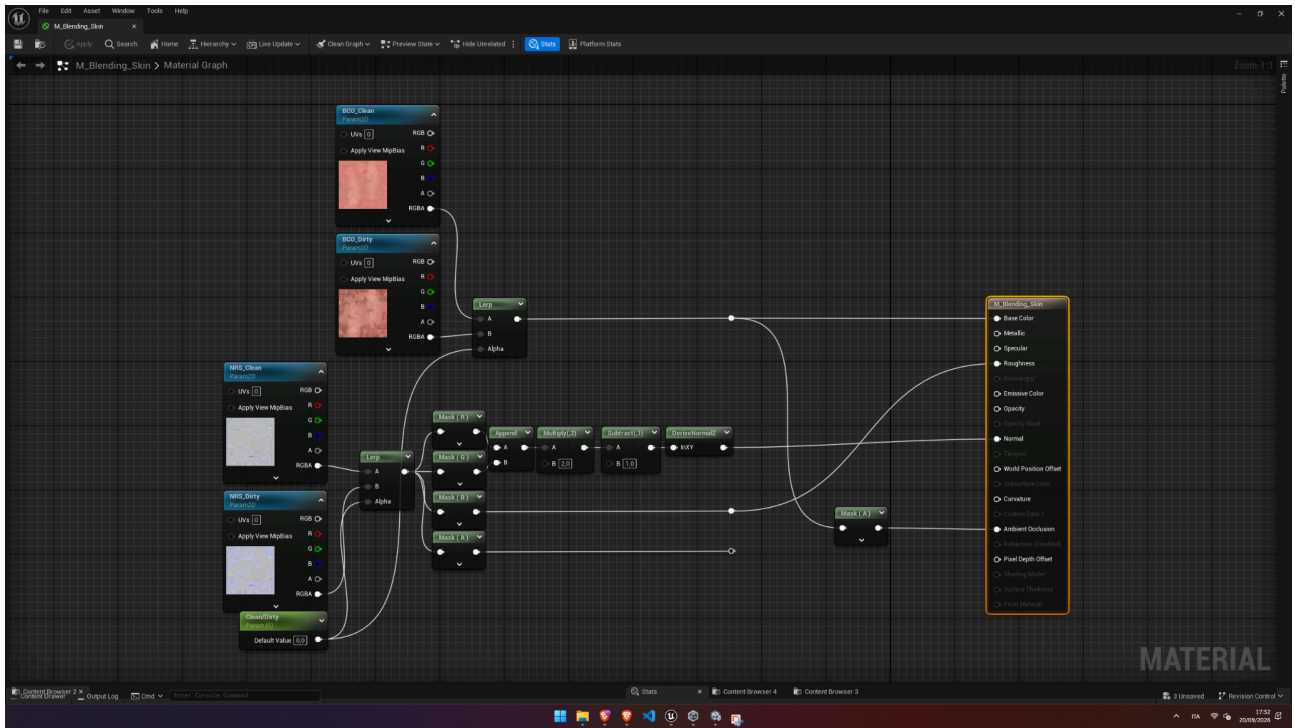
Use Simple when a piece's dirt level never changes: pick the clean or dirty instance and you're done. Use Blending when dirt changes at runtime, or when you want a specific amount between clean and dirty. Blending costs twice the texture samples, so don't use it where Simple will do.

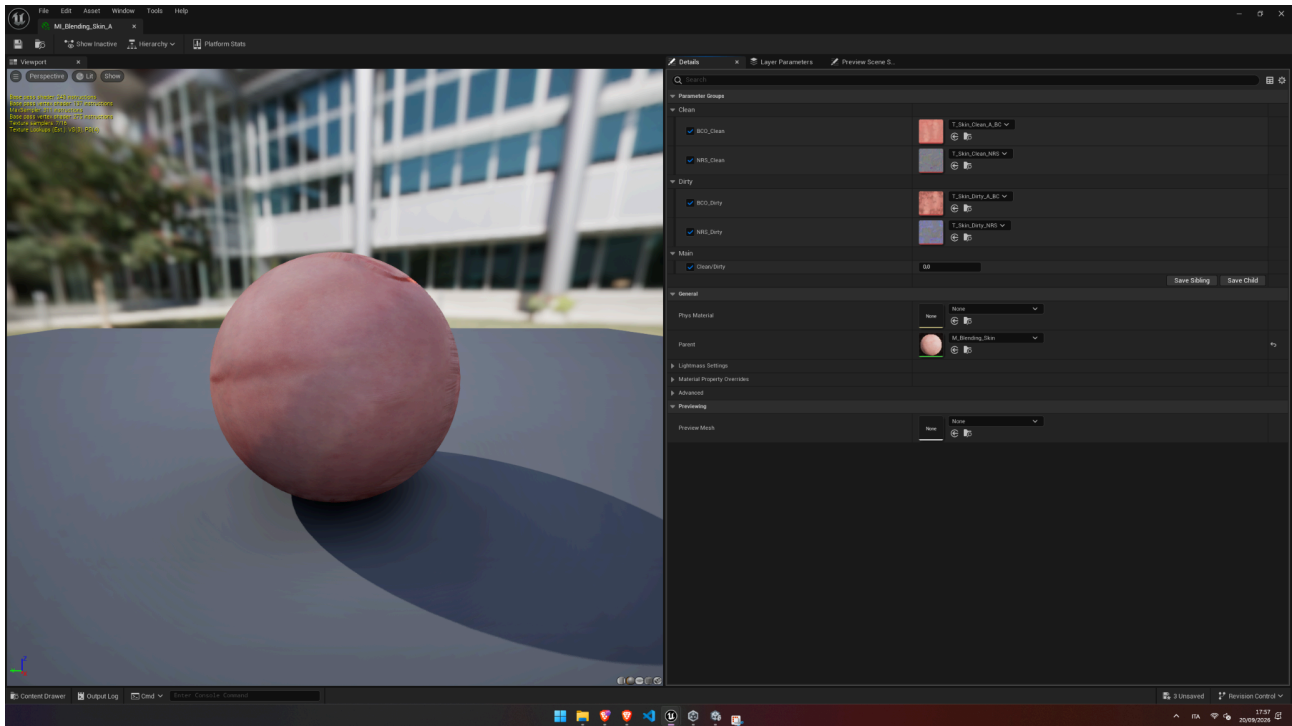
## Material instances

Each texture variant has its own material instances with its textures assigned, named MI\_[Clean, Dirty or Blending]\_[Piece]\_[Variation]









## Texture packing

Each texture set has two textures, named  $T_{[Piece]_{[Clean \text{ or } Dirty]_{[BCO \text{ or } NRM]_{[Variant]}}$ .

| Texture      | R                   | G                   | B                   | A                                    | sRGB |
|--------------|---------------------|---------------------|---------------------|--------------------------------------|------|
| <b>BCO</b>   | <b>Base Color R</b> | <b>Base Color G</b> | <b>Base Color B</b> | Ambient Occlusion                    | On   |
| <b>NRM/S</b> | <b>Normal X</b>     | <b>Normal Y</b>     | <b>Roughness</b>    | <b>Metallic (Scattering on skin)</b> | Off  |

sRGB only applies to the RGB channels, so the occlusion in the BCO alpha stays linear.

If you reimport or resize textures, keep sRGB off on every NRM. With sRGB on, the normals, roughness and metallic all come out wrong.

### Normal Z reconstruction

The NRM texture stores only the X and Y of the normal, which frees B and A for roughness and metallic. The material rebuilds Z in two steps:

1. Remap X and Y from the 0 to 1 range into -1 to 1:  $XY = NRM.rg \times 2 - 1$
2. Derive Z with the **DeriveNormalZ** node:  $Z = \text{sqrt}(\text{saturate}(1 - (X^2 + Y^2)))$ . The node outputs the full normal, which goes into Normal.

This saves one texture sample per material, two in M\_Blending, at the cost of a few extra ALU instructions. That's a good trade for first-person arms, which cover a large part of the screen.

## Exporting from Painter NRM

When exporting your custom textures from Substance Painter remember to flag the NRM textures as Raw otherwise the export data is wrong if you opt to use the current Material Graph.

## Runtime dirt control

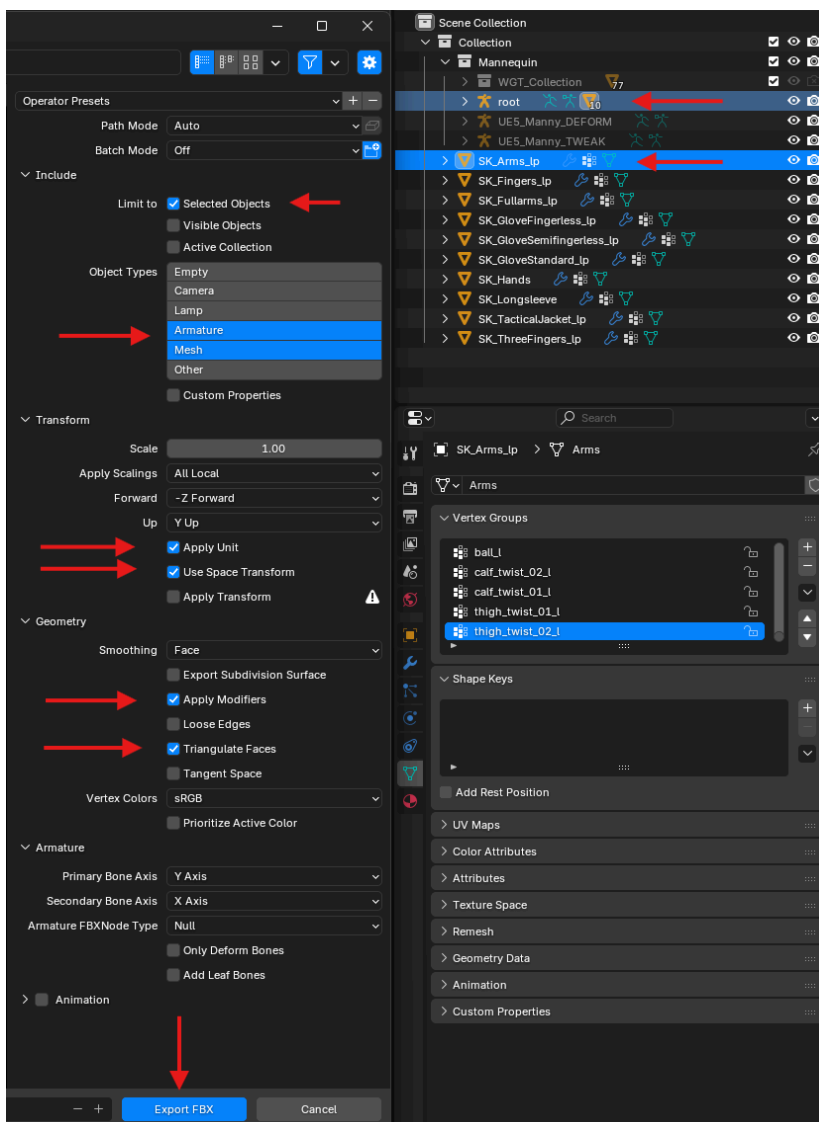
M\_Blending exposes a scalar parameter that blends **from clean (0) to dirty (1)**. The parameter is set per material instance, so each piece has its own value. This value can also be controlled from your blueprint if you set it up.

## Source files

### Blender rig file (Blender 4.5)

One file holds every piece as a separate object, so the pack stays modular. The armature matches the SK\_Mannequin skeleton and is organised in three bone collections: ROOT, DEFORM and TWEAK. All pieces are weight painted.

For correct exports make sure to follow these settings.



## Meshes

**Low poly with UVs:** FullArms\_lp, Arms\_lp, Fingers\_lp, Hands\_lp, ThreeFingers\_lp, GloveFingerless\_lp, GloveStandard\_lp, Longsleeve\_lp, TacticalJacket\_lp and GloveSemifingerless\_lp.

**Decimated high poly for baking:** Arms\_dec, FingerlessGloves\_dec, LongSleeve\_dec, StandardGloves\_dec, TacticalJacket\_dec, SemiFingerless\_dec (fingers and hands use the same as arms).

**Rigged meshes with root bone:** one FBX per piece.

## Baked maps

Every piece comes with its bakes, so you can texture without rebaking. The file suffixes are \_alpha, \_ao, \_cavity, \_convex, \_normal, \_normalobj, \_objid, \_thickness and \_position. The position map was baked in Painter with One axis set to Y.

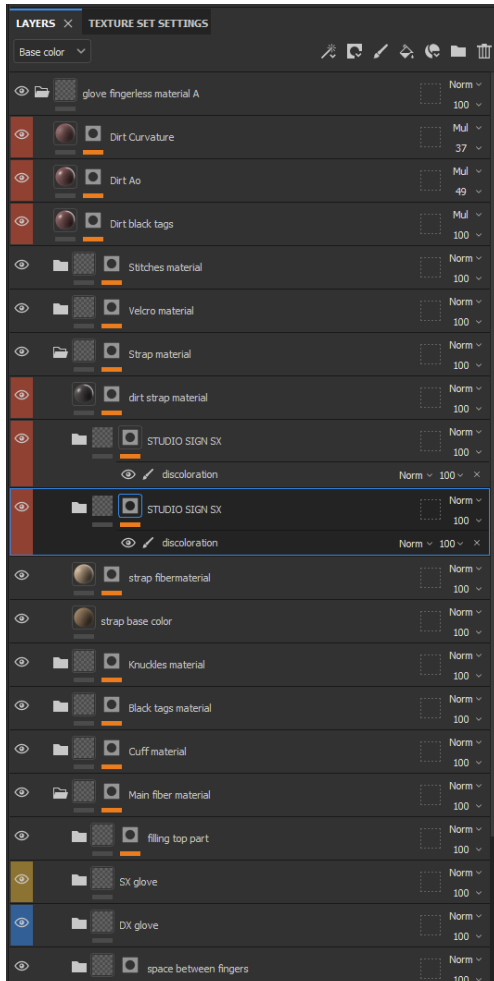
## Smart Materials (Substance 3D Painter 9.1.2)

There is one smart material preset per piece variation. Arms and fingers share one smart material, built on.

The layers are colour-coded:

- **Red:** dirt and the studio logo. Turn them off for the clean version.
- **Blue and yellow:** Warp Projection materials for the right and left sides of the mesh. They're used in areas where each side needed its own projection.

The Extra folder holds the materials and alphas we used on top of Painter's own library.



# Texturing

If you want to make adjustments to textures of a mesh in Substance 3D Painter using our included Smart Materials follow these steps:

1. **Select the correct source file** from the following folder:

**Meshes\_for\_bake > Lowpoly\_Meshes**

2. **Import all the matching baked maps:**

**Textures\_Baked > BAKED [MESH NAME]**

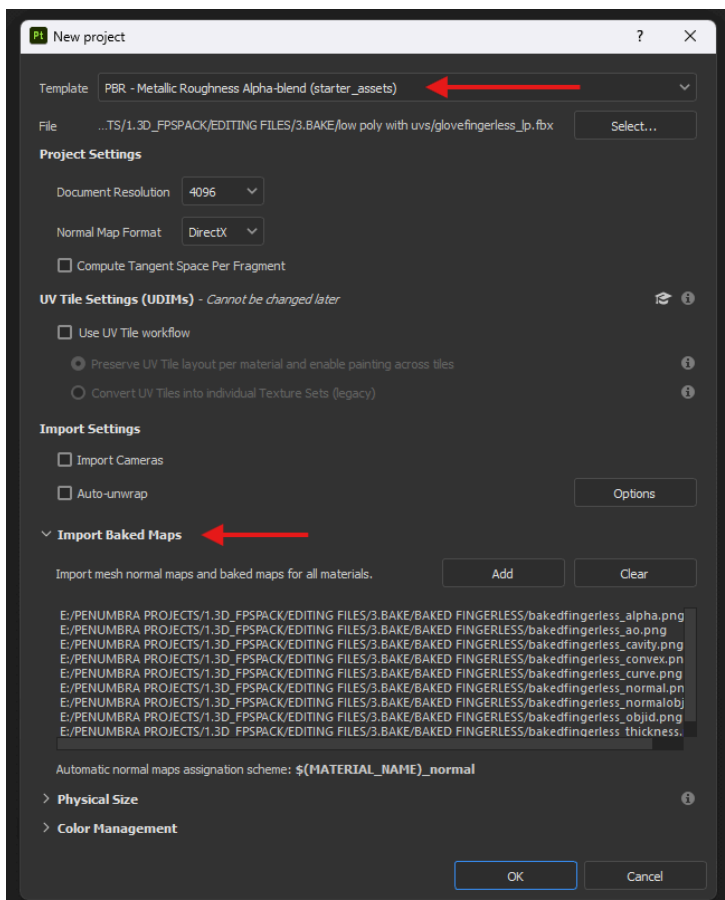
Please note that you will not find the **Position Map** in this folder, as it is always baked directly in Substance 3D Painter.

## Rebaking Textures

If you need to bake any texture, you can use the decimated meshes (as high poly) located in:

**Meshes\_for\_bake > Decimated\_Meshes**

For the Object ID Map, we recommend sticking with the original source map, as rebaking from the decimated version would produce poor results, that file cannot reproduce the same Object ID Map as the original source files. We provided a merged, decimated file rather than the full-quality high-poly due to file size constraints.



## Useful links

**Support:** [support@penumbraworks.com](mailto:support@penumbraworks.com)

**Website:** [www.penumbraworks.com](http://www.penumbraworks.com)