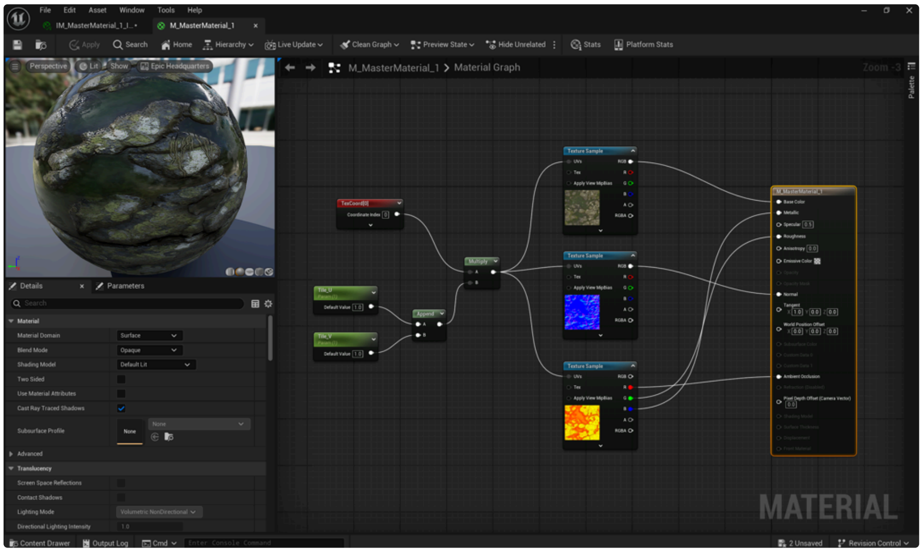
Technical Writing Example 1

Creating a PBR Material Shader in Unreal Engine 5



Material

Material Domain: Surface

Blend Mode: Opaque

Shading Model: Default Lit

Two Sided: □

Use Material Attributes: □

Cast Ray Traced Shadows: ☑

Subsurface Profile: None

Advanced

Translucency

Screen Space Reflections: □

Contact Shadows: □

Lighting Mode: Volumetric NonDirectional

Directional Lighting Intensity: 1.0

Purpose

This guide outlines the process for creating a physically based rendering (PBR) material shader in Unreal Engine 5 (UE5) for a 3D asset, ensuring realistic lighting and texture integration. The shader will include base color, metallic, roughness, and normal map inputs, suitable for assets like characters or props in an animation pipeline.

Prerequisites

- Unreal Engine 5.3 or later installed.
- 3D asset imported into UE5 with prepared texture maps (Base Color, Normal, Metallic, Roughness) supplied by training.
- Basic knowledge of UE5 interface and Material Editor.
- Access to texture files in .TGA or .PNG format.

Tools

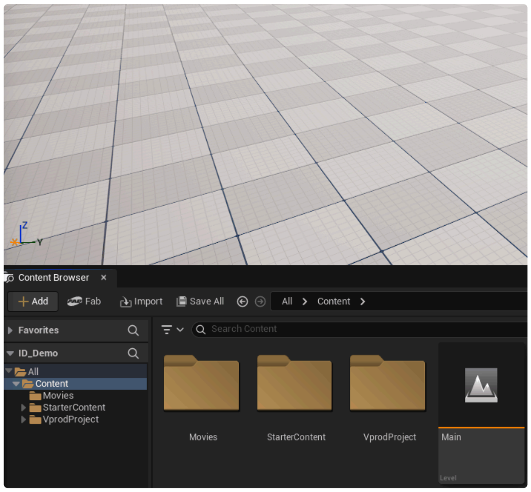
- UE5 Material Editor (**For now, don't use Substrate for materials*)
- File Explorer (for texture import)
- Texture authoring software (e.g., Substance Painter, optional for reference)

Files

- **Textures (**download files for Example)

Procedure

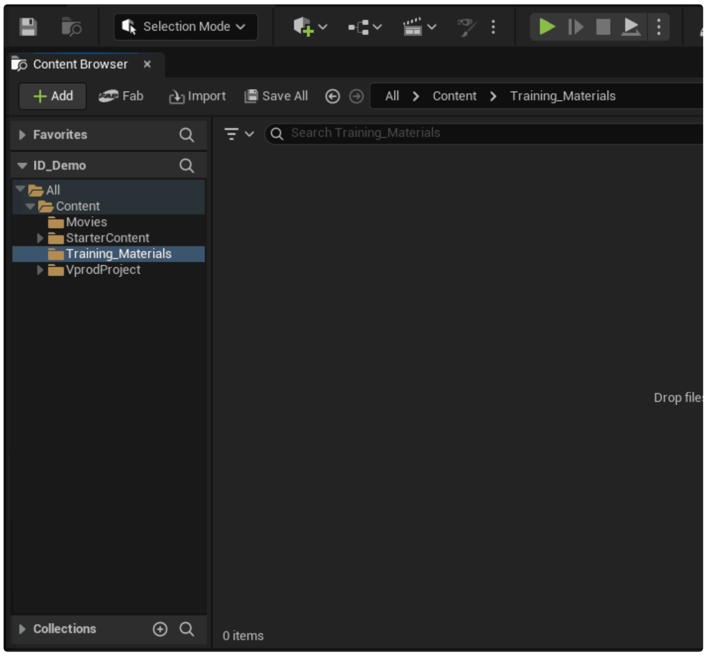
Step 0: Create a New Folder



Content Browser Tab at the bottom of the UI

1. Open your UE5 project.
2. In the **Content Browser**, create a new folder called **Training_Materials**.

Step 1: Create a New Material

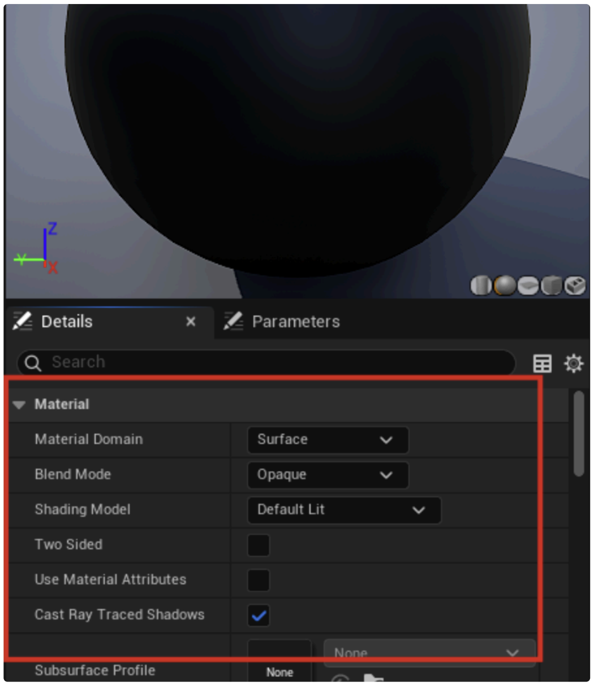


Create and Name a Material in the Folder Created

1. Open the **Training_Materials**.
2. Right-click, select **Create Basic Asset** > **Material**.
3. Name the material descriptively (e.g., `M_Character_Skin` or `M_Prop_Metal`), for this demo, we're going to call it: **M_MasterMaterial_1.**
   *” **M_**” *as a prefix designates that this is a material*

4. Double-click the material to open it in the **Material Editor**.

Step 2: Set Material Properties



Material

Material Domain: Surface

Blend Mode: Opaque

Shading Model: Default Lit

Two Sided: ☐

Use Material Attributes: ☐

Cast Ray Traced Shadows: ☑

None

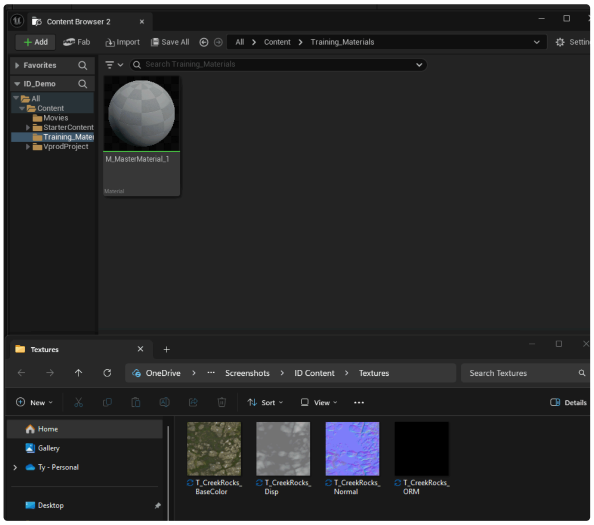
Subsurface Profile: None

Details Tab in the Material Editor

1. In the **Details** panel (left side of Material Editor), locate the **Material** section.
2. Set **Material Domain** to `Surface` (default for PBR materials).
3. Set **Blend Mode** to `Opaque` for non-transparent surfaces or `Masked` for materials with alpha (e.g., foliage). For translucent materials (e.g., glass), select `Translucent` and adjust later.
4. Ensure **Shading Model** is set to `Default Lit` for standard PBR rendering.

Step 3: Import Texture Maps

There are two ways of importing textures. You can either drag them from the folder that they reside in on your hard drive into the folder in Unreal that you created (**Training_Materials** in this case)

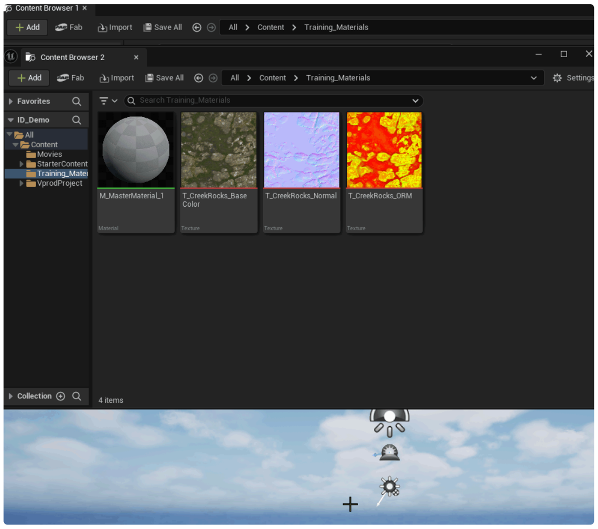


Dragging from Explorer

Import choice 1 - Drag and Drop

1. Go to the folder on your computer that holds the textures you want to import.
2. Select those textures and simply drag them into the folder you created (**Training_Materials**) in the **Content Browser**.

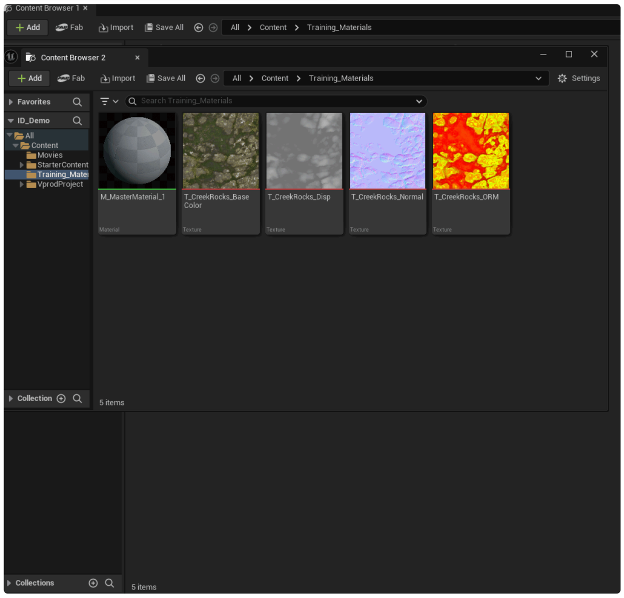
or



Import Button to Bring in Textures

Import choice 2 - Import Button

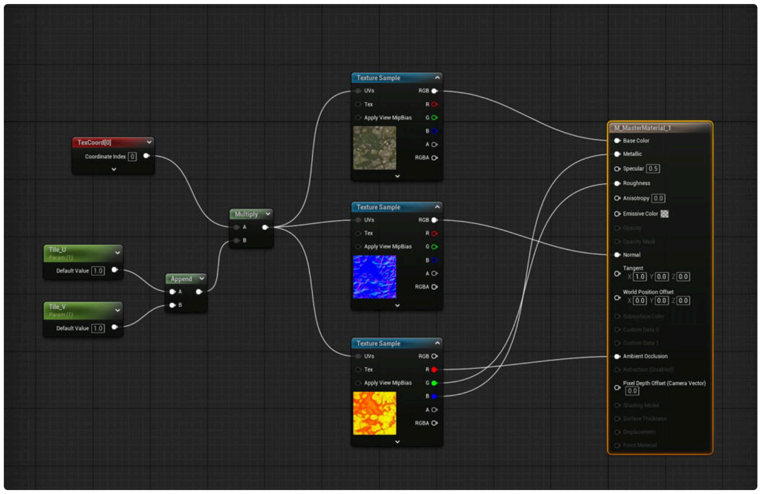
1. In the **Content Browser**, click **Import** at the top of the browser or right-click in the folder and select **Import to /Content/Textures**.
2. Import your texture files (e.g., `T_CreekRocks_BaseColor.png`, `T_CreekRocks_Normal.png`, `T_CreekRocks_ORM.png`). * *”**T_**” as a prefix designates that this is a Texture. ***ORM** is Ambient Occlusion, Roughness and Metallic*



Set Details based on image

3. Verify import settings in the **Texture Editor**:
   - Double Click on Texture to open in the **Texture Editor**.
   - Set **Texture Group** to `World` or `Character` based on asset type.
   - For Normal maps, enable **Normal Map** in the texture settings.
   - Disable **sRGB** for Normal, Metallic, and Roughness maps to preserve linear data. Double Click on each of these files and uncheck **sRGB**.
4. **Save all textures.**
   ∗Note that all of these are usually set automatically in Unreal during import with the exception of the sRGB setting in the packed map or maps that are not using color data (ORM or ARM) but it’s still a good idea to verify that these settings in the Texture Editor are correct.

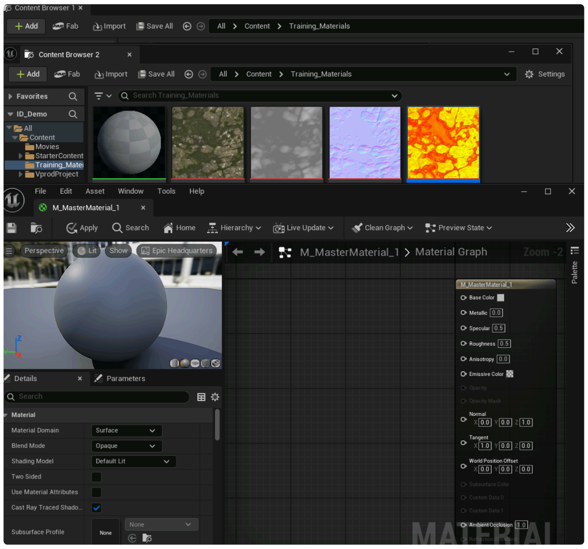
Step 4: Build the Material Graph



Finished Material Graph

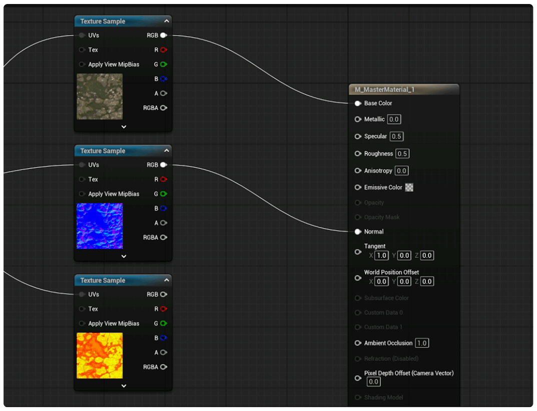
Textures are often optimized for Unreal by being packaged in the same textures. Meaning the Ambient Occlusion, Roughness, and Metallic can be separated by different channels (RGB) in the same texture. Normal maps separate so when building a basic material, you could have an Albedo/Diffuse map, a Normal Map and a Packed Map.

1. Double Click the `M_MasterMaterial_1` to open the **Material Editor**
2. We are going to drag the imported materials into the **Material Editor**
3. From the **Content Browser**, drag the texture into the **Material Editor**:
   - Drag each texture from the **Content Browser** into the graph grid area to auto-create **Texture Sample** nodes (containers for the textures).
   - Alternatively in the Material Editor, you can Press `T` and Left Click in the graph to add a **Texture Sample** node for each texture (Base Color, Normal, Metallic, Roughness) then in the Details, add the texture to the Sample.



Material
Material Domain: Surface
Blend Mode: Opaque
Shading Model: Default Lit
Two Sided: □
Use Material Attributes: □
Cast Ray Traced Shadows: ☑
Subsurface Profile: None

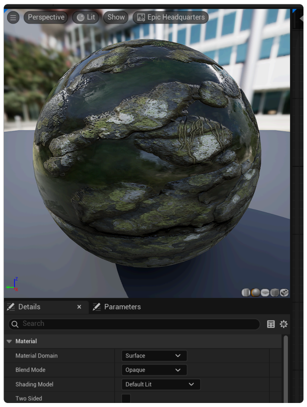
4. Connect the nodes to the main material node:
   - **Base Color**: Connect the RGB output of the Base Color texture to the **Base Color** input.
   - **Normal**: Connect the RGB output of the Normal texture to the **Normal** input.
   - **Ambient Occlusion:** Connect the R (red channel) output of the Metallic texture to the **Metallic** input
   - **Metallic**: Connect the R (red channel) output of the Metallic texture to the **Metallic** input.
   - **Roughness**: Connect the R (red channel) output of the Roughness texture to the **Roughness** input.



Connecting the Occlusion, Roughness and Metallic (RGB)

5. (Optional if you don't have an ORM Texture) Add a **Scaler Parameter** node (press S and LMB click) for fine-tuning:
   - Set a value (e.g., 0.5) and connect to **Roughness** or **Metallic** to adjust if textures lack variation.
6. If using a **Masked** blend mode, connect the alpha channel of the Base Color texture to the **Opacity Mask** input.

Step 5: Preview and Adjust



Material Domain: Surface

Blend Mode: Opaque

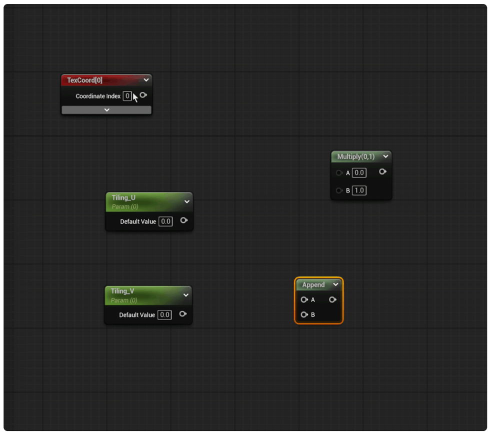
Shading Model: Default Lit

Two Sided

Material Preview in the Material Editor

1. In the **Material Editor**, use the **Preview Window** to view the material on a sample mesh (e.g., sphere or cube).
2. Adjust texture tiling if needed by creating a UV controller by creating a UV Hierarchy of nodes:

Create the Nodes to Control UVs

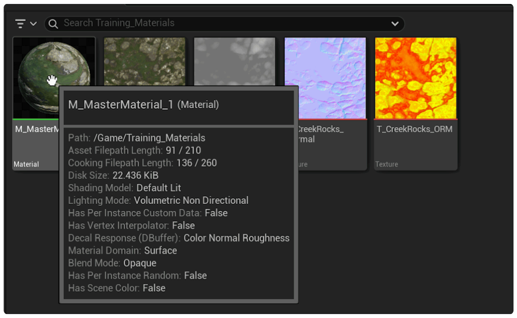


Connect the nodes and set Scaler Parameters to 1

- Add a **Texture Coordinate** node (press `U` and LMB click).
- Add a **Scaler Parameter** node (press `S` and LMB click, then Name it **Tile_U** and set the Default value to `1`).
- Add a **Scaler Parameter** node (press `S` and LMB click, then Name it **Tile_V** and set the Default value to `1`).
- Add a **Multiply** node (press `M` and LMB click).
- Add a **Append** node
  a. Either **Right-click** anywhere in the Material Graph or hit **Tab**
  b. Type "**Append"** in the search box
  c. Select **Append Vector** from the list
- Connect the **Multiply** node to the **UVs** input of each **Texture Sample** node.
- Connect the **Texture Coordinate** to the **Multiply "A"** input node.
- Connect the **Append** node to the **Multiply "B"** input node.
- Modify the default values of the Scaler Parameters name Tile_U and Tile_V to scale UVs (e.g., for tiling textures).

3. Test lighting by rotating the light source in the **Preview Window** to ensure normal maps and roughness behave correctly(press `L` in the **Preview Window**, Hold LMB and move the mouse around).

Step 6: Create a Instance Material



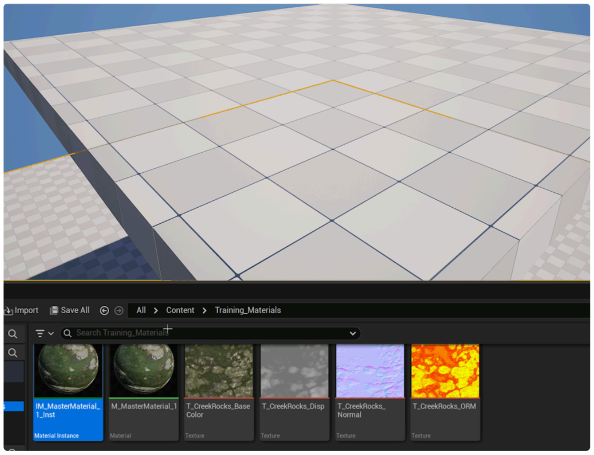
M_MasterMaterial_1 (Material)

Path: /Game/Training_Materials
Asset Filepath Length: 91 / 210
Cooking Filepath Length: 136 / 260
Disk Size: 22.436 KiB
Shading Model: Default Lit
Lighting Mode: Volumetric Non Directional
Has Per Instance Custom Data: False
Has Vertex Interpolator: False
Decal Response (DBuffer): Color Normal Roughness
Material Domain: Surface
Blend Mode: Opaque
Has Per Instance Random: False
Has Scene Color: False

Create an **Instance Material**

1. Save the material (Ctrl+S or **File > Save**).
2. In the **Content Browser**, locate `M_MasterMaterial_1`.
3. Right Click on the Material and chose **Create Instance Material**,
4. Rename `M_MasterMaterial_1 to IM_MasterMaterial_1`.
   (**IM is short for Instance Material, the instance material will allow you to change parameters on duplicate instances of the material*)

Step 7: Apply the Material



Apply and adjust the Instance Material

1. Save the Instance material (Ctrl+S or **File > Save**).
2. In the **Content Browser**, locate your 3D asset you want to apply the material to.
3. Drag the material onto the asset in the **Viewport** or assign it via the **Details** panel under **Material Slots**.
4. Adjust the Scaler Parameters in the Instance Material to scale the texture as needed. (.01 and up)

Step 8: Optimize and Save

1. In the **Material Editor**, check the **Stats** panel to ensure shader complexity is reasonable (e.g., instruction count below 500 for simple materials).
2. Optimize by reducing texture resolutions if performance is an issue (e.g., 2K instead of 4K for minor assets).
3. Save all assets and compile the material.
4. Package the project or test in a level to confirm compatibility.

Troubleshooting

- **Black Material**: Check if textures are properly connected or if sRGB is incorrectly enabled on non-color textures.
- **Normal Map Issues**: Ensure the Normal texture is set to **Normal Map** in the Texture Editor.
- **Performance Lag**: Reduce texture sizes or simplify the material graph (e.g., remove unnecessary nodes).
- **Incorrect Lighting**: Verify the **Shading Model** and ensure Normal map is correctly formatted (e.g., OpenGL format).
- **Parameters Greyed Out and not working**: To adjust parameters in an Instance Material, make sure to check them to override the default values or inputs in the Master Material.

Best Practices

- Use consistent naming conventions (e.g., `M_` for materials, `T_` for textures, IM_ for instance materials).
- Store materials and textures in organized folders (e.g., `Content/Materials/Characters`).
- Test materials under multiple lighting setups to ensure versatility.
- Document custom nodes or complex setups in the **Material Editor** using **Comment** nodes (right-click > Add Comment).

References

- Unreal Engine Documentation: [Materials]
- Texture Preparation: Use Substance Painter or Photoshop for PBR texture creation.
- Optimization Tips: [UE5 Performance Guidelines]