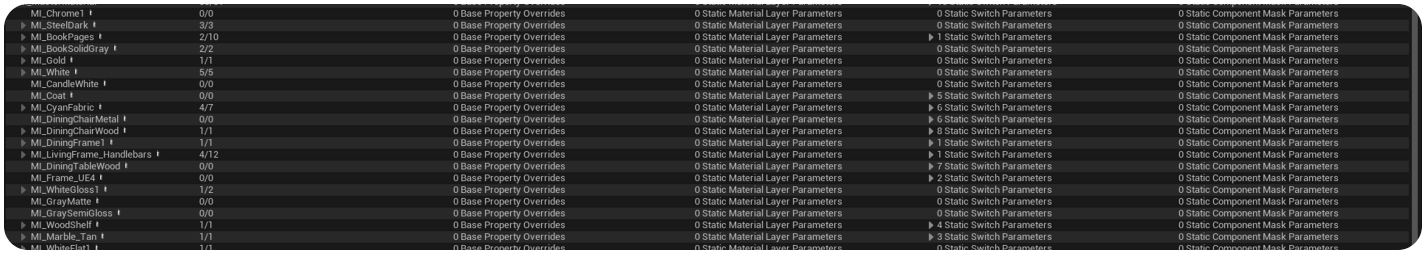
Material Analyzer

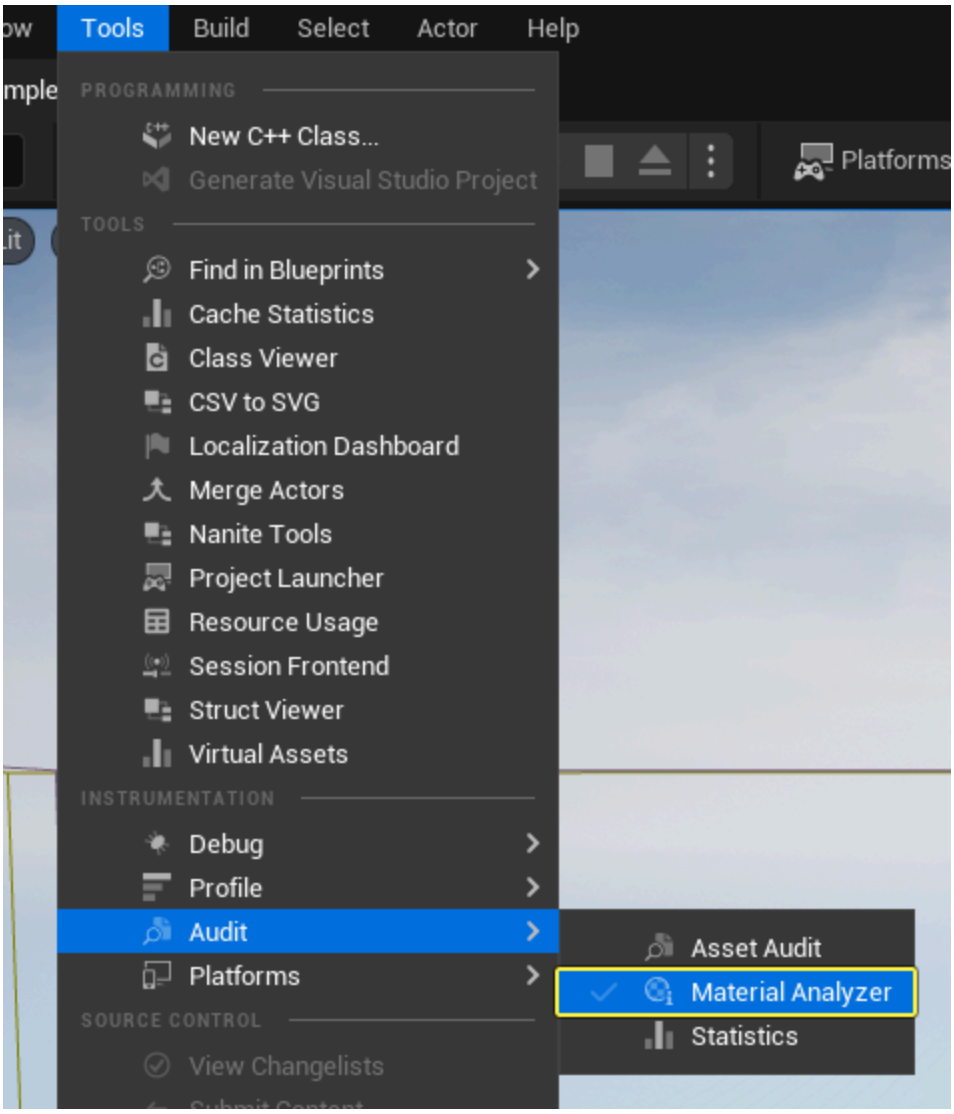
This page describes how to locate and use the Material Analyzer tool.



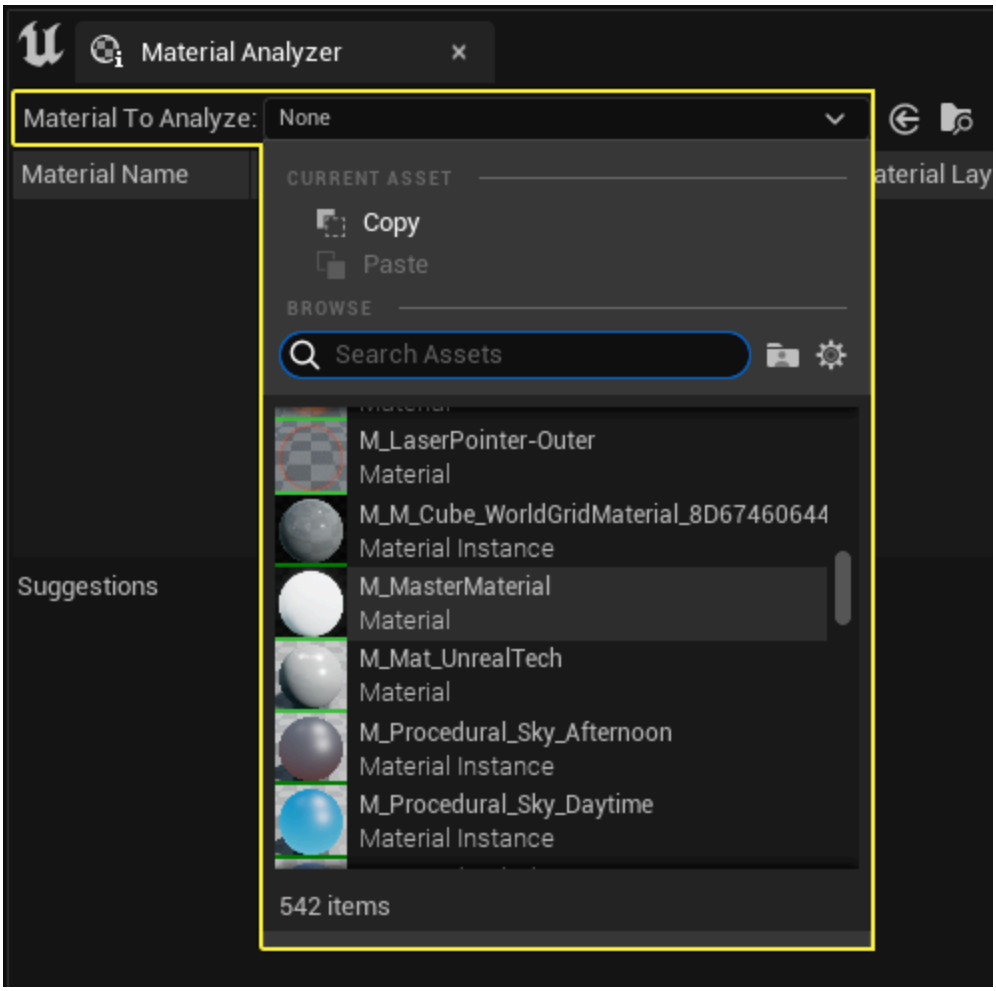
The **Material Analyzer** is a developer tool that helps you identify and analyze all Materials, or Material instances, in your project. This enables you to make changes that can provide savings in shader permutations and data storage. When you select a Material or Material instance to analyze, the tool will find all descendants of that Material (or all descendants of the Material instance's parent Material). The tool also identifies all of the base property overrides, static switches, and static component mask parameters.

Opening the Material Analyzer

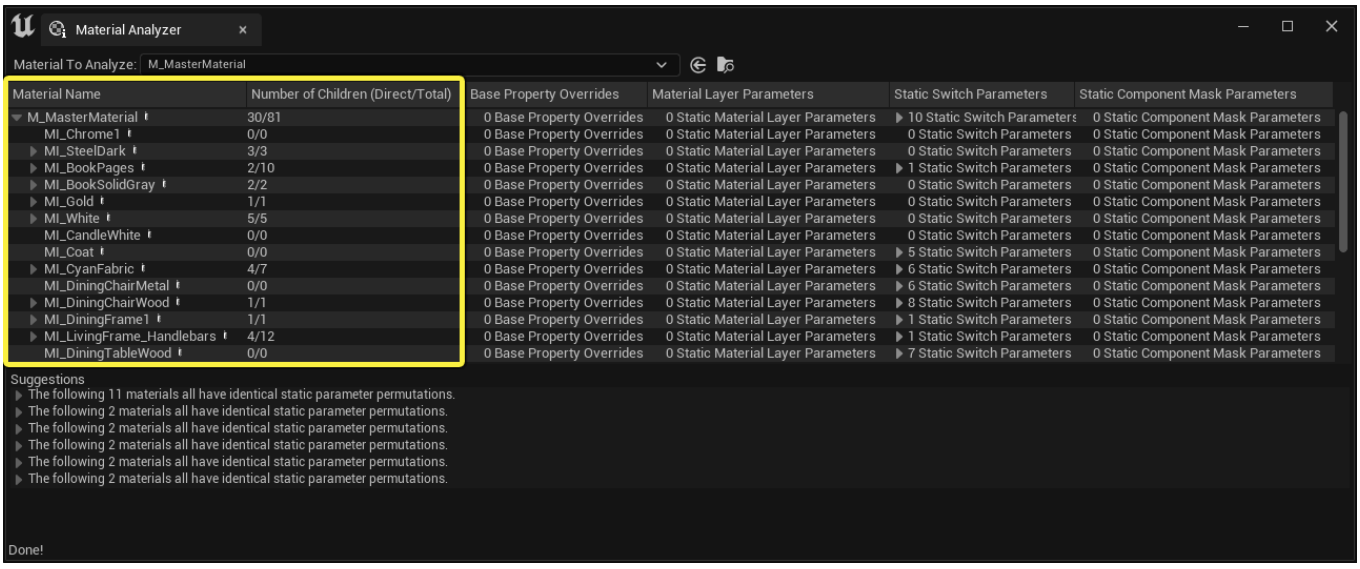
1. In the menu bar, click **Tools > Audit > Material Analyzer**. The **Material Analyzer** window opens.



2. Click the dropdown menu next to **Material to Analyze**. Use the list or search bar to select the Material or Material instance you want to analyze.

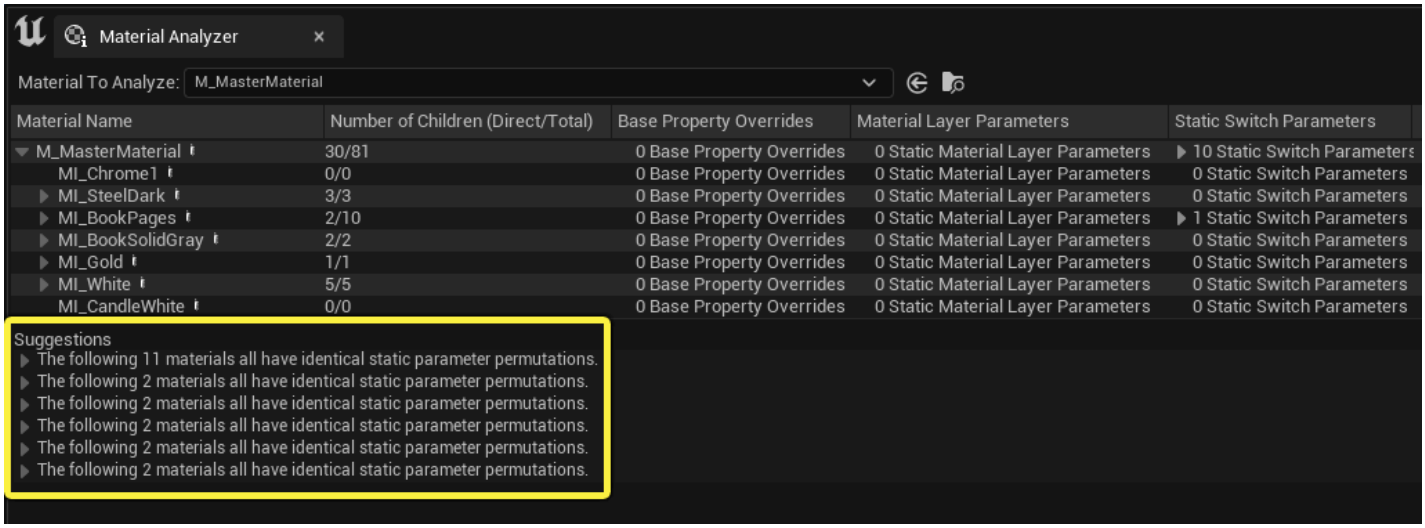


3. The Material Analyzer tool displays a list of all instances of the Material you selected.



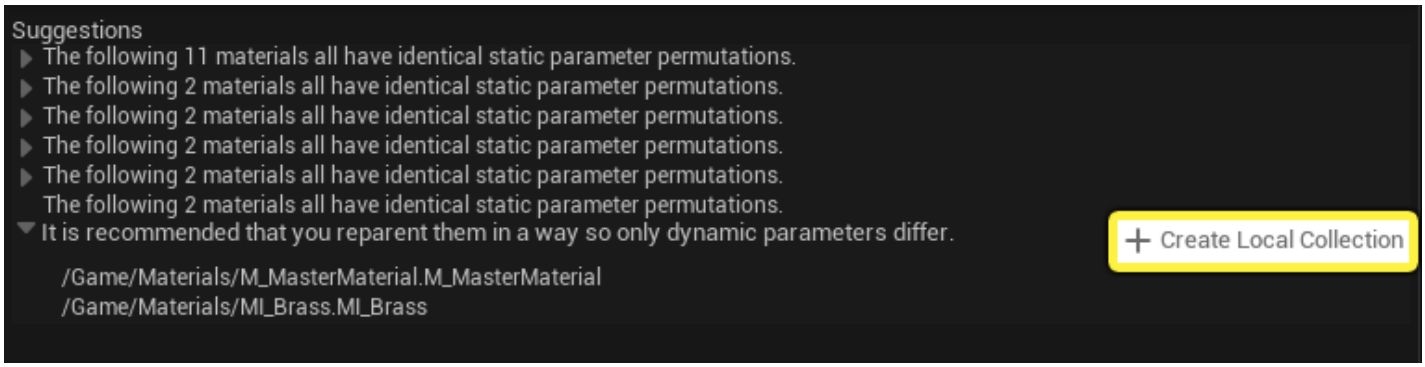
Viewing the Suggestion List

Below the Material instance hierarchy is a suggestion list. The suggestion list groups all Material instances with the same set of static overrides. You can click the arrow next to each line to see the specific instances identified.



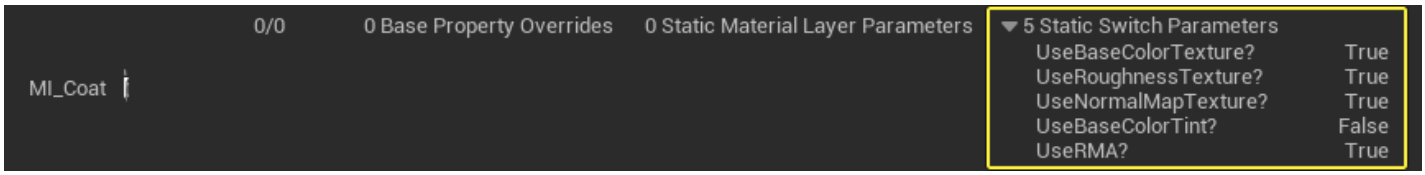
Create a Local Collection

Each suggestion list has a **Create Local Collection** button. Click this button to place all the related instances into a local collection, so you can easily find them and update them to have more efficient parameter setups.



Viewing the Static Switch Parameters List

To see the static switch parameters for a Material instance, click the arrow next to the Static Switch Parameters column to display the full list. The columns are resizable, so you can move them if the text is cut off.



Reparenting Material Instances

You can reparent these Material instances to a new instance that has those same static overrides, so that the reparented Material instances only change their unique overrides. This can provide a savings in shader permutations and data storage. Make sure to remove all the static parameter overrides from the Material instances you have reparented, or else the additional data is still stored.