

# Basic Mesh Combiner

## Index

Intro .....	1
Merge Types Info .....	1
Submesh per Material .....	1
Visual Merge Effect .....	2
Submesh Per Material .....	2
UI Panels .....	3
Merge Panel .....	3
One_Sub Mesh_per_material .....	3
User manual .....	6
Common practice .....	6
Cannot combine mesh does not allow access: MeshName .....	7
Why did the Verts/Tris count increase in my scene after merging? .....	7
SetPerIndexUVs failed because the output has more then 64K vertices .....	8
Mesh has null in Mesh Filter .....	8
Mesh Renderer Component missing .....	8
Mesh Filter Component missing .....	9
Demo Scene .....	9

## Intro

Basic Mesh Combiner represents an Basic iteration of the Advanced Mesh Combiner tool, offering a Small range of Basic features and comprehensive support. This powerful tool facilitates a seamless and efficient workflow, enabling users to effortlessly merge meshes. Furthermore, the customizable preferences provided further optimize the workflow, ensuring a professional and streamlined experience.

## Merge Types Info

When using the Basic Mesh Combiner, there is a single method available for merging. If you require additional support and more merging methods, I would recommend purchasing the Advanced Mesh Combiner. For further assistance, please don't hesitate to contact me at [detoolsassetstore@gmail.com](mailto:detoolsassetstore@gmail.com) with your suggestions.

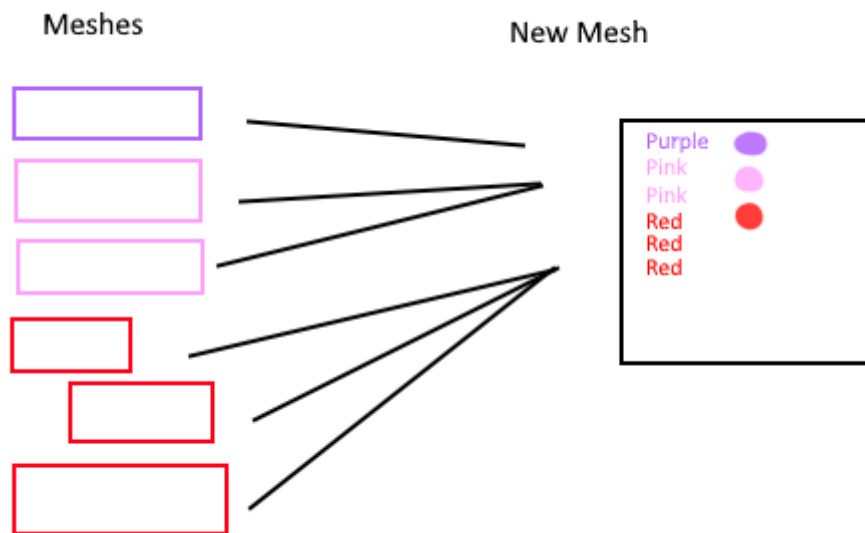
### Submesh per Material

This represents the fundamental merging method commonly employed in projects. The Merge operation scans each selected object listed under "Selected game objects" to identify their respective

materials and combines them into submeshes, thereby reducing the draw call rate. While this method is generally reliable, there may be occasional instances where a UV error is present within the merged result.

## Visual Merge Effect

Submesh Per Material



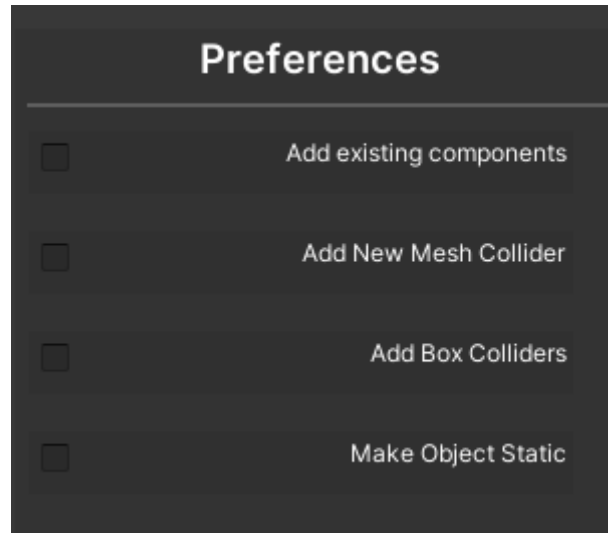
## UI Panels

In this section, there will be a sequential presentation of panels, each of which will be explained in detail.

### Merge Panel

One\_Sub Mesh\_per\_material

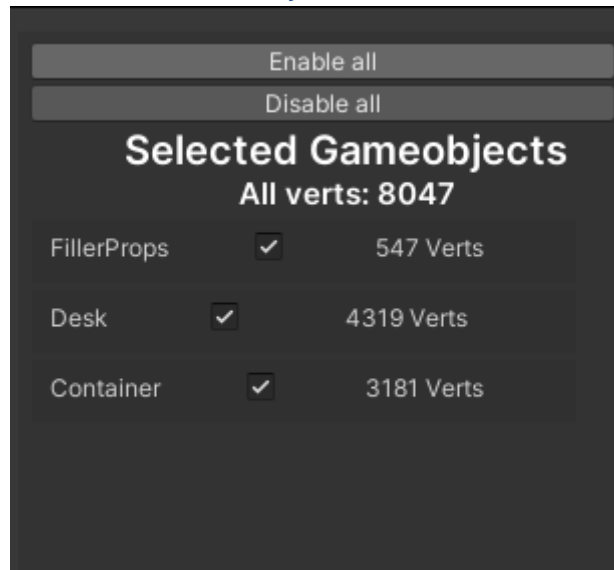
#### *Preferences*



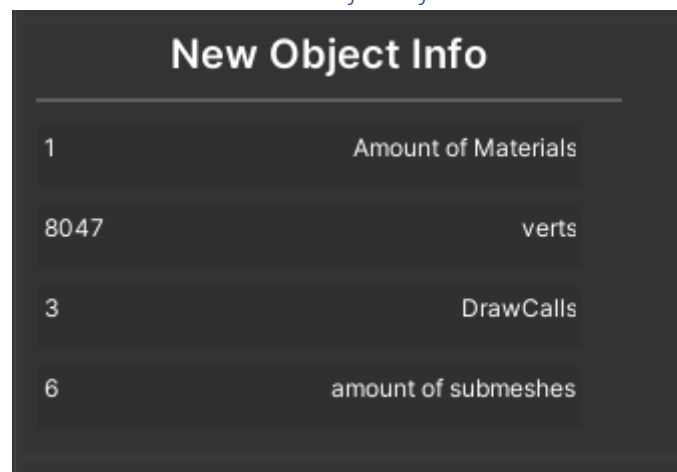
The "Preferences" box, located in the top left corner of the main panel, contains various user preferences and features, including:

1. "Add existing components": This feature converts all components from the selected game object to the new game object.
2. "Add New Mesh Collider": Enabling this feature adds a mesh collider to the new mesh.
3. "Add Box Colliders": Enabling this feature adds box colliders around all the submeshes.
4. "Make object static": Enabling this feature sets the new game object as static.

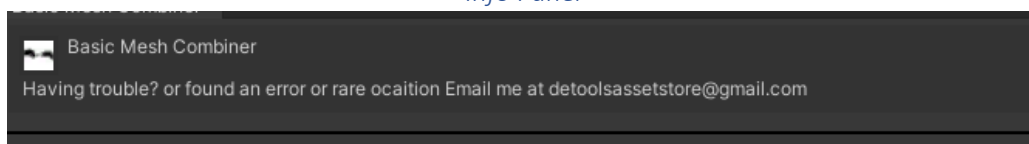
If you find any preferences missing or require more advanced options, I recommend checking out the Advanced Mesh Combiner tool for additional features and customization.

*Object List*

To select game objects, navigate to the top right corner under the "Selected Gameobject" tab. Objects that are disabled will not be included in the merge, allowing you to exclude specific items. This feature also helps in identifying any missing or unintended objects.

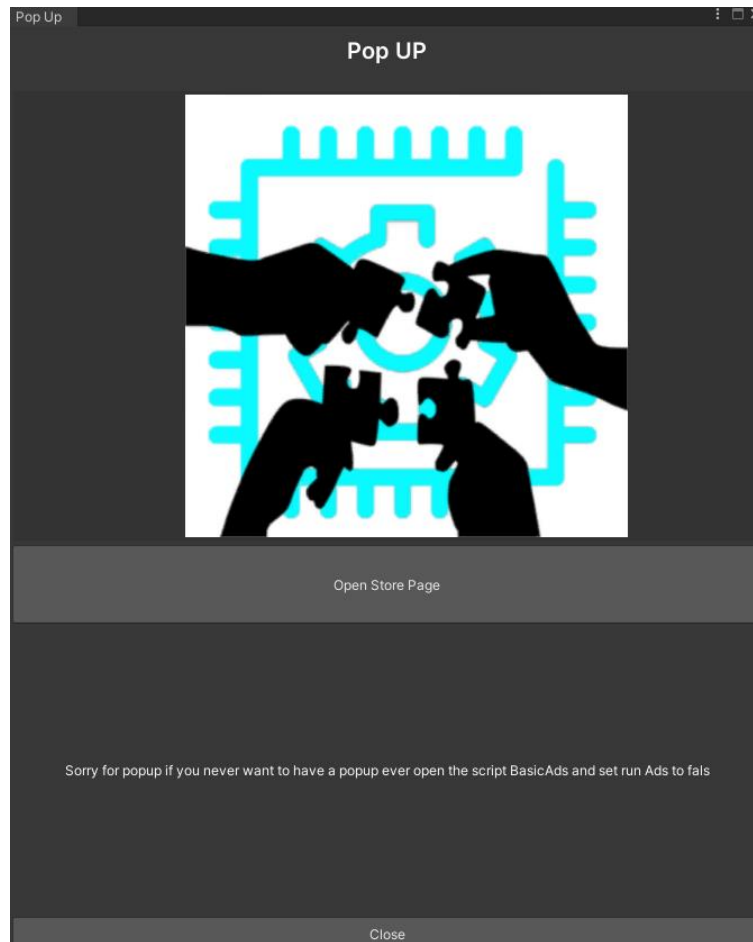
*New Object Info*

Below, you will find the section titled "New Object Info," where the updated information regarding the object's statistics will be displayed.

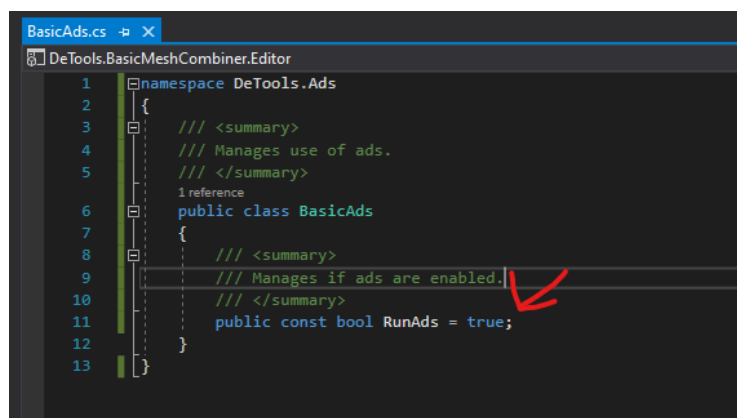
*Info Panel*

The info panel is an small panel that shows how to reach support.

### Advertisement



Due to marketing reasons, I have implemented a 1/5 chance for a small advertisement popup window to appear. This popup showcases marketing ads for DeTools. However, please note that this feature can be disabled. Please refer to the text and explanation below for further details.



This is a brief tutorial on how to disable advertisements in the Basic Mesh Combiner:

1. Open the BasicAds script.
2. Locate the "RunAds" variable.
3. Set the value of "RunAds" to false.
4. Save the script.

By following these steps, you will effectively disable the advertisements in the Basic Mesh Combiner.

## User manual

First, you open the tool by navigating to the "Tools" tab at the top. Then, under "DeTools," select "Basic Mesh Combiner" to open the tool.

The first step in merging your game objects is selecting and filtering the specific game objects that you wish to merge.

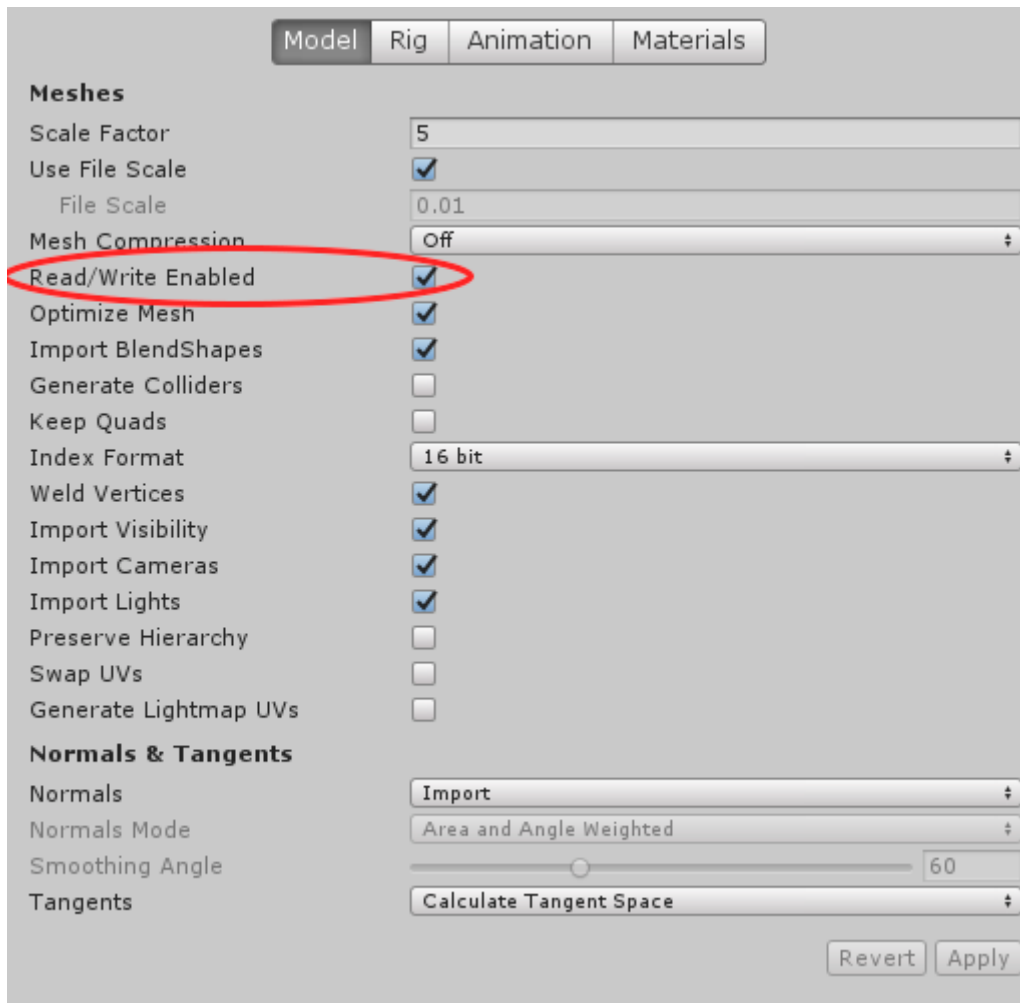
Once you have set everything up, simply press the "Combine" button, and a new mesh will be generated as a result of the merge.

## Common practice

To ensure a user-friendly and secure experience, I have included a "Common Practices" tab. This tab provides valuable information for users, highlighting potential issues and explaining how to avoid them. By familiarizing yourself with these common practices, you can enhance the safety and ease of your usage.

### Cannot combine mesh does not allow access: MeshName

When encountering this error, a common practice is to enable the read/write permission on the asset. You can refer to the image bellow.



### Why did the Verts/Tris count increase in my scene after merging?

In certain cases, when you combine meshes in your scene, you may observe a slight increase in the vertex count compared to individual uncombined meshes. This is a normal behavior in Unity, as the entire mesh is rendered even if only a portion of it is visible. Consequently, the viewed vertex count increases as Unity renders the entire mesh.

Typically, this is not a cause for concern. However, if you notice a significant increase in vertices after merging, it could indicate that the resulting merged mesh is exceptionally large. In such cases, even if the player's view is limited to a smaller section of the combined mesh, Unity still renders the entire mesh. To address this, you can consider combining meshes in smaller groups rather than merging a large area of your scene into a single mesh.

For example, if your scene spans 1000 meters, try grouping meshes in sets of 100 or 200, with each group containing the closest meshes. By adopting this approach, when the camera focuses on a particular area, Unity will only render the specific group of meshes relevant to the visible portion. This optimization strategy can help manage performance by reducing the rendering load on large combined meshes.

### SetPerIndexUVs failed because the output has more then 64K vertices

this error occurs when you enable Lightmap support during the merge process. Let's consider an example: if a merged mesh, without lightmap support, consists of 5,000 vertices, enabling lightmap support will cause the mesh to have 10,000 vertices. However, if the mesh exceeds 64,000 vertices during the conversion to lightmap support, Unity will cancel the lightmap support for that specific mesh. It's important to note that this limitation is inherent to Unity.

To overcome this limitation, there are two recommended approaches:

1. Reduce the vertex count: To mitigate the issue, you can try reducing the number of vertices in the objects intended for merging. This can be achieved through optimization techniques such as simplification algorithms or LOD (Level of Detail) systems.
2. Merge fewer meshes at once: Instead of merging a large number of meshes in a single operation, consider splitting the merging process into multiple stages. By creating more parts of the merged mesh and merging fewer meshes at once, you can effectively manage the vertex count and work within Unity's limitations.

By implementing these strategies, you can navigate around the vertex limit imposed by Unity's lightmap support and achieve successful merges without encountering this error.

### Mesh has null in Mesh Filter

This issue arises when the Mesh Filter component is not associated with a mesh. To resolve this problem, you have two options:

1. Associate a mesh: Ensure that you have a mesh available and properly assigned to the Mesh Filter component. You can select a suitable mesh from your project assets or create a new one. Associate the mesh with the Mesh Filter component to resolve the issue.
2. De-select the GameObject: If you don't intend to use a mesh for the GameObject, you can simply de-select it. This means removing the Mesh Filter component altogether, as it won't have any purpose without a mesh. By de-selecting the GameObject, you can eliminate the problem caused by the missing mesh association.

By either assigning a mesh to the Mesh Filter component or removing it by de-selecting the GameObject, you can address this problem and ensure the correct functionality of your scene.

### Mesh Renderer Component missing

This issue occurs when a GameObject in your scene is missing the Mesh Renderer component. To identify and fix this problem, you can utilize the Combiner Tool, which offers a convenient solution:

1. Utilize the Combiner Tool: Open the Combiner Tool and initiate the detection process. The tool will scan your scene and identify any GameObjects that do not have the Mesh Renderer component.
2. Identify the faulty GameObject: The Combiner Tool will provide information about the specific GameObject that is missing the Mesh Renderer component. Take note of this GameObject for further action.



3. Correct the issue: Once you have identified the faulty GameObject, you can resolve the problem by adding the Mesh Renderer component to it. The Mesh Renderer is responsible for rendering the mesh of the GameObject and is essential for proper visualization.

By utilizing the detection capabilities of the Combiner Tool, you can quickly pinpoint GameObjects without the necessary Mesh Renderer component and take the appropriate steps to fix the issue. This ensures that your scene functions correctly and displays the meshes as intended.

#### Mesh Filter Component missing

This issue arises when a GameObject in your scene does not have the Mesh Filter component. To identify and resolve this problem, you can utilize the Combiner Tool, which offers a convenient solution:

1. Use the Combiner Tool: Open the Combiner Tool and initiate the detection process. The tool will analyze the scene and identify any GameObjects lacking the Mesh Filter component.
2. Locate the faulty GameObject: The Combiner Tool will provide information about which GameObject is missing the Mesh Filter component. Take note of this GameObject for further action.
3. Correct the issue: Once you have identified the faulty GameObject, you can rectify the problem by adding the Mesh Filter component to it. This component is crucial for proper rendering and mesh-related operations.

By utilizing the Combiner Tool's detection capabilities, you can quickly identify GameObjects lacking the Mesh Filter component and take the necessary steps to resolve the issue, ensuring the proper functioning of your scene.

#### Demo Scene

Inside the demo scene, you will find several merge examples and basic assets that you can use for testing purposes.