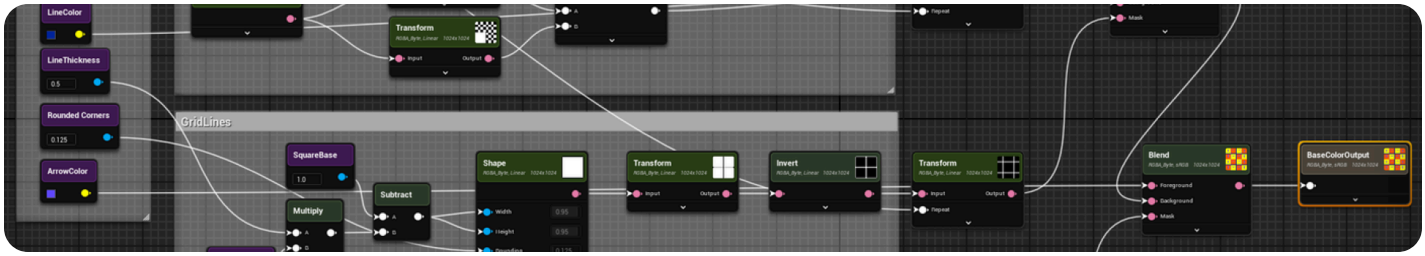
Making your First Texture Graph

Procedurally create a UV checker pattern texture with the Texture Graph Editor.



Learn to use this **Experimental** feature, but use caution when shipping with it.

The **Texture Graph Editor** is a node-based interface for procedurally creating textures in Unreal Engine. This guide steps you through making a texture graph that generates a custom UV checker pattern.

The graph provides customization for the number of tiles, colors, grid lines, and arrows. Constructing this graph explores some core concepts and workflows associated with a texture graph.

Before building your first graph, it is helpful to review the Getting started with Texture Graph guide for an overview of the editor and some useful general concepts.

Loading the Plugin

The Texture Graph Editor is an experimental plugin not loaded by default when you start the engine.

To enable the plugin, follow these steps:

1. In the **menu bar**, select **Edit > Plugins**.
2. In the search bar, type "texture graph".
3. Enable the **TextureGraph** plugin, and select **Yes** in the dialog popup.
4. Restart the engine.

Texture Graph Asset

To create a new texture graph, follow these steps:

1. Open the **Content Drawer** and click **Add > Texture > Texture Graph**.
2. Rename the new asset to "TG_UVChecker".
3. Save the asset and then double-click it to open the editor.

Build the Base Checker Tile

The base of the pattern is a two-by-two tile that you can repeat and edit.

This guide builds the pattern from scratch to explore some workflows and user controls.

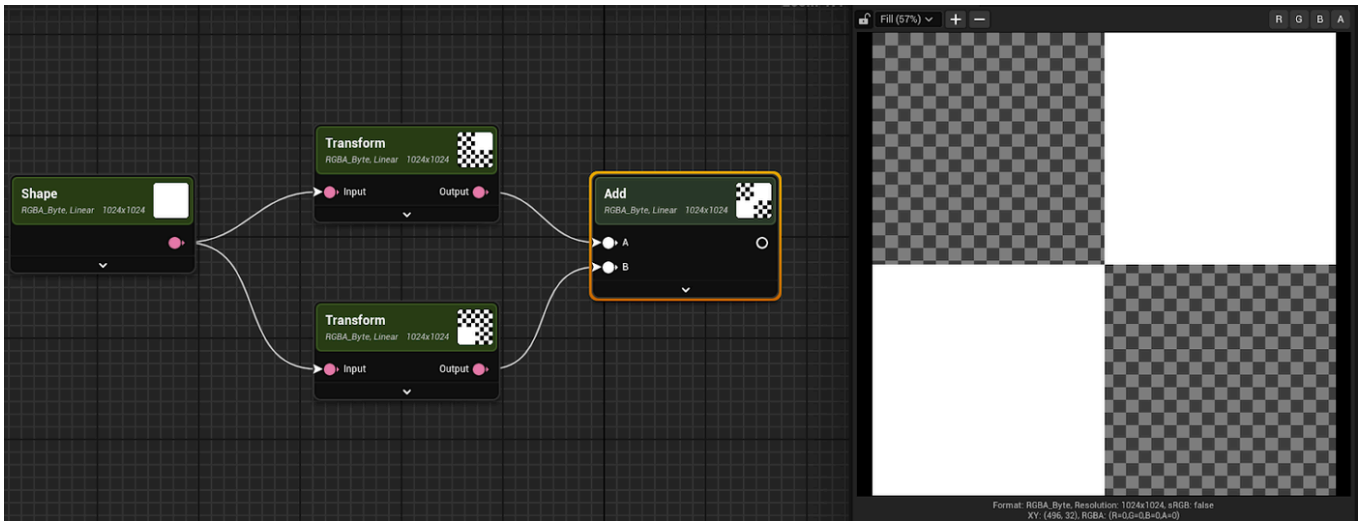
> If you need a quick checker pattern, you can explore the **Pattern** node, which has options for several standard pattern types.

To start creating the checker pattern, follow these steps:

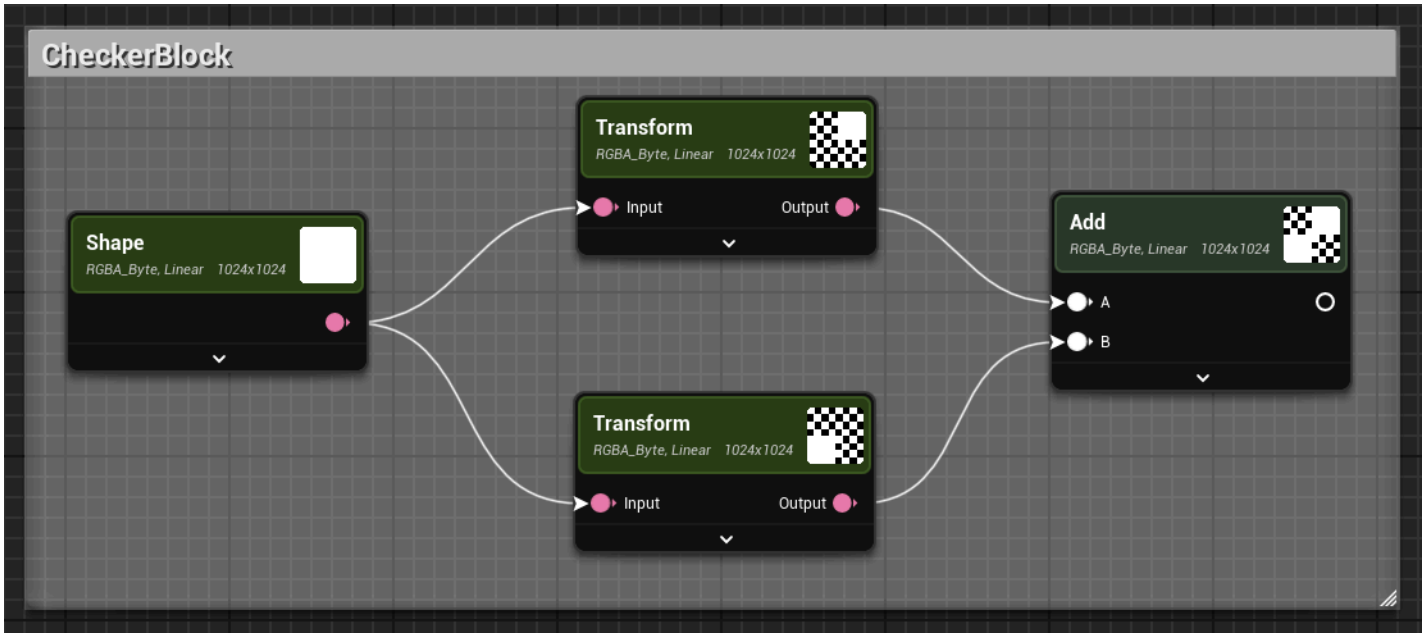
1. From the node palette, access the **Procedural** section and then drag the **Shape** node into your graph. Expand the node by clicking the down arrow to view all the possible options.
2. Change the **ShapeType** to a rectangle.
3. Click and drag out the **Output** pin. When you release the selection, the node palette opens. Search and select **Transform** from the list. The Shape and Transform nodes automatically connect.

a. When creating nodes using this technique, remember the connection connects to the top input value.

4. Expand the **Transform** node and adjust the **Coverage** to 0.5, 0.5. Adjusting the coverage in transform scales the input image (in this case, a simple rectangle) to the specified value. By default, the fill color is empty with color or alpha value. You can set the fill color in the details.
5. From the **Shape** node, drag out a second **Transform** node. Set the **Coverage** on the second transform to 0.5, 0.5 and the **Offset** to 0.5, 0.5. You can now minimize the transform nodes so they occupy less space.
6. Right-click in the graph view, and from the menu, find the **Add** node by scrolling down to math or searching.
7. Connect the output values of the Transform nodes to the A and B inputs of the **Add** node. The result is a single 4-square checker tile.



8. As you build out your graph, it's important to keep it organized. Select all the nodes, right-click the graph, and then search and select **New Comment**. The comment box contains the four nodes. Name the comment "CheckerBlock".

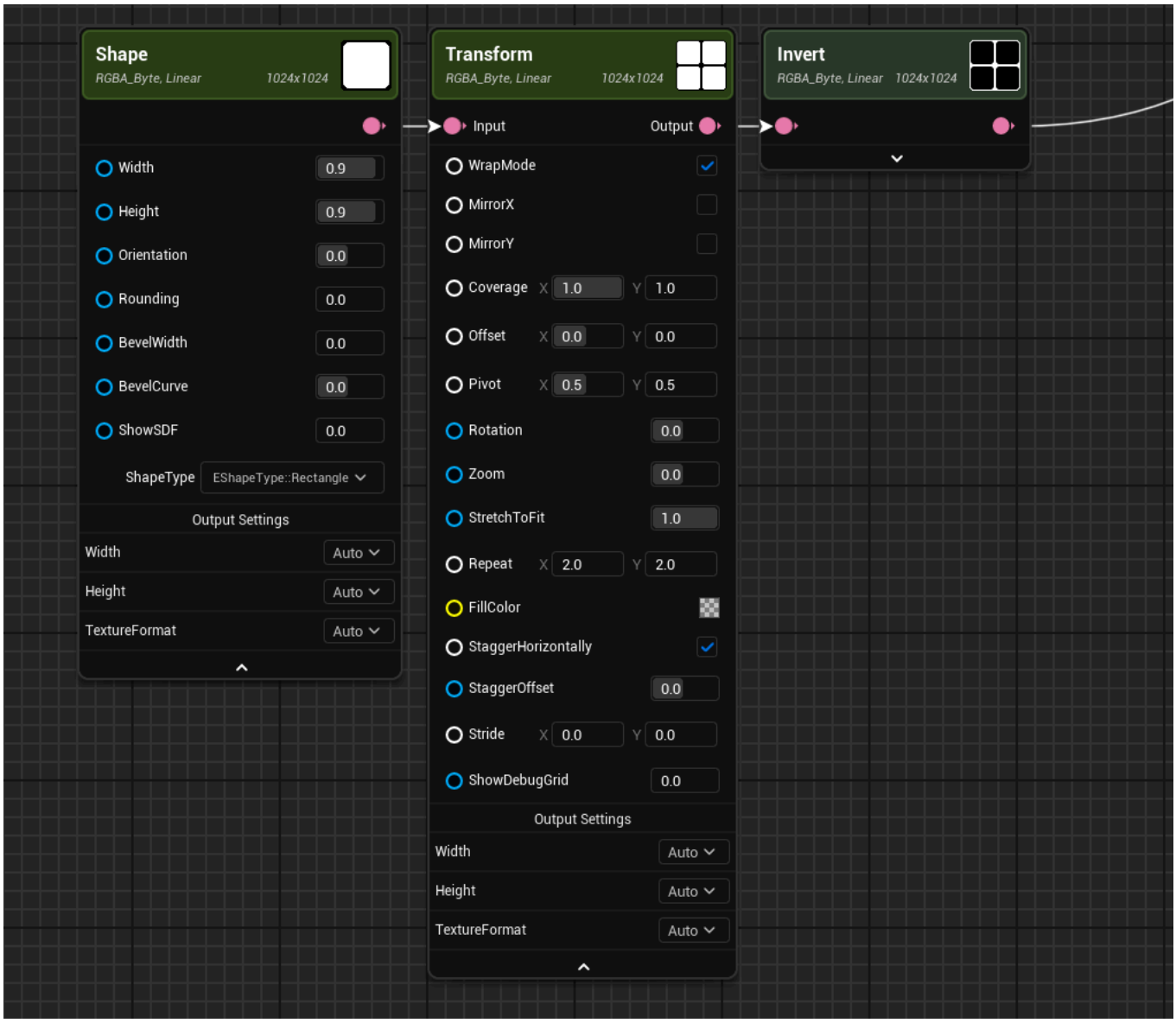


Creating Dividing Lines

With the basic checker block, you can overlay dividing lines and control the thickness and corner radius value.

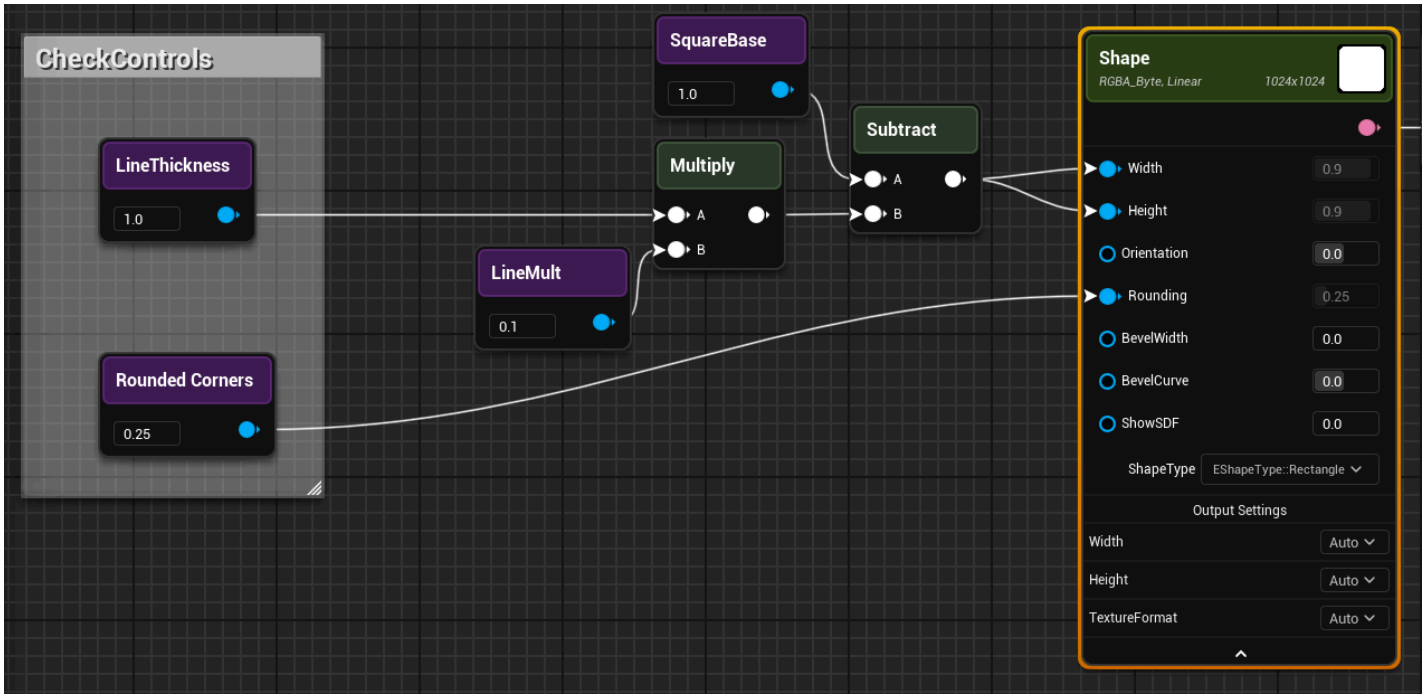
To add the diving lines, follow these steps:

1. Create a **Shape** node, set the shape to rectangle, and set the width and height to 0.9.
2. Drag out the output pin of the Shape node, then search and select **Transform**.
3. In the Transform node, set the **Repeat** value to 2.0,2.0.
4. Drag out the output pin, then search and select **Invert**. This node gives you the proper values for your line mask.
5. In the Invert node, enable **Clamp**.



6. With the basic structure for the lines created, you can add a mechanism to adjust the line thickness and corner radius of the lines. Right-click in the graph, then search and select **Scalar**.
7. Rename the node to "Rounded Corners" by right-clicking the node and selecting **Rename**. Connect the node to the **Rounding** pin of the **Shape** node. Set the value to .25.
8. Repeat the process 3 times to create multiple scalar nodes and rename them to "Linethickness," "SquareBase," and "LineMult."1. Drag out the **LineThickness** pin, then search and select **Multiply** from the list.
9. Connect the **LineMul**t output to the **B** pin of the Multiply node and set the value to 0.1.
10. Drag out the **SquareBase** pin, then search and select **Subtract** from the list.
11. Connect the output of the **Multiply** node to the **B** value of the Subtract node.
12. Connect the output pin of Subtract to the **Width** and **Height** values of the **Shape** node.

13. Select all the nodes and create a new comment box around them to help organize your graph.



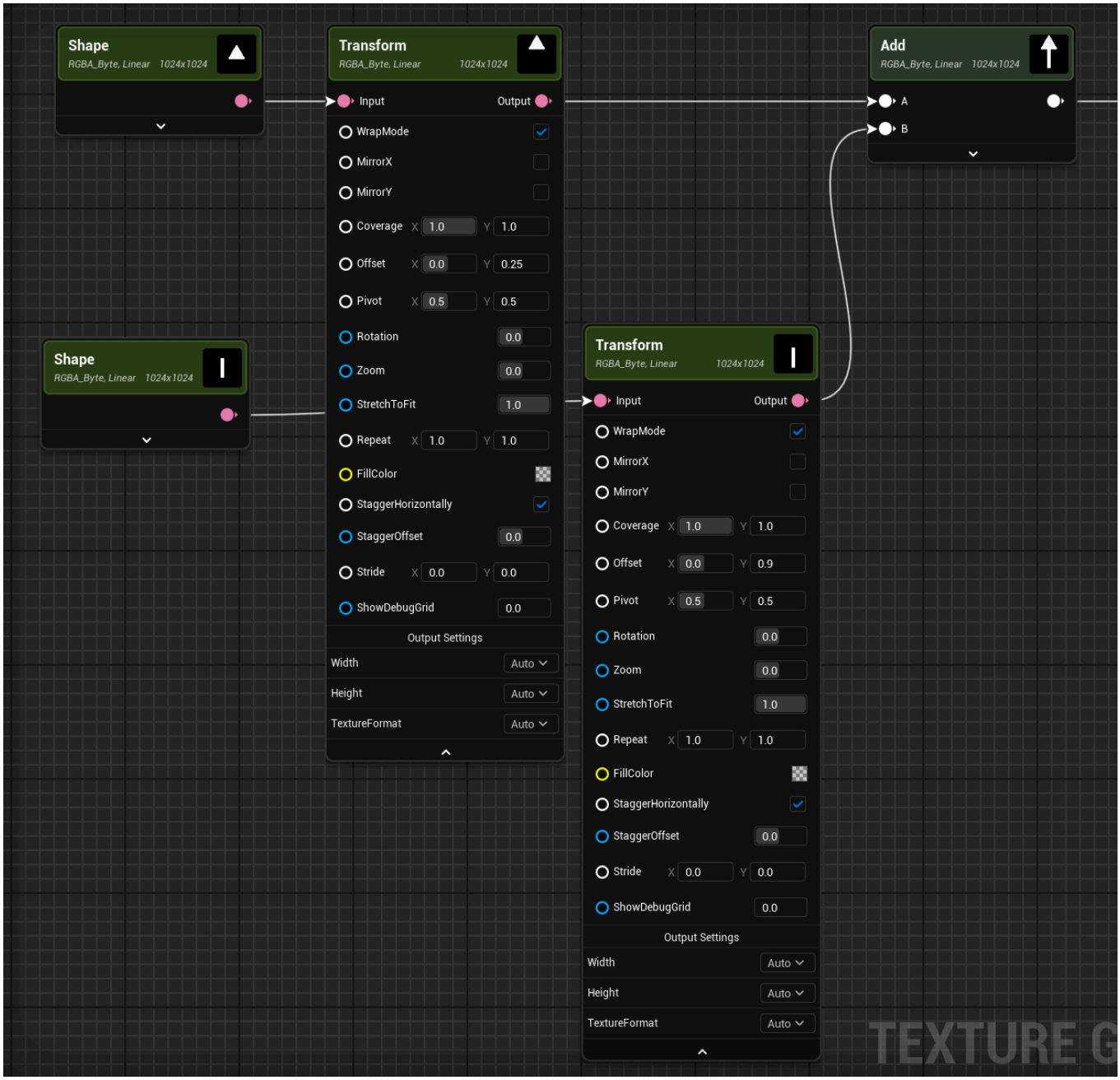
Creating Arrows

To help tell the orientation of the pattern, you can add some arrows to the center of each square.

To add this final element, follow these steps:

1. Right-click in the graph, then search and select **Shape**. Repeat this step to create two Shape nodes.
2. Set the shape of the first node to **Triangle** and then set the width to 0.6.
3. Drag out the output pin, then search and select **Transform**. Set the **Offset** value to 0.0,0.05.
4. Set the second Shape node to **Rectangle**, then set the width to 0.1 and height to 0.4.
5. Drag out the output pin, then search and select Transform. Set the **Offset** value to 0.0,0.8.
6. Right-click in the graph, then search and select **Add**. Connect the output pins of the Transform nodes to the Add node.
7. Select the **Add** node, then in the preview window, click the lock icon. This icon locks the preview to the node even when you select other nodes. The lock preview provides the

means for you to edit the transform values and see the final result.



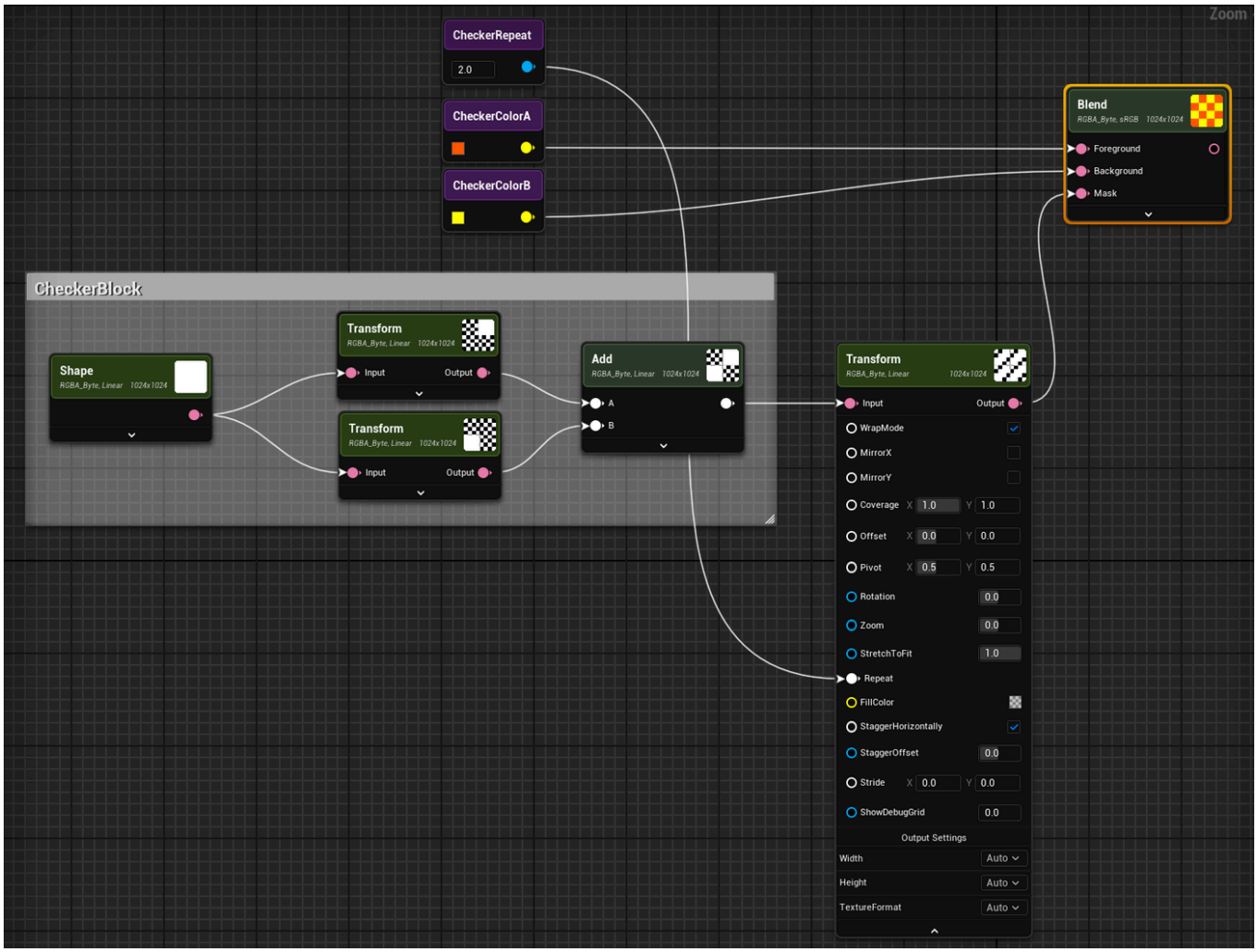
8. Transform the single arrow shape into our standard 4 square tile base. To do this, drag out the output pin from the **Add** node, then search and select **Transform**.
9. In the Transform node, set the **Repeat** value to 2.0, 2.0.
10. Uncheck the lock on the preview to see other node previews.
11. The arrows are a little too large. Set the **Zoom** value to -0.1 to scale each repeated tile. A negative value zooms out, resulting in a smaller arrow.
12. Zooming out exposes the empty area in between each tile. In the **Details** panel, click the **Fill Color** value and set the **A** (alpha) value to 1. The color is already set to black by default. Adjusting these values creates a solid mask.

13. Select all the nodes and create a new comment box around them to help organize your graph.

Connecting the Components and Adding Color

You now have your core building blocks to make a scalable checker pattern. You can add more controls similar to the grid line section to adjust the pattern for future use.

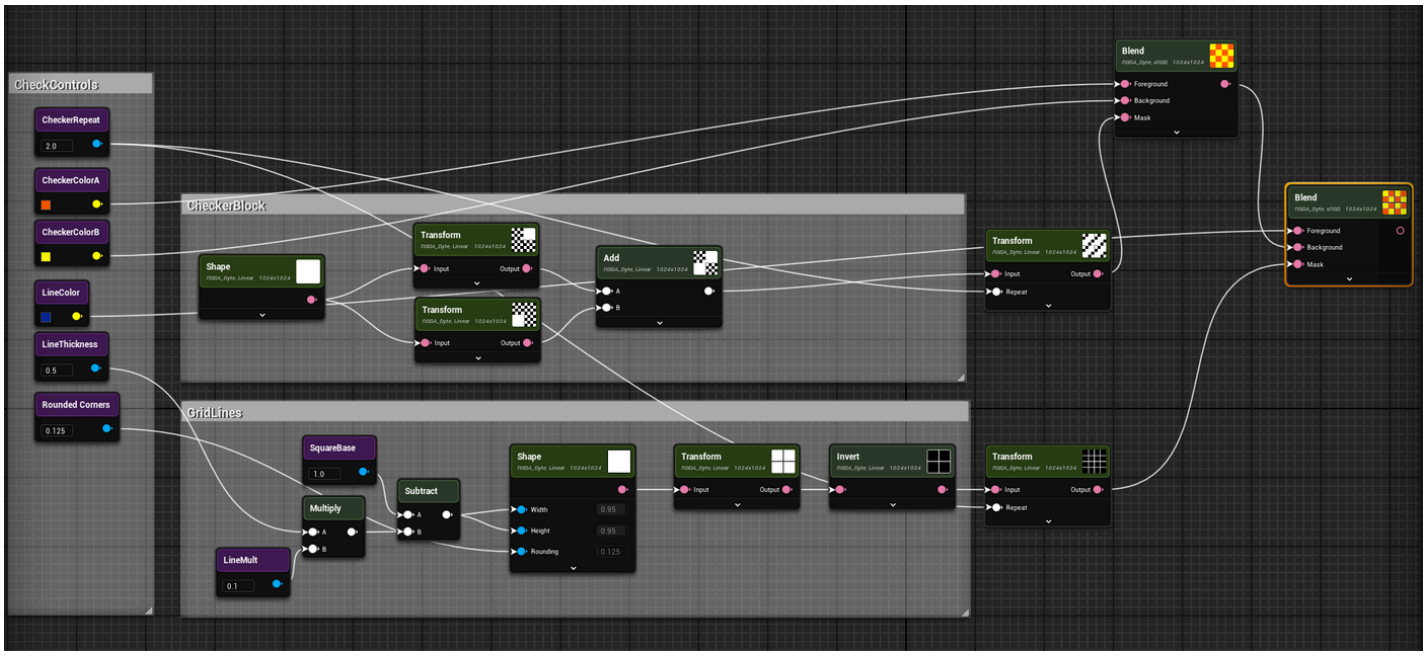
1. Start with the most important element, the checker color. Drag out 2 Color nodes from the node palette.
2. Rename the nodes to "CheckerColorA" and "CheckerColorB".
3. Adjust the **Color** nodes to your desired checker colors in the **Details** panel.
4. From **CheckerColorA**, drag out the output pin, then search and select **Blend**. Blend nodes are powerful functions with many features for complex blend operations. For this example, keep the default settings.
5. Attach the output pin of **CheckerColorB** to the **Background** value of the Blend node.
6. From the output pin of the **Add** node in your Checkerblock, drag out a new **Transform** node.
7. Create a new **Scalar** node and rename that to "CheckerRepeat".
8. Set the **CheckerRepeat** value to 2.0.
9. Connect the output pin of the **CheckerRepeat** to the **Repeat** pin of the newly created Transform node. The single scalar input applies to both X and Y repeat values.
10. Connect the output pin of the **Transform** node to the **Mask** value of the Blend node.



Grid Line Color

You can repeat a similar workflow for your grid lines. You already have controls for the line thickness and rounding, but you don't have a setting for your line color.

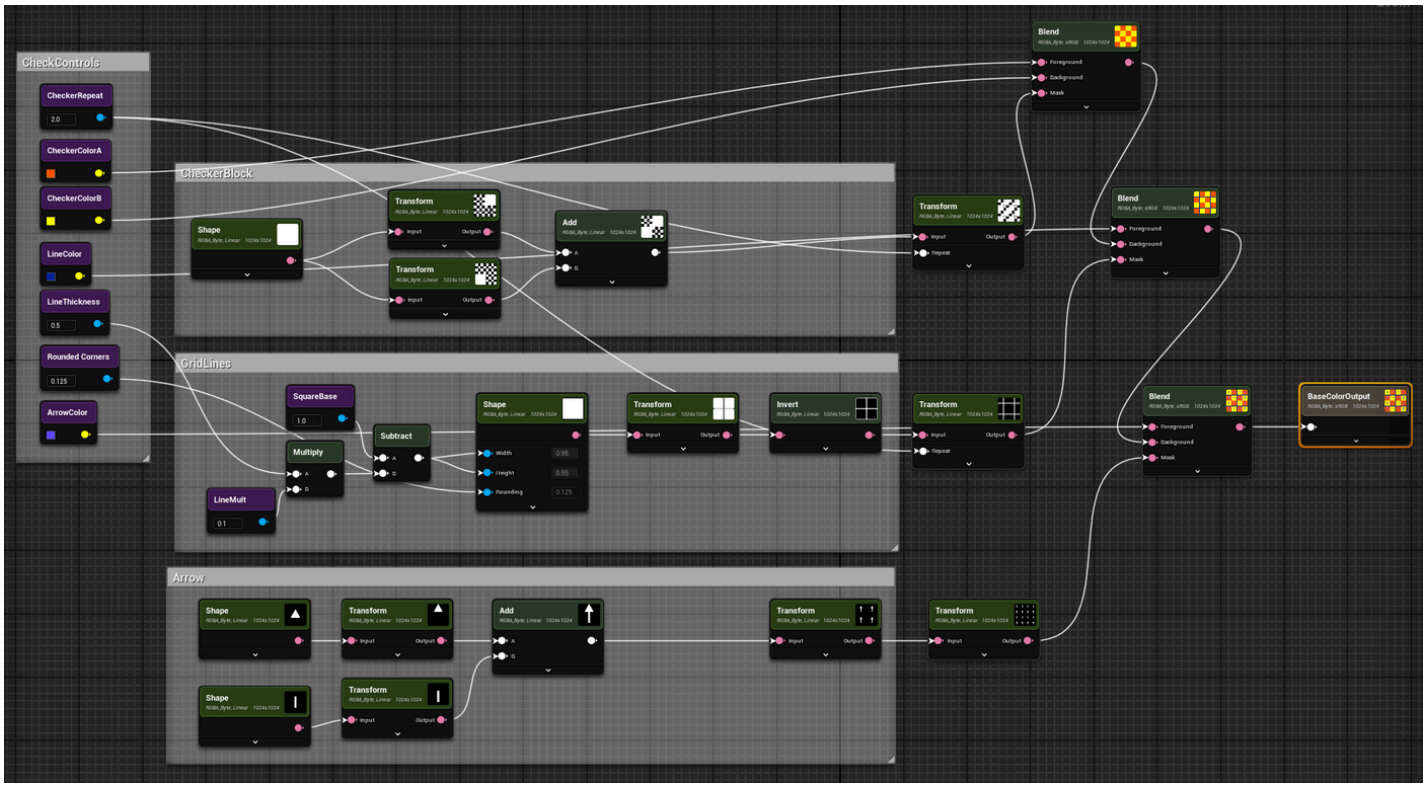
1. Create a new **Color** node and rename it "LineColor".
2. Adjust the **LineColor** node to your desired color in the **Details** panel.
3. Drag out the output pin of the **LineColor** node, then search and select **Blend**.
4. Connect the output pin of our previous checker Blend node to the **Background** input of the new Blend node.
5. In the GridLines comment box, drag out the output pin of the **Invert** node, then search for and select **Transform**.
6. Connect the **CheckerRepeat** node to the **Repeat** input of the new Transform node. You can use an output pin multiple times.
7. Connect the output pin of the **Transform** node to the **Mask** input on the new Blend node.



Arrow Color

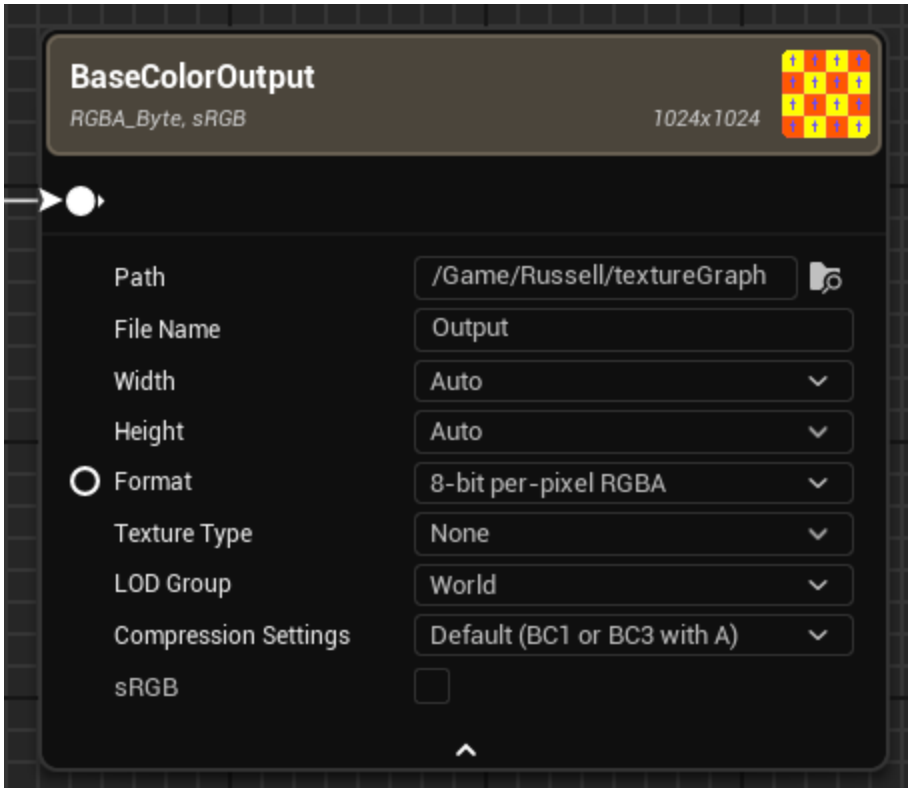
Repeat this process to add your arrows over the checker pattern.

1. Create a Color node and name it "ArrowColor".
2. Adjust the **ArrowColor** node to your desired color in the **Details** panel.
3. Drag out the output pin of the **ArrowColor** node, then search and select **Blend**.
4. Connect the output pin of our previous checker Blend node to the **Background** input of the new Blend node.
5. In the Arrow comment box, drag out the output pin of the **Transform** node, then search and select **Transform**.
6. Connect the **CheckerRepeat** node to the **Repeat** input of the new Transform node.
7. Connect the output pin of the **Transform** node to the **Mask** input on the new Blend node.
8. Drag out the output pin of the Blend node, then search and select **Output**. Rename the Output node to "BaseColorOutput".



Outputs and Export

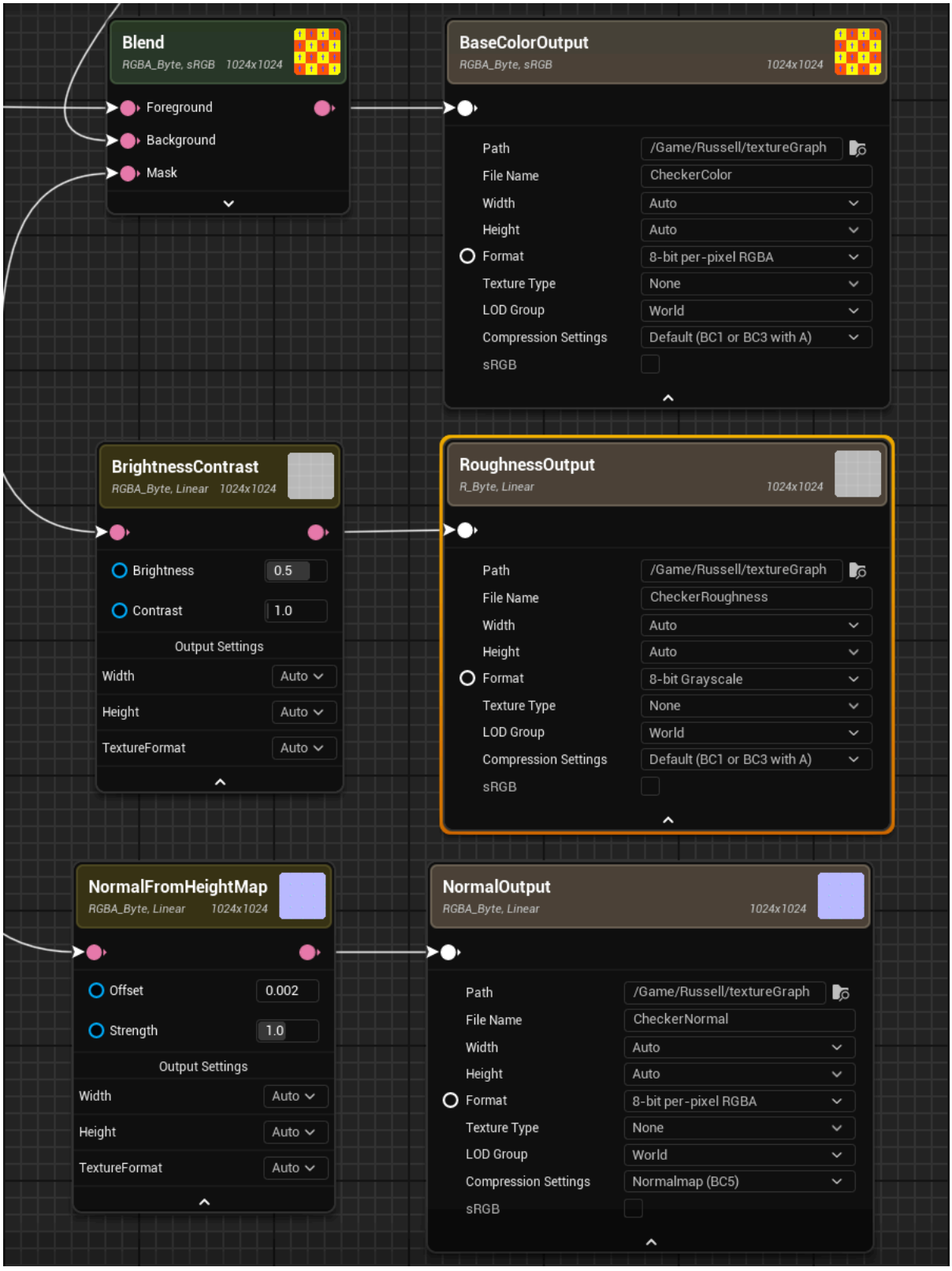
The **Output** node has settings for common texture attributes and the output name, folder path, and resolution.



You can add multiple outputs to a single graph. This workflow is useful when creating more complex texture sets for materials requiring a color, roughness, normal, and metallic map.

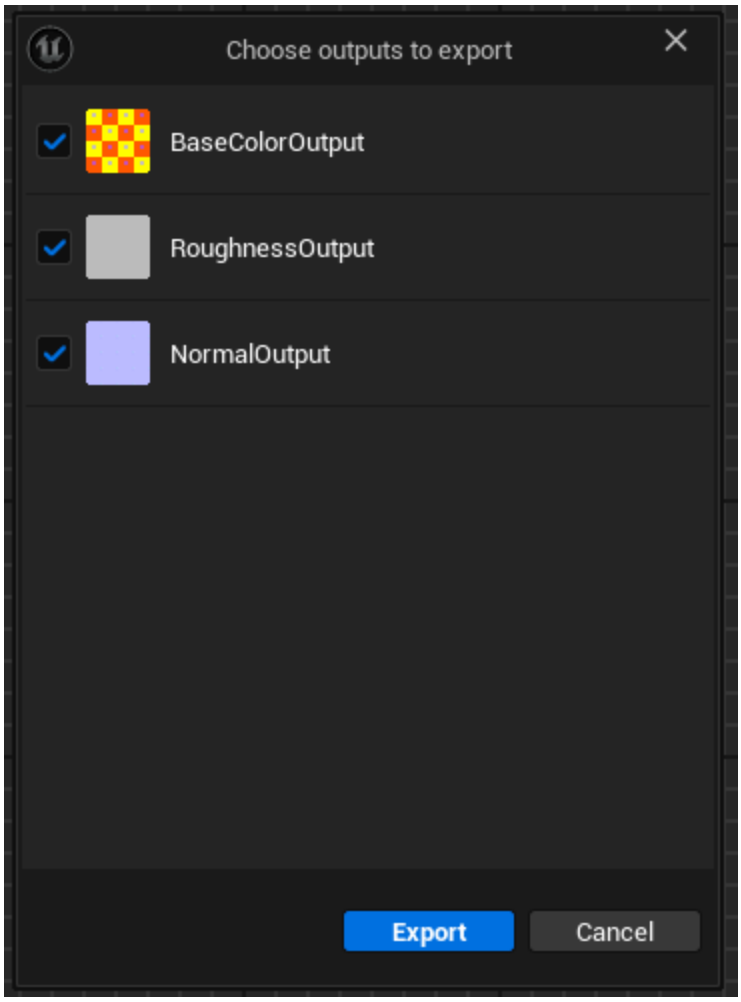
To set up multiple output types, follow these steps:

1. In the GridLines comment box, drag out the output pin of the **Transform** node, then search and select **BrightnessContrast**.
2. Adjust the Brightness to 0.5 to create a more flattened greyscale image.
3. Drag out the output of the BrightnessContrast node to create a new **Output** node. Rename the node to "RoughnessOutput".
4. From the previous Transform node, drag out the output pin, then search and select **NormalFromHeightMap**.
5. Drag out the output of the **NormalFromHeightMap** node to create a new **Output** node. Rename the node to "NormalOutput".
6. Adjust the output names and compression settings for each output node.



7. In the main menu bar, click **Export**.

The export window opens and shows the output nodes as potential maps to output. You can select or deselect these to iterate on one or all of the maps as needed. Click **Export** in the window to create the new texture maps.



You can use the Texture Graph Editor for a wide range of texture workflows, from basic edits to existing textures, creating new textures, packing, atlasing, or combining with Blueprints. You can use texture graphs as pipeline utilities to help with common processes.