

Terrain Slicing Guide

Author: Kyle Gillen – Visit www.deepspacelabs.net/terrain_slicing.html to find full documentation

Contact me at kgillen@deepspacelabs.net if you have any questions or concerns

4.x.x Users

To add some functionality, please perform the following two steps (unless you've already performed them).

1) Delete the SliceTerrain.js file found in the editor folder. This file is for 3.x users and does not contain some functionality available to 4.x users.

2) Open the SliceTerrain.txt file and resave it as SliceTerrain.js (so Unity recognizes it as a JavaScript file). This file contains code for copying terrain materials and terrain texture normal maps.

How it Works

The script works by taking a base terrain, and dividing its properties by a certain value, which we'll refer to as size. In the editor window you'll actually choose the dimension of the resulting terrain grid after slicing is completed, which is simply size x size. So for example, if you choose 2x2 as the grid dimensions, the script will take your terrain's width (measured along the x axis), length (measured along the z axis), detail resolution, control texture resolution (also known as the alpha map resolution), heightmap resolution, and base texture resolution, and divide each of these values by 2 (actually for the heightmap, you would subtract 1, then divide by 2, and then add 1 back to get the new value -- or simply divide by two and round down).

The new terrains are created on the fly, and their terrain data is stored in the Asset Database (in the TerrainSlicing > TerrainData by default, though this can be changed). Because the terrain data is saved in the asset database, multiple terrains can be added to an empty game object and saved as a prefab, or loaded dynamically later. All trees and detail meshes are also copied to the new terrains (though the detail meshes will not be 100% the same)

Before Slicing

Before slicing, it's important to make sure the base terrain you're intending to slice is able to do what you want. If you haven't created your base terrain yet, the process is easy. Simply think about the properties you wish the new terrain slices to have, and create the base terrain to make these properties possible; for instance, if you want 4 terrains, each with a size of 200 x 300 and all resolutions at 1024 (with heightmap at 1025), you'll need to calculate the base terrain accordingly. First, determine the size you'll be working with by taking the square root of the number of terrains you want to end up with; in this case, the square root of 4 is 2. Now simply multiply all of your desired values by this size. You should find that you'll need a base terrain with a size of 400 x 600 and all resolutions at 2048 (with heightmap at 2049).

If your terrain is already created, then don't worry, you shouldn't have much trouble; you simply might be limited in the number of terrains you can create. The reason for this is there's a minimum limit for each resolution (16 for base texture and control texture resolution, 33 for the heightmap, and the detail resolution per patch for the detail resolution if you plan on having detail meshes -- 0 if you don't).

Because of these minimum limits, your base terrain resolutions divided by size cannot be smaller than these values. Most of the time you'll probably be working with a base terrain of resolution 512 or greater, so the only time you'd have to worry about these constraints is if you're aiming for a 16 x 16 or 64 x 64 grid (not tested or recommended -- that's a lot of terrains!). Still, it's something to keep in mind, especially if our base terrain has smaller resolutions.

Terrain Naming

The resulting terrain slices are named based on the base terrain's name, as are the terrain data. Terrain slices take the form of BaseTerrainName_Slice_row_column, while terrain data follows the format BaseTerrainName_Data_row_column. A row consists of all terrains that share an x value for position, while a column consists of all terrains that share a z value for position.

For example, if the base terrain is named "Terrain", and four slices are created, then the first terrain (situated at bottom left – aka x=0, z=0 if the base terrain is at 0, 0), will be named Terrain_Slice_1_1. The second (which is in the next column but same row – aka just to the right) will be named Terrain_Slice_1_2, the third Terrain_Slice_2_1, and finally Terrain_Slice_2_2 for the last slice.

Warning!! Because of this naming convention, if you attempt to slice a terrain that shares a name with a terrain you've already sliced, the resulting TerrainData will overwrite the existing TerrainData from the previous terrain. By default the program will tell you when you try to overwrite existing data and keep you from doing it. You'll need to select the checkbox for "Overwrite Terrain Data" in the Slicing Window in order to actually overwrite data. If you wish to avoid overwriting

Folder Structure

It is perfectly fine to move the files around, but there are two things to keep in mind. First, the SliceTerrain.js and SetNeighborsEditor.js (see Neighbors Guide for more info) must be placed in a folder named "Editor" in order to work. Second, in order to change the location where the terrain data is created and stored, you must enter the file path in the appropriate text field on the Slice Terrain Window (more on that later).

The SetNeighbors.js file can be placed anywhere.

The Slicing Process

- 1) To begin the slicing process, select the Terrain option at the top of the Unity Editor, and then click the "Slice Terrain" option. A window will appear with some configuration information.
- 2) Drag the terrain you wish to slice into the "Terrain to Slice" field. Alternatively, if you had the terrain selected when you clicked the "Slice Terrain" option in step 1, the terrain will already appear in this field.
- 3) Enter the Detail Resolution Per Patch you desire. Preferably, this value should match the base terrain's detail resolution per patch value. This information cannot be accessed via scripting, which is why you must enter it here. You can enter a different value than what the base terrain is set to, however this will result in less accuracy in how well the detail meshes (plants and grass) will be copied over (note: some inaccuracy will occur no matter what, due to how detail meshes are painted and stored – more on that later).
- 4) Select the slicing dimensions you wish to end up with. 2 x 2 simply means the base terrain will be divided along the x axis 2 times and z axis 2 times, to create 4 terrain slices. 64 x 64 is untested and not recommended, so use at your own risk.
- 5) Set the file path where you wish to store the terrain data. By default this is Assets/TerrainSlicing/TerrainData. If you wish to temporarily create terrain data in a different folder, simply enter the new path here. If you wish to permanently change the default folder, enter the new file path and select the "Save Current File Path as Default File Path" button. Please ensure there is no "/" after the last folder name in the file path (for example, after TerrainData for the default file path), or else an error will occur.

- 6) Select whether to overwrite existing terrain data when you click the “Create Terrain” button. This is a security feature to ensure you don’t accidentally overwrite terrain data you’ve already created. If you attempt to overwrite data while this value is unchecked, a warning message will appear telling you to check this value, and the slicing operation will not begin.
- 7) Click the “Create Terrain” button and wait for the progress bar to fill. If the progress bar doesn’t show, a notification message alerting you to an error will show in the editor window. Occasionally you may need to check the console for more detailed information. Finally, ensure you only try to execute the slice terrain script while in edit mode.