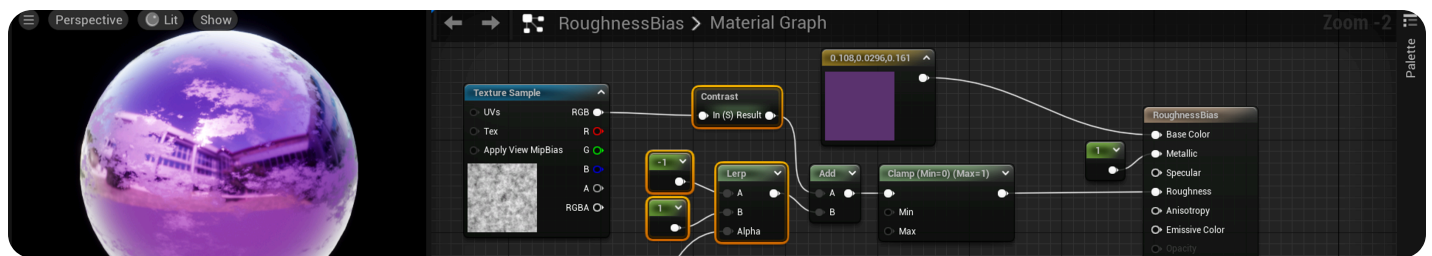# Material Editor Guide

Learn about Unreal Engine's node-based material editor to create materials which define the appearance and surface properties of objects in your scene.


The **Material Editor** is a node-based graph interface with which you can create **Materials** (also called shaders) for your project. Materials define the appearance and surface properties of the objects in your scene, including Static and Skeletal Meshes, Landscapes, UI, and VFX.

The pages of this user guide will introduce you to the Material Editor interface while providing a high level overview of Material creation in Unreal Engine.

> ⓘ   It is recommended that you read the Essential Material Concepts page before continuing below. This page provides critical background information about the Materials workflow in Unreal.

Material Creation Workflow

A very simple demonstration of the Material creation process is shown in the video below.
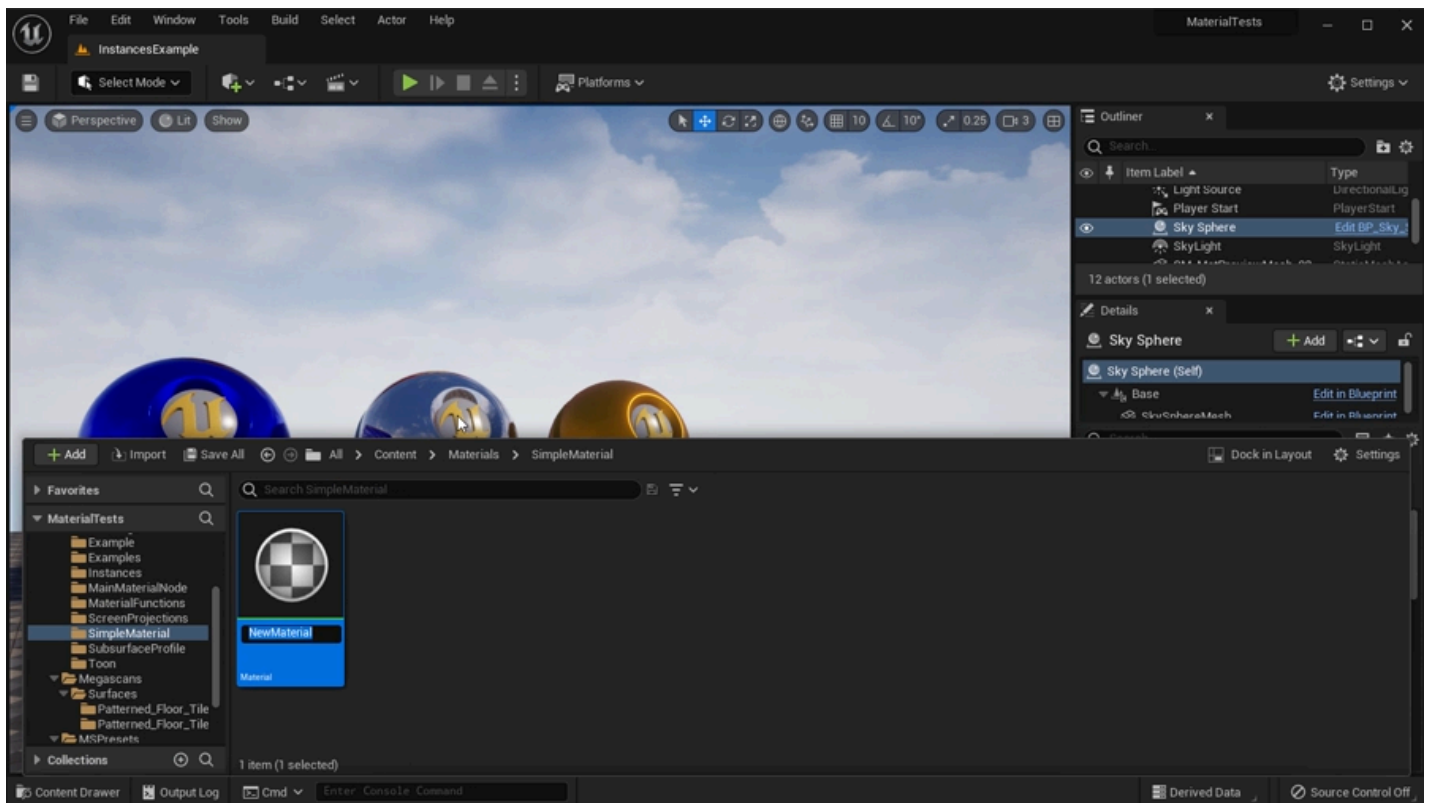
As shown above, these are the steps to create a simple Material:

1. Create a new **Material** asset in the Content Browser.

2. Open the Material Editor.

3. Place Material Expressions and Functions in the Material Graph.

4. Connect Material Expressions to inputs on the Main Material Node to define the Material's attributes.

5. Compile and Save the Material.

6. Apply the Material to objects in the Level.

The bulk of the Material creation process occurs during steps three and four, and this is where you perform most of the work that goes into defining a specific type of surface and adding your creative touches. **Material Expressions** and **Functions** are the building blocks of Unreal Materials, with each one performing a specific action in the Material graph. By combining these nodes in unique ways you can define an endless array of physical surfaces.

Creating a new Material

1. There are two ways to create a new Material asset:
   a. Click the **Add (+)** button on the left side of the Content Browser and select **Material** from the list.
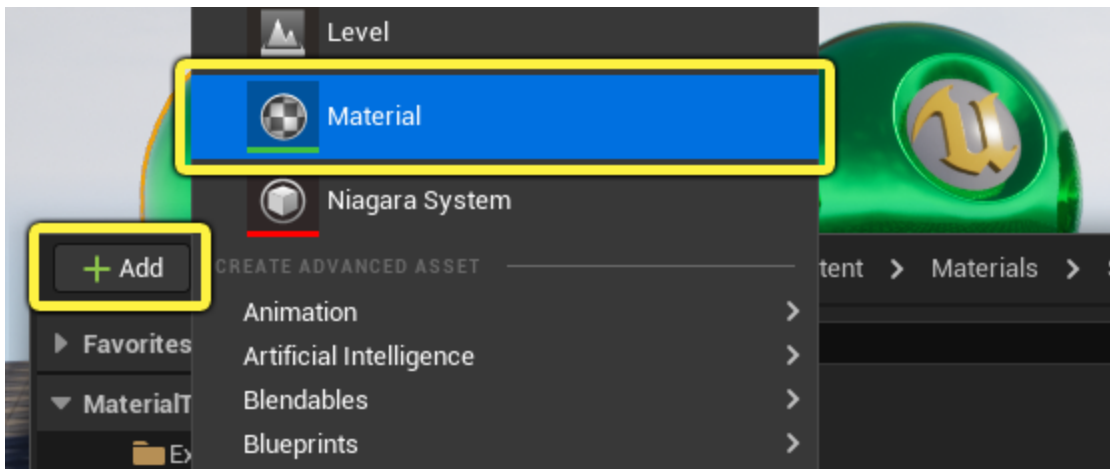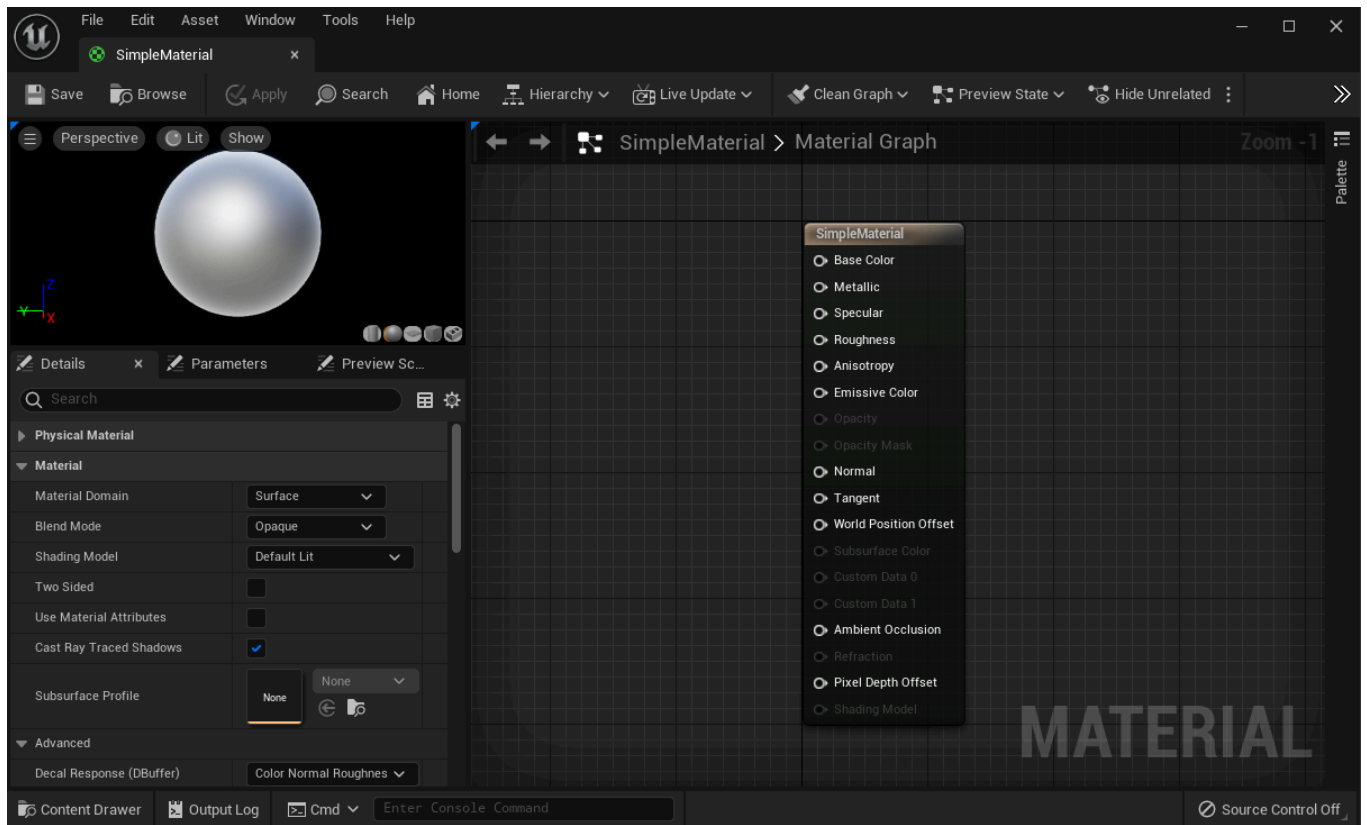
b. You can also **right-click** anywhere in the background of the **Content Browser** and select **Material** from the context menu.
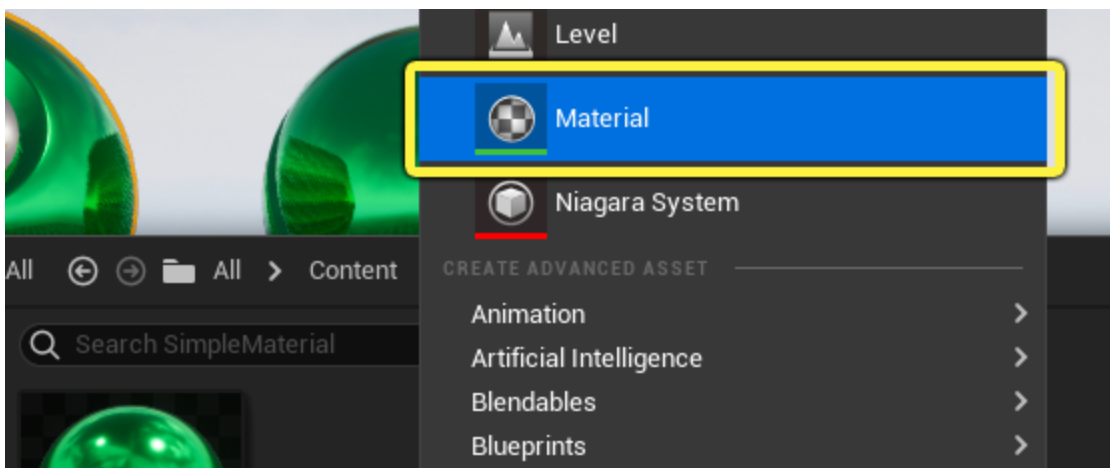
2. A Material is created in the Content Browser. Rename the Material and then double-click the thumbnail to open the Material Editor.
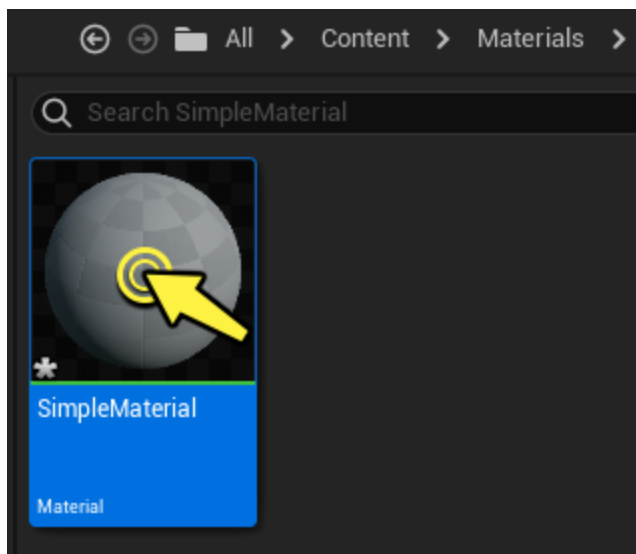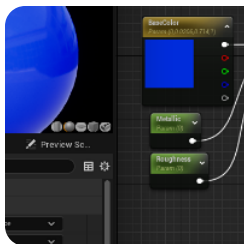

3. The **Material Editor** opens.

Using the Material Editor

The five pages linked here provide detailed information about how to use the various parts of the Material Editor. These documents elaborate on the steps shown in the video at the top of the page, and are meant to teach you the practical mechanics of Material creation. It is recommended you read them in the order presented below to gain a thorough understanding of how to interact with the tools available in the Material Editor.
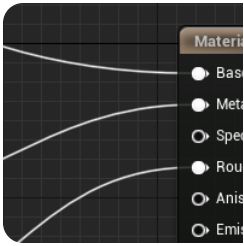

Material Editor UI

A breakdown of the Material Editor user interface.
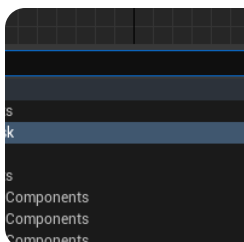
Placing Material Expressions and Functions

Guide for placing Material Expressions and Functions inside the Material graph.
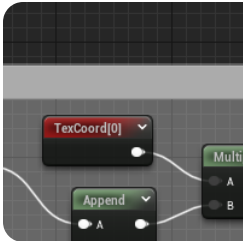
Using the Main Material Node

Guide for setting up and using the Main Material Node
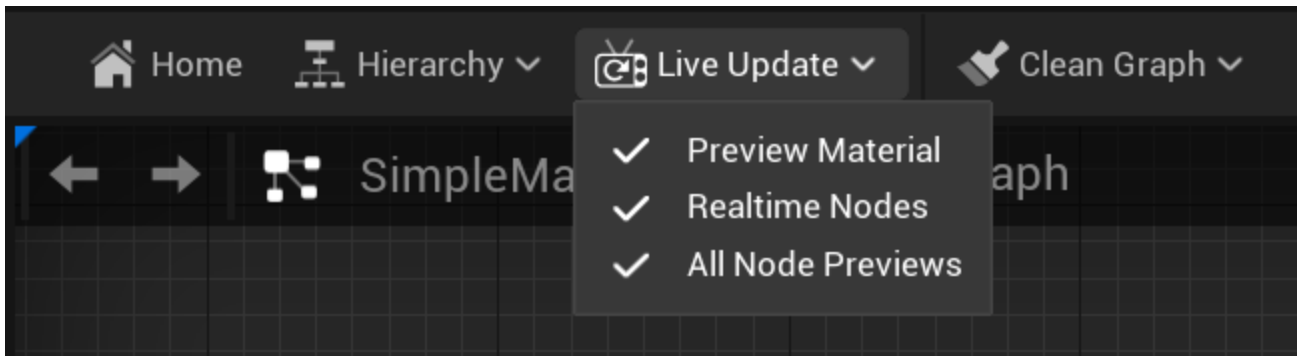

Previewing and Applying your Materials

A guide for previewing Materials and applying them to Actors.

Organizing a Material Graph

How to use commenting, reroutes, and other strategies to organize Material Graphs and keep them easy to read and edit.

Live Update Options

When making changes to a Material network, it is useful to see immediate feedback from each change in realtime. The Material Editor offers three options under the **Live Update** menu to control which elements in the Material Editor update in real time — **Preview Material**, **Realtime Nodes**, and **All Node Previews**.

Each of these options gives you the ability to toggle instantaneous feedback for a different part of the Material Editor.

- **Preview Material -** This option allows for any changes to update automatically in the Material Preview viewport in real time without having to use the **Save** or **Apply** buttons.
- **Realtime Nodes -** Nodes affected by time-based functions, such as panners, will update in realtime.

- **All Node Previews -** This option allows for each node within the network to have its shader recompiled whenever a change is made. These changes include creation of new nodes, deletion of nodes, node connections and disconnections, and changes in properties. This recompilation is necessary so that the Material Preview drawn at that node is up to date. However, recompiling these intermediate shaders can be time-consuming, especially if your Material contains a large network. If you're experiencing long wait times after every change, you may want to deactivate **All Node Previews** option.

Consider the example below, in which a panning grid texture is multiplied by a vector expression, which supplies color.
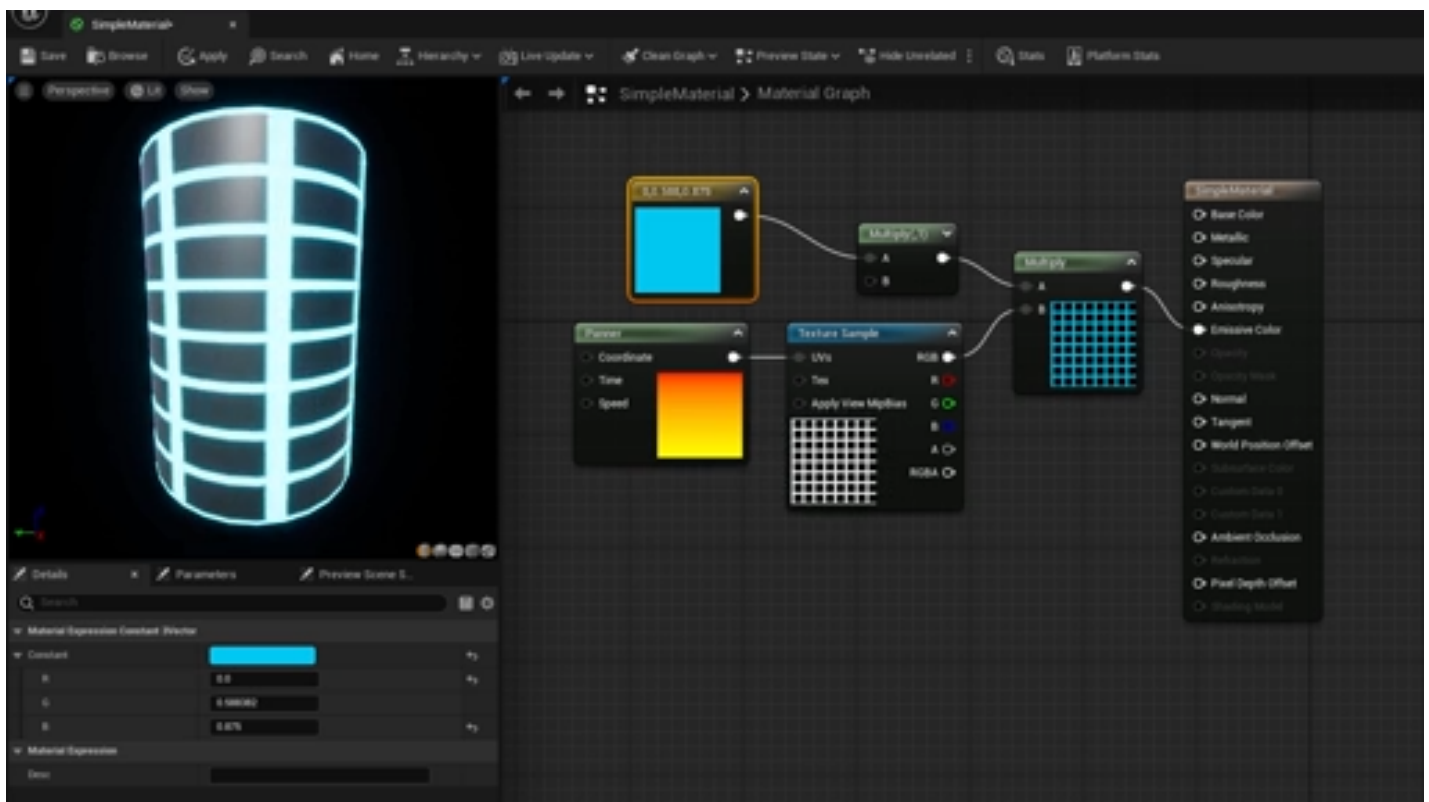
- In this example, activating **Realtime Nodes** causes the texture to pan in real time in the node's preview thumbnail. If **Realtime Nodes** is deactivated, the texture remains stationary even though the Panner is telling it to move. You may, however, notice small updates as you move your mouse around the graph area.
- If you change the color in the vector expression, the change only appears in downstream nodes if **All Node Previews** is enabled. With this option disabled, the change in color does not propagate to node thumbnails even though the color in the vector expression indeed changed.

When **All Node Previews** is deactivated, you can force-update all previews manually by pressing **Spacebar**. Fast iteration can be achieved by disabling **All Node Previews** and then pressing spacebar whenever you want to view your changes.

Compiler Errors

Each time a change is made to the Material network, the Material must be compiled to see the changes. Compiler errors occur if any **required inputs** on an expresssion within the Material networks are passed the wrong type of data, or do not receive any connections.

These types of errors are indicated in two places.

- The node that is generating the error will display **"ERROR!"** along its bottom.

- The **Stats** panel will also display an error message that identifies what is causing the compiler to fail. If your Stats window is not opened, you can open it by going to **Window** > **Stats**.

Compiler errors let you know that a problem exists and provide information about the type of Material expression they occurred on and the description of the error.
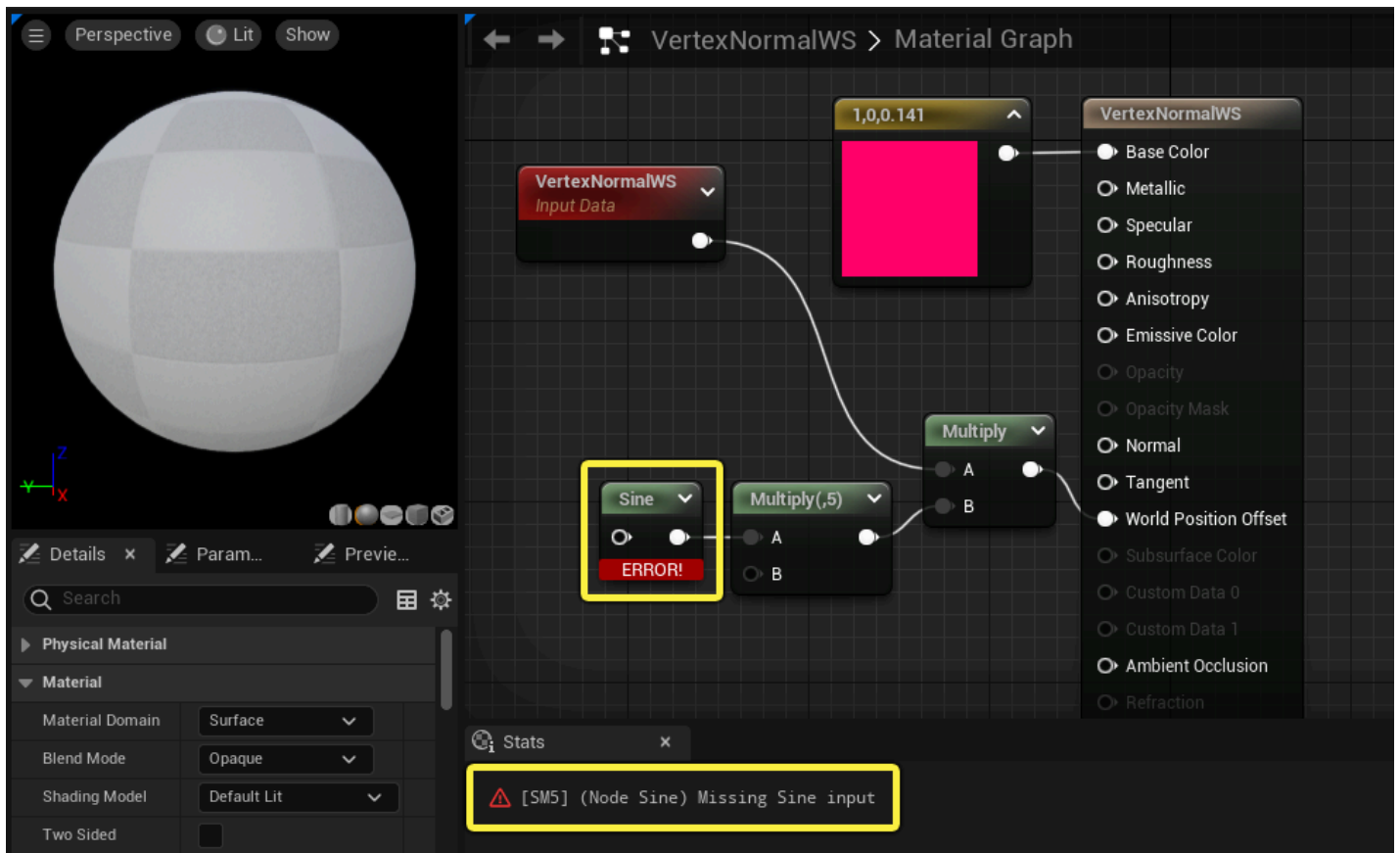
Sine node displays an error message becuase a required input is not receiving any data. This error is also displayed in the Stats panel.

Material Graph Search

The search functionality in the Material Editor enables you to quickly find any nodes (including comments) within the Material network that contain a specific piece of text in their description or certain other properties specific to individual types of nodes. This makes it easy to add identifying keywords to nodes and jump to them at a later time without sifting through the network of nodes in your graph haphazardly.

You can open this tab by going to **Window** > **Find Results**.

Typing a full or partial keyword into the search box will perform a search against the properties of the nodes present within you your material graph. The currently selected result will be brought into view and highlighted.
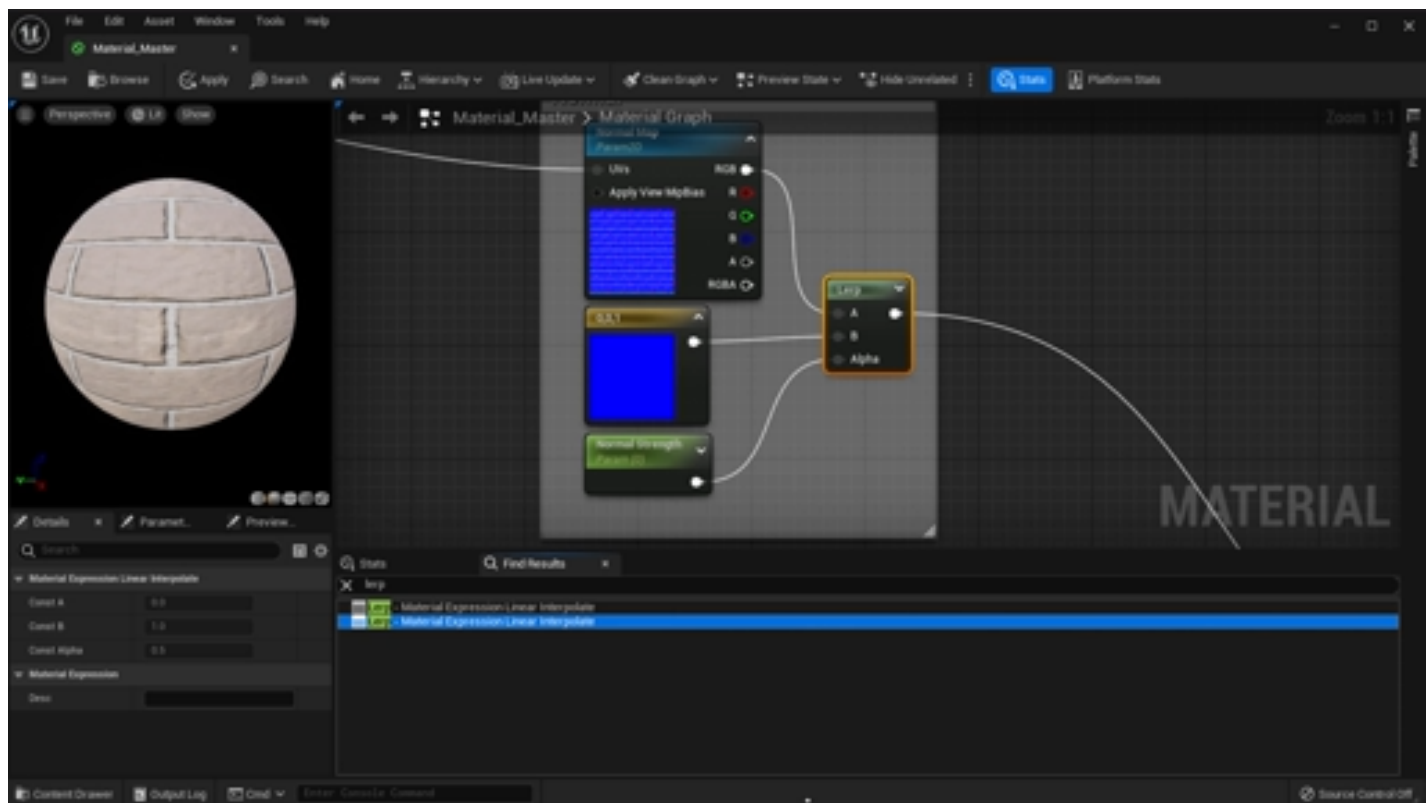
Searches are performed against the following property values:

Searched Properties	Expression Type
Desc	All Nodes
Texture	Texture Sample
ParamName	Parameters
Text	Comment
Font	Font Sample
Material Function	MaterialFunctionCall

You can also search for specific types of expressions by using the `NAME=` switch with your search. For example, to find all texture samples, you could use the following search:

```
NAME=texture
```

When a new match is clicked in the **Search** panel, it is brought into view in the graph table if it is not already visible.

To clear a search, simply press the **Clear** (X) button.