MicroSplat F.A.Q.

These are the most common questions I get, just about all of which are covered by reading the included documentation.

[Pre-Purchase Questions]
[First Use Questions]
[Feature Questions]
[Common Issues]
[Other Questions]
[Common Unity Warnings]
[Advanced Topics]

Pre-purchase questions:

Q: **“Does the terrain collection include module X”**
A: The terrain collection description on the asset store lists exactly which modules it includes. If it is not in that list, it is not included.

Q: **“Will module X be added to the terrain collection?”**
A: No

Q: **“Does bundle X include module Y?”**
A: This is all clearly listed on the asset description.

Q: **“Do I need to buy the HDRP support module to make MicroSplat support HDRP?**
A: Uh, yeah.

Q: **“Do I need to buy the URP support module to make MicroSplat support URP?”**
A: Read the last question, then take a guess.

Q: **“Why does the X2020 adapter require an upgrade fee?”**
A: Because unity doesn’t provide any unity in their shader system and changes it on a regular basis in a way that breaks every shader in existence requiring extensive rewriting of the shader. Shaders are not only not compatible between render pipelines, they are often not compatible between different versions of the same render pipelines as well. For instance, the change from HDRP7.x to HDRP10.x required a full rewrite of about 10k lines of shader code, of which Unity provides no documentation and incomplete notes about what has changed. The change from URP7.18 to URP7.21 broke shadow mapping for URP, and was included in a point release of an LTS cycle, breaking almost every shader. Supporting the render pipelines took about 60% of my

development time in 2019 and 2020, and is responsible for the largest percentage of support requests as well. I should be charging far more for the work.

Q: **“When will you ship the 202x.3LTS version of the adapter?”**
A: Unity has to actually release the 202x.3LTS version before I can start to work on it, so sometime after that.

Q: **“Does this work in some tech/beta/alpha release of an SRP?”**
A: No idea, and if it doesn’t, don’t bother telling me about it, because I only support LTS versions of SRPs.

Q: **“Does module X do Y?”**
A: The documentation for every module is included in the free Core module, so you can look through how everything works before purchase

Q: **Does module X work with pipeline Y?**
A: sans any active bugs, all modules work in all render pipelines, or are noted as such in the render pipeline asset description.

Q: **Can I get a discount because of some flimsy reasons I just made up?**
A: No

Q: **Bundle X is on sale, can I refund the assets I’ve been using for the last year to buy the bundle instead?**
A: No

Q: **But the assets store says “Refunds, we’ve got you covered”**
A: They lie

Q: **If I buy your asset, you’ll code my game for me right?**
A: No, you are buying software, nothing more. I’m not even required to provide support if I don’t want to. Quite a few people have been tossed out of my support channels for acting entitled, so realize that the thousands of hours I spend helping people out in my discord and forum posts are me donating my time, not something I owe anyone.

Q: **Will you add custom feature X for my game? Do you want to work for a credit in my game, it’s sure to be huge!**
A: I’m a professional graphics engineer with 28 years of experience, started multiple successful companies, have many hit games, and well over 50 million units sold. My time is extremely valuable, to both me, and to my clients. I do custom work for people all the time, but I’m not cheap.

First use Questions:

Q: **“How do I start?”**
A: Preferably by reading the documentation I spent so much time writing, or watching one of the many videos I provide on YouTube. Each module has a video showing off it's features, and the documentation has a walk through of setting up those features and explaining what each property does. For instance, if you go to Core/Documentation and read the MicroSplat Documentation pdf file, it will walk you through setting up a terrain, same as the YouTube video linked in the asset description does. Further, all documentation for the modules is included in core, so you can read up on exactly how any module works before purchasing.

Q: **“Something looks ever so slightly different, will you re-record a new version of the YouTube video?”**
A: No, videos take a lot of time to produce, and I’m not going to update them unless a large change is made making them no longer valid. The documentation is always up to date, because editing text is a lot easier than re-recording hours of video.

Q: **“I keep installing the software but it’s not there”**
A: It’s installed as a package, so it’s not in the Assets folder

Q: **“The samples won’t install”**
A: The package manager interface has buttons you can use to install samples into your project. Make sure it’s set to view the packages in your project, not in your asset library. They are not installed by default to save significant install time, and to not clutter up your project.

Q: **“I installed MicroSplat and the URP/HDRP module on Unity 2026 alpha and the shader is pink!”**
A: You need to set the render loop on your shader to the appropriate render pipeline; this is covered in the documentation for the render loop adapters. Also, read the supported versions for the SRP modules, as Unity often breaks compatibility in SRP pipelines and only LTS versions can be reasonably supported.

Q: **“The textures are too large/small”**
A: UV scaling is available as both a global under the “Splats” section of the material, or in the per-texture properties section at the bottom of the shader. When you convert a terrain, if it has existing textures with very different scales, it will turn on per-texture UV scale, set the global scale to 1, and compute the scale for each texture as part of the conversion.

Q: “The textures are blurry!”
A: See above

Q: “I added a new texture and it’s blurry!”
A: See above

Q: “I added a new texture and it shows up black”
A: Make sure your Max Texture count on the material is higher than the number of textures you have in your Texture Array Config

Q: **“I keep editing the terrain layers but they just get ignored”**
A: As stated in the documentation, once you convert to MicroSplat you manage all textures from the Texture Array Config. This is because Unity’s layer system is not extensible, and does not have a way to store all the types of textures MicroSplat supports.

Q: **“How do I add a texture?”**
A: See above

Q: **“I added a texture but it won’t let me paint it”**
A: See above

Q: **My texture is shiny**
A: This is controlled by the smoothness texture provided in the Texture Array config, if you did not provide one, smoothness values are automatically generated. Other 3d applications often use roughness maps, which are just inverted roughness maps, which can be automatically inverted by clicking on invert in the texture array config on the smoothness texture. There is also a per-texture property to adjust the smoothness of each texture. Finally, read the core documentation on the Unity lighting model and alternate lighting models provided if you are running the standard renderer. Note that a gloss map is not the same as a smoothness map, and is a remnant from the non-PBR rendering days.

Q: **“The blending looks different than the Unity Terrain”**
A: By default, MicroSplat uses a height map based blend for terrain blends. If you prefer blurry Unity style blends, you can disable height map based blending in the core features section, or adjust the amount of height blending with the interpolation contrast slider.

Q: **“I want multiple terrains to share the same texture arrays and shader and such”**
A: Simply select all the terrains when doing the conversion, and one MicroSplatData directory will be created for all of them. If one terrain has been converted already, simply add the MicroSplatTerrain component to the new terrain, set the template material to the one in your MicroSplatData directory, and press sync.

Q: **“Can I rename things willy nilly in the MicroSplatData directory?”**
A: Sure, but you might break things. Best to not rename this directory or anything in it- that said, you can move the entire folder to where ever you want it.

Q: **“The interface looks every so slightly different in this YouTube video, will you go re-record all 30 hours of YouTube videos so it matches exactly every week when you make changes?”**
A: No

Q: **“I don’t have height/normal/smoothness/etc maps for the textures and it says MicroSplat will create them, but I don’t see them being created”**
A: They don’t get written onto disk or show up in the interface. Rather, they are generated from the best data you provide and stored directly in the resulting texture arrays. For instance, if you only provide a diffuse map, it will generate the height map from the luminosity of the diffuse map. However, if you have a normal map, it will generate the height from the normal map. It’s better to provide these textures yourself. If you don’t want any smoothness, for instance, you’d have to provide a black texture in the smoothness slot.

Q: “I removed MicroSplat and it destroyed all my work!!!”
A: No, it didn’t. After removing MicroSplat from a terrain you may have to (depending on Unity version)

- Reset the material on the terrain back to the Unity default terrain material or any other you want to use
- Reset base map distance to something larger (say 200 or more, MicroSplat often sets this to 0)
- If you delete your MicroSplatData directory, any terrain layers created in there will be deleted with it, causing missing layers on the terrain. When you add new texture via the MicroSplatConfig, it will automatically generate a terrain layers for this texture in the MicroSplatData directory. Preferably don’t delete these files.

Q: “I’m using map magic and having issue X”
A: I did not write the map magic integration, but have noticed it often doesn’t setup all the data needed on the terrain component, and that is likely what’s causing the issue.

Feature Questions:

Q: **“I turned on Triplanar and now my textures are gone”**
A: No, they are just in a very different scale. When not set to triplanar, the default is to scale them based on the UVs, which are 0-1 across the whole terrain. So a UV scale of 500 makes th texture scale across the terrain 500 times. But when it triplanar, they are based on meters. A value of 500 would tile the texture 500 times over a 10 meter area.

Q: “**When I turn on tessellation, my characters feet go through the terrain.”**
A: This is because the physics engine cannot collide with the tessellated surface, as that is happening on the GPU. The best workaround is to use the displacement offset and center each of the textures- this is covered in the tessellation documentation. Keeping displacement distances reasonable, offset from center, and possibly combining with parallax, can greatly enhance the depth effect without offsetting the surface too far from the original positions.

Q: **“How do I set value X on the material?”**
A: You can grab the materialTemplate from the MicroSplat* component and set it through the standard material.SetFloat style interface. Afterwards, call Sync on the component, or MicroSplatObject.SyncAll if more than one thing is using the same template.

Q: **“How do I set a per-texture property via scripting?”**
A: There is an API on the PropData object (in the MicroSplatData directory and assigned to your MicroSplat terrain/mesh component) which will allow you to set any value you want, after which you have to call a sync function (MicroSplat.SyncAll(), or Sync() on the terrain or MicroSplatObject.

Q: I turned on terrain blending and it stretches on the side of my object
A: That’s because it’s doing exactly what the terrain is doing- texturing based on a top down projection. If you want the sides to not stretch, you’ll need to turn on triplanar texturing.

Common Issues

Q: My per texture properties aren’t working
A: Most likely you renamed something in the MicroSplatData directory. Materials are not able to point to objects, so the material looks up the PropData object based on the materials name, and creates one if it’s not found. So if you rename things inside the MicroSplatData directory, an extra PropData object might be created. The terrain might still point to the old one, while the material finds the one with the correct name.

Q: I’m seeing some white shine in the distance
A: Unity has a quality settings which allows you to reduce the resolution of your textures. However, when this is used, the API which returns the number of mip maps returns one less mip map than it should, which causes the texture array to leave the last mip blank (white). To fix, set your quality settings to use full size textures and repack the arrays by selecting the TextureArrayConfig and pressing “Update”

Q: I Updated to the latest, but I still have issue X
A: The package manager is broken in many, many versions of Unity, from 2019 to 2021. It will show that you have the latest version of the package, but instead install an old version from it’s

cache. Thus, the only way to get it to actually update packages from the asset store is to delete the package manager's cache folder before downloading an update.

Q: **I just installed/updated MicroSplat and am getting compile errors**
A: See above, but also realize that when you update some MicroSplat module or purchase a new one, it's most likely that other modules have changed since you last downloaded them, and you need to update them as well.

Q: When I compress into ETC2/ATSC/PVR format, etc, I see errors about the format not being supported
A: When unity compresses a texture array, it then has the editor load it. When you compress into a format like ETC2, for say, an android device, it then loads the texture in the editor and says "Hey, windows doesn't support this format" and throws an error. It then uncompressed the data and uses it on windows as uncompressed data. Normally this wouldn't happen because the actual platform specific textures are stored in the library folder, but in the case of texture arrays they are in the project themselves- in this case as an ETC2 texture, which causes the editor to complain.

Q: **"My textures don't tile and I see cracks at the edges of the terrain with tessellation"**
A: Use world space UVs

Q: "I'm running with addressables/asset bundles on android and the terrain draws black"
A: Unity has a bug, which they have marked "Won't fix", where if you don't include a terrain in the main build, it strips some of the things needed for terrain instancing. You can either run without draw instancing, or include a terrain in the main build.

Other Questions:

Q: **"I turned on a per texture property and it's not working"**
A: 99% of the time it's because something in the MicroSplatData directory has been renamed. The material editor, for instance, looks up the object which holds the per-texture properties based on the material name - it has no way to know which objects may be using it, so it can't find it from the terrain or mesh. However, if you rename the material, then it will create a new PropData object to hold these properties, but the terrain or mesh will still be using the old one. Unless you really know what you are doing, don't rename things in the MicroSplatData directory.

Q: **"Something isn't working!?!"**
A: The basic process I will put you through will look something like this:

- Do you know about the package manager bug?

- Are you in an SRP? What version?
- Which Unity version are you running?
- MicroSplat version? (it's on the material editor)
- Windows or OSX? Runtime or editor time? What runtime target?
- Do you have compile errors in your project?
- Did you try the included example? Does it work there?
- Did you try in a clean install, instead of your project with 7000 other assets stuffed into it?
- Did you follow the docs when setting it up?
- Did you try toggling off and on certain features to isolate the issue?

If you have not done these things, or included this information, you can save us both a lot of trouble by doing these things first, and often end up solving your issue in the process. Often, people will post issues with little to no context, a screenshot saying "What's that!?" and I'm like "I have no idea, I have no context for what it's supposed to look like". In general, support requests that have more information and context get more of my attention than ones which a speck of thought hasn't been put into, and are therefore more likely to be simple user errors.

Q: **"I get an error about too many samplers"**
A: Most platforms have a maximum number of samplers, 16 or 32. MicroSplat tries to share samplers whenever possible, but sometimes a sampler is needed and cannot be shared- if these get over 16, Unity will not compile the shader, and you will have to disable an option to reduce sampler count. On some platforms, you can increase the number of available samplers by increasing the shader model in the core/advanced section. Note that Unity internally uses some of its own samplers for lighting and other effects, which contribute to this limit.

Q: **"When I turn on feature X, the shader does not compile, I get an error in the console"**
A: Please send me this error, along with a screenshot of your material (the top section, but preferably the whole thing), or alternatively the keywords file. Unity version and which SRP and version you are using as well.

Q: **"How do I read what texture the player is on?"**
A: Same as you would with Unity's shader, unless you are using custom splat maps or procedural texturing. For custom splat maps you'll have to read the data yourself. For PT, there are utility functions that allow you to do this.

Common Unity Warnings

Q: **"I get binary to Yaml conversion errors when I install"**
A: These are a Unity bug, supposedly fixed in an upcoming release. They cause no issues.

Q: "I get an error about duplicate _MainTex"

A: This has recently been fixed by Unity and is being back ported into latest patch releases. It causes no issue.

Q: **“I see a warning on the terrain component about the shader needing tangents”**
A: This is an erroneous warning and can be ignored.

Advanced Topics

Q: “I want to change individual textures at runtime, as this is a user generated content game”
A: The easiest way to do this is to put each texture the user can select into its own texture array to pack it into the correct format, then use Graphics.CopyTexture to copy the texture into the target array. However, if you are downloading the textures as png’s or something like that, then you will have to pack them into the correct format yourself. The code in TextureArrayConfigEditor.cs is what you want to look through to understand this.

Q: How do I find out what texture I’m on?
A: For Unity terrains, it’s the same way as you would a standard Unity terrain, using the terrain API. For meshes or terrains using external splat maps, you’d get the UV coordinate from the raycast, then you’d have to query the splat map data (usually a list of textures) and find the one with the highest weight.

Q: I want to turn on/off feature X at runtime to provide various detail levels for the user
A: Shaders cannot be compiled at runtime, so you must pregenerate a shader with the options you want and swap the shader on the material at runtime. The usual way to do this is to swap the shader on the template material and then call Sync or MicroSplatObject.SyncAll()). Note that you may have to hook up other things, like the keyword file, or PropData file if you want per texture properties to be different for that version.

Q: In builds the terrain is black
A: This can happen for a number of different reasons. If you’re using asset bundles, usually this is because some data wasn’t loaded that needs to be, or you are running with draw instanced on android. Unity has a known bug which they have marked “Won’t Fix”, and the workaround is to either include a copy of one of your terrains in the actual build instead of in a bundle, or turn off draw instancing.

If you are not using asset bundles, then as a first step I would gather data by doing the following:

- Debug.Log out as much info as you can about the MicroSplatTerrain component and material, to make sure everything is hooked up correctly.
- See if it’s some kind of startup/script ordering issues by calling MicroSplatTerrain.SyncAll() after a frame has passed

- Try toggling Draw Instanced on the terrain
- See if particular features affect it, especially ones like tessellation or detail noise
- Try clearing the library folder, as it gets corrupted sometimes.

These types of issues rarely ever repro in fresh projects, and almost always come down to modifications made in the users project. I'm happy to help when I can, but you're going to have to do the heavy lifting here.