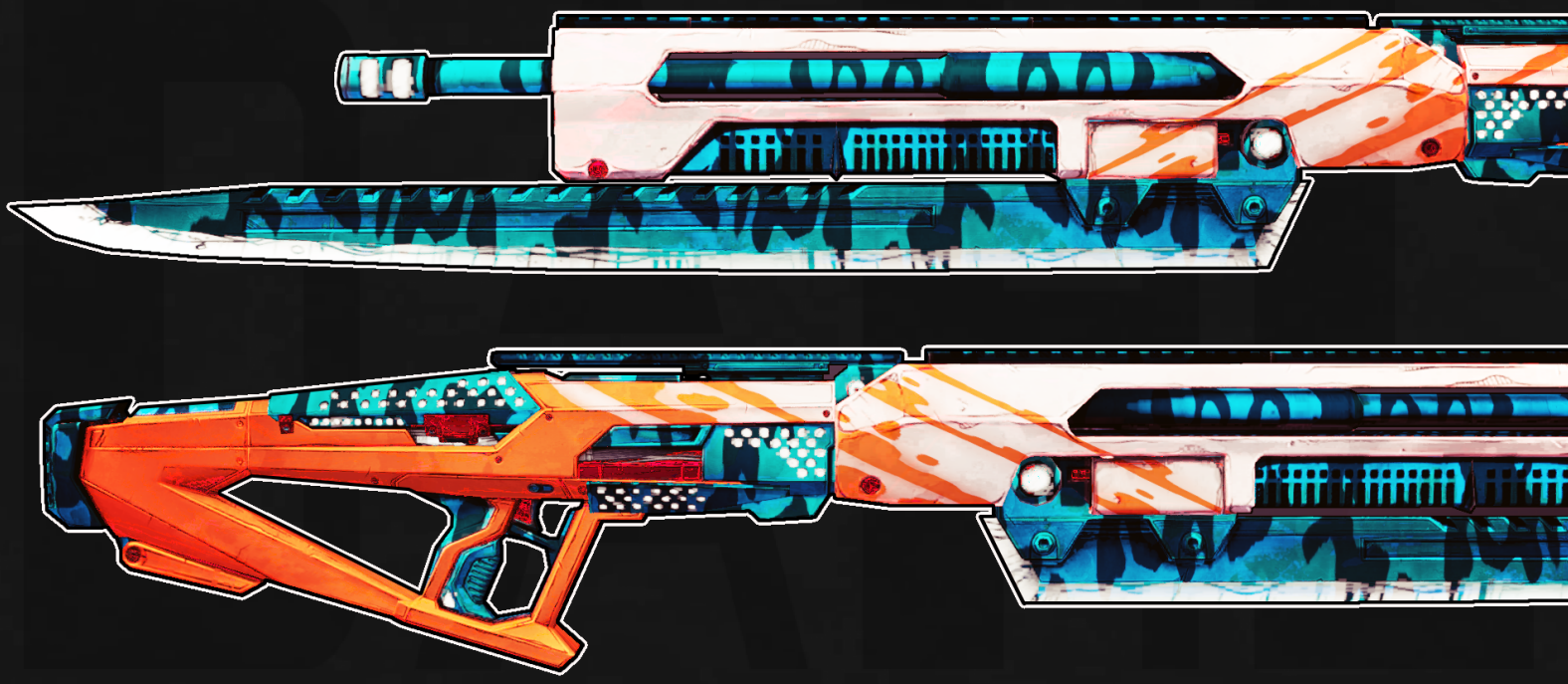
MODDING WEAPON SKINS IN BORDERLANDS 2



An in-depth guide to texture editing via console

--by Tofu! 木良

0 - INTRODUCTION

So, you want to make cool skins huh? Well, if that's the case that's good because that's exactly why we're here!

First off, a couple words: a massive shoutouts to Craig a.k.a. Shadowevil and Turk645 for getting me into this.
I learned everything I'm about to share in this guide on my own, however, if it wasn't for those two gentlemen showing me that editing was possible in the first place I wouldn't have discovered any of this.
Also, a very special thanks to Mr. Masterkizz for first getting me into Borderlands texture modding to begin with way back when TexMod editing was the only tool we had at our disposal. His first TexMod tutorial was the reason why I started spending so much time doing this stuff, so thanks for that mate.

Second, before we can actually get started you need to have in-game console access enabled, and that is because we're going to be modifying the game through a series of 'Set' commands. If you're familiar with the unofficial Community Patch for Borderlands 2 then you can probably skip this step because chances are you've already gone through this and you can already execute commands. If you don't, then here's a link to a quick tutorial explaining everything you need to know; it only takes a minute and it's very simple. Once you've done that, we can finally get started.

ALSO, ALL RESOURCED LINKED IN THIS GUIDE WILL ALSO BE AVAILABLE IN MY GITHUB RESOURCES FOLDER LOCATED HERE: [LINK]

1 - BASIC INFORMATION

1.1-Commands

Let's start from the very beginning: commands.
Luckily for us, we only need two commands to do any of this: 'exec' and 'set'. The first is used to (obviously) execute our mods which will then *instantly* take effect inside the game, and the second is used within said mod files to specify what parameters we want to modify. Ultimately, the only command we will be inputting in our in-game console will be 'exec' followed by our mod file name. To execute your mod file, simply place it in your \Borderlands 2\Binaries folder, go in-game, open your console and type:

exec filename.txt

Remember to include the "*.txt*" after whatever you named your mod file.
Skin mods *CAN be executed in real time, even while inspecting a weapon*, which is in fact really nice. You can simply alt-tab, change any value, *save* your file, alt-tab back to the game and re-execute it to see changes immediately happen. It's literally magic.

Speaking of mod files, here's what an unpacked blank structure looks like:

```
############ SKIN STRUCTURE #################

##set <ID> Parent <ID>

set <ID> TextureParameterValues (

(ParameterName="p_NormalScopesEmissive",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Nrm',ExpressionGUID=(A=-1743872746,B=1126171774,C=119496871,D=-1952271718)),
(ParameterName="p_Diffuse",ParameterValue=Texture2D'Common_GunMaterials.CompTextures.Weap_LauncherShotgunPistol_Comp',ExpressionGUID=(A=1757607260,B=1326539502,C=-317215581,D=-565807463)),
(ParameterName="p_Masks",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Comp',ExpressionGUID=(A=-330624041,B=1167251458,C=-335191907,D=-166684394)),
(ParameterName="P_SimpleReflect",ParameterValue=Texture2D'Common_GunMaterials.Env.Gold',ExpressionGUID=(A=-858148940,B=1327945772,C=148462268,D=1899047224)),
(ParameterName="p_Pattern",ParameterValue=Texture2D'Common_GunMaterials.Patterns.Patter_MaliwanFire',ExpressionGUID=(A=534250533,B=1202550002,C=1578302861,D=-717876416)),
(ParameterName="p_Decal",ParameterValue=Texture2D'Common_GunMaterials.Patterns.Pattern_JakobsEpic_SpaltedMaple',ExpressionGUID=(A=-923052711,B=1309861752,C=584229786,D=-1229888527))
)

set <ID> VectorParameterValues (

(ParameterName="p_AColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143)),
(ParameterName="p_AColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473504356,B=1338058895,C=824823046,D=864253013)),
(ParameterName="p_AColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590041,B=1156405294,C=-1015192900,D=687313410)),
(ParameterName="p_BColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170714760,B=1132476783,C=-275668290,D=655702143)),
(ParameterName="p_BColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473594356,B=1338758895,C=824823946,D=864253813)),
(ParameterName="p_BColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590341,B=1156435294,C=-1015192901,D=687313413)),
(ParameterName="p_CColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170714760,B=1132476783,C=-275668290,D=655702143)),
(ParameterName="p_CColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473594356,B=1338758895,C=824823946,D=864253813)),
(ParameterName="p_CColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590341,B=1156435294,C=-1015192901,D=687313413)),
(ParameterName="p_DColor",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=696455109,B=1155878830,C=-1741888361,D=802120528)),
(ParameterName="p_EmissiveColor",ParameterValue=(R=0.000000,G=0.000000,B=0.000000,A=1.000000),ExpressionGUID=(A=-2074486426,B=1296399582,C=-2021314681,D=-350758005)),
(ParameterName="p_ReflectColor",ParameterValue=(R=16.000000,G=16.000000,B=16.300000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976)),
(ParameterName="p_ReflectionChannelScale",ParameterValue=(R=0.450000,G=0.60000,B=0.900000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976)),
(ParameterName="p_PatternColor",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=676539706,B=1125682796,C=1871983293,D=-2049503601)),
(ParameterName="p_PatternScalePosition",ParameterValue=(R=12.000000,G=8.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=-2005018406,B=1132497243,C=-39915121,D=208423616)),
(ParameterName="p_PatternChannelScale",ParameterValue=(R=0.500000,G=0.050000,B=1.000000,A=0.000000),ExpressionGUID=(A=439432319,B=1091149893,C=-1991909502,D=1816944627)),
(ParameterName="p_DecalColor",ParameterValue=(R=0.800000,G=1.600000,B=1.600000,A=1.000000),ExpressionGUID=(A=1691998600,B=1239094551,C=2074257317,D=1844701893)),
(ParameterName="p_DecalScalePosition",ParameterValue=(R=17.100000,G=24.100000,B=0.400000,A=.500000),ExpressionGUID=(A=395540170,B=1243133493,C=-1264190552,D=123075385)),
)


set <ID> ScalarParameterValues (

(ParameterName="p_HighlightsIntensity",ParameterValue=0.450000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173)),
(ParameterName="p_ShadowsIntensity",ParameterValue=1.000000,ExpressionGUID=(A=437293753,B=1205147708,C=-775723903,D=1480014964))
(ParameterName="p_ReflectColorScale",ParameterValue=1.000000,ExpressionGUID=(A=1875785607,B=1186033550,C=-1822263113,D=-1465755701)),
(ParameterName="p_ReplacePattern",ParameterValue=1.000000,ExpressionGUID=(A=-2084339847,B=1096440125,C=439008937,D=45433490))
(ParameterName="p_DecalRotate",ParameterValue=0.000000,ExpressionGUID=(A=-276527909,B=1298581551,C=856978878,D=743944047)),
(ParameterName="p_ReplaceDecal",ParameterValue=1.900000,ExpressionGUID=(A=85863466,B=1257609701,C=-728575820,D=1337098176)),
(ParameterName="p_UseFullColorDecal",ParameterValue=1.000000,ExpressionGUID=(A=-1064329812,B=1077705328,C=339664807,D=1869745420))
)
```

As you can see, (ignoring the 'set <ID> Parent <ID>' line for a second) the file is divided in 3 main sections: TextureParameterValues, VectorParameterValues and ScalarParameterValues. The first one - TextureParameterValues - is the section we'll use to select our texture files which include both mapping and patterns/decals. The second - VectorParameterValues - is where the fun stuff happen. This section is reserved to properties such as colors and texture size/placement. Lastly, the third section - scalarparametervalues - is where so called Scalar Parameters go. We'll use this one the least and mostly for tweaking purposes. We'll go through all of these individually and explain each one in detail. Let's get started!

1.2 - Creating a mod file

First off of course, we need a file to write stuff in and that we will then execute. Luckily for us, this is extremely simple: all you need is simply create a new text file (.txt) so go grab your trusty Notepad, open up a new page and let's start writing stuff. Here's a link to the base structure pictured in the previous image. Feel free to copy all of it and paste it in a new document or just use it to grab specific lines you may need. At the bottom there's a few other extra things, don't worry about that just now. As mentioned, save your mods in your \Borderlands 2\Binaries folder.

1.3 - Acquiring weapons' ID's

Before starting to modify stuff we need to obviously tell the game which skins we're interested in editing. In Borderlands 2, weapons are assigned what is called a Material which determines what they will look like. Luckily for us, every single weapon has a unique Material which can be specifically edited without altering anything else in the game. Here's a link to a log including *all* Materials that exist in Borderlands 2. I suggest you copy that whole list and save it in a new text file somewhere, so that you can retrieve it any time you need it! Once you've done that, simply press Ctrl+F and type the name of the weapon you're looking for. The very large majority of them are named what you'd expect and should be very easy to find; some of them aren't titled exactly what they're called in-game, however, their names are still extremely obvious (for example, the Bunny ID does not say "Bunny", instead it says "Tediore Legendary Launcher". Should still be easy enough to find).

Once you've found the line you're looking for, the weapon's Material's ID is stated right after "MaterialInstanceConstant" and ends right before ".Name".
Here's an example to make this absolutely painfully easy to understand: entry 379 is the Material assigned to the Emperor SMG. The highlighted part is the ID we're after:

```
Log: 379) MaterialInstanceConstant Common_GunMaterials.Materials.SMG.Mati_Dahl_Legendary_Emperor.Name = Mati_Dahl_Legendary_Emperor
```

1.4 - Why "set Parent" is bad

Now, this is a minor sidenote in but it's worth talking about. The set Parent command can (ish) be used to swap any pre-existing skin onto any other weapon. Here's an example of how to use it:

```
set Common_GunMaterials.Materials.Launcher.Mati_BanditLegendaryRL Parent Aster_GunMaterials.Materials.SMG.Mati_Bandit_Quartz_SMG
```

As you can see, the syntax for it is the following:

```
set <ID> Parent <ID>
```

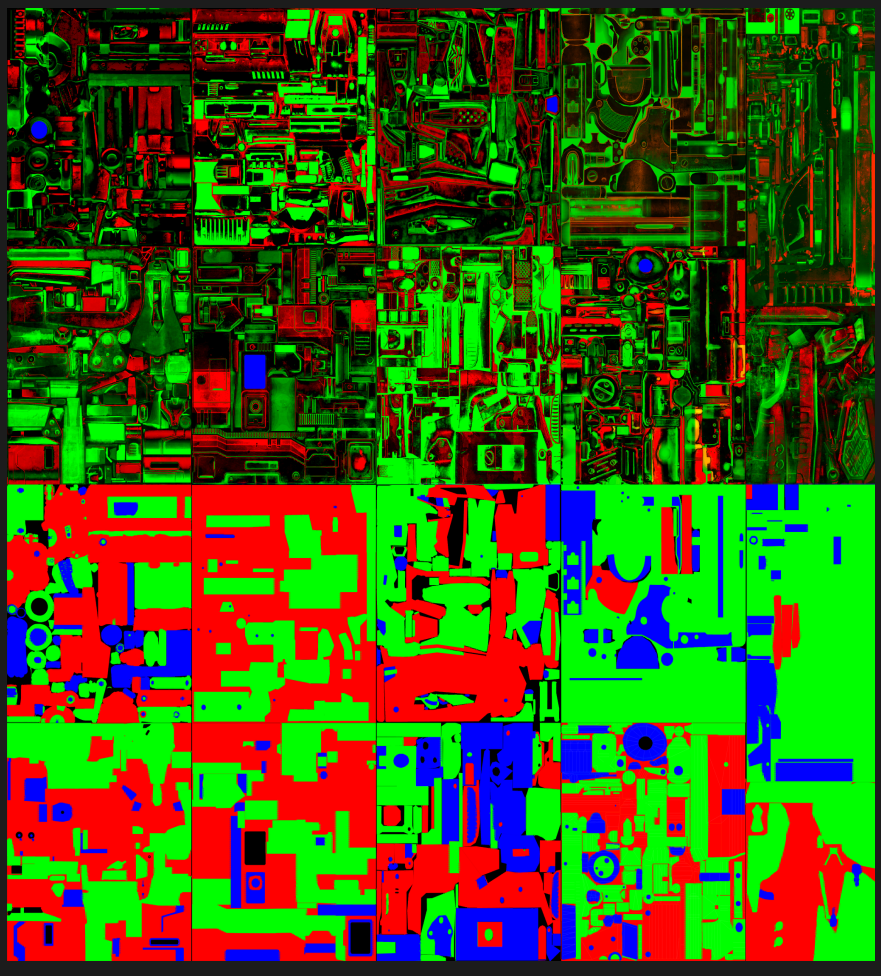
In the previously shown example, the Quartz skin from Tiny Tina's DLC is being applied to the Badaboom's Material.

Now, while this can *sometimes* be used as a quick solution if you just want to do a basic skin swap, there's many reason why it's a bad idea. First off, Materials are defined on a 'per case' basis, meaning that they are technically only meant to be put on the weapon they were originally made for. If you try to put a foreign Material on a weapon you often end up with stuff looking weird and out of place simply because it wasn't meant to be there in the first place, and therefore it's missing information that would normally tell the engine what to do with said Material. This basically means that it only works *sometimes* and between a few selected Materials and weapons combinations; overall it's not reliable at all. Moreover, if you use a Parent replacement you are de-localizing things: at that point you're not modifying one single skin anymore, but you're also bringing in an additional one into the mix which might result in "collateral damage" and alter other things you might not want to change. Besides, once you learn how to edit skins you don't need to do that anymore anyway; you'll be able to simply recreate any other Material in the game with little effort. So yeah, in case it wasn't clear enough, don't use Parent replacements. Just don't. Please.

2 - GUNS STRUCTURE

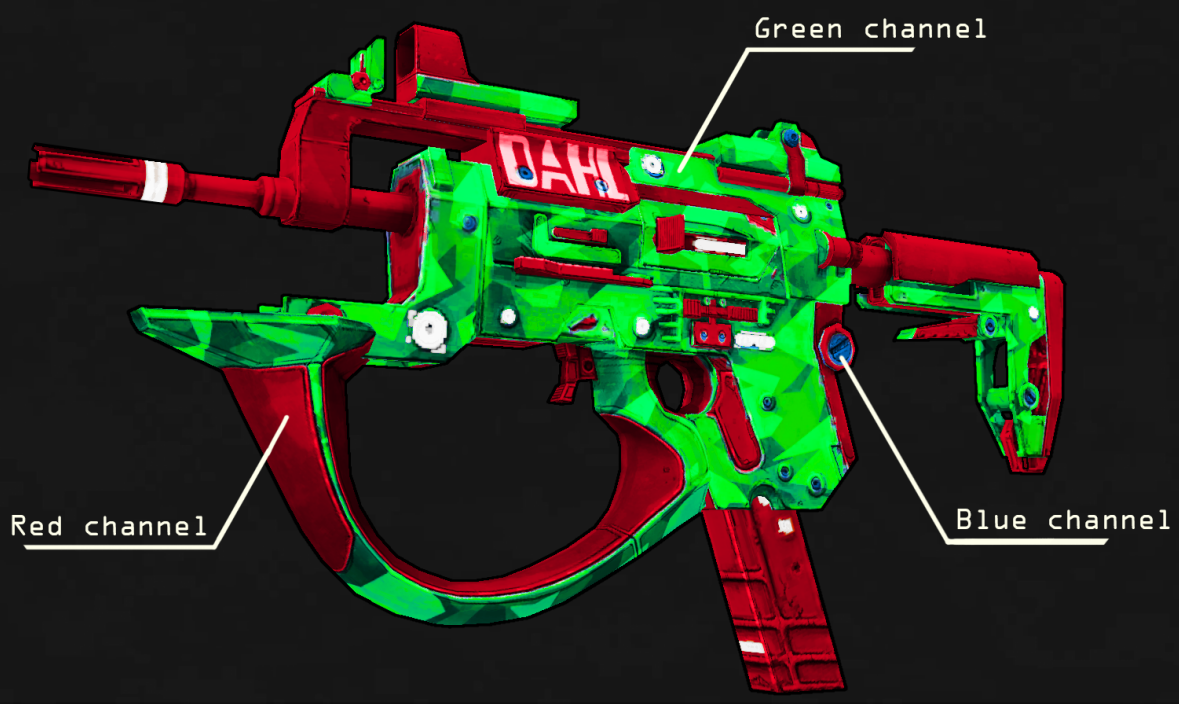
2.1 Channels

Enough boring stuff, let's start actually explaining how things work.
First off, RGB channels. This is what a texture map in Borderlands 2 looks like:



As you can see, there's a bunch of red, green and blue going on. That's because layers are placed on weapons using RGB (Red-Green-Blue) channels. We can use these to specifically decide which parts of our weapon need to be colored a certain way, have a certain pattern, reflection layer, etc. Here's a visual representation of what that would look like on a gun if it used those colors as the actual physical texture:

Visual representation of RGB channels on a Dahl SMG



Every weapon always has 3 'sections' plus a glowing one

Depending on the gun type and manufacturer, those 3 can vary in size and placement, however, they never change in number: no matter what weapon it is, it will always have 3 RGB channels plus a glowing one (White in the previous picture; this 4th channel is used for the elemental glow you see on weapons skins. If the gun doesn't have an element then it can be colored. More on the specifics of that later).

Roughly, these 3 channels are used to "block out" the various parts each gun has, and those usually are the main body, the metal parts and finally the smaller details like screws or handle covers and stuff of the sort. Think for example what the original vanilla Emperor looks like: you have the main body which is orange, black metal parts, and lastly a few smaller accents which are barely even noticeable.
We will learn how to actually utilize channels in a moment, however it is very important to understand how they work on a conceptual level first. Think of channels not as something you can directly see but more as a guidance for what you do see.

2.2 - Layers VS Channels

So at this point you might be wondering "Ok cool, I understand what channels do, but if there's only 3 of those on every gun how come there's guns with a whole bunch of stuff, logos, patterns, all kinds of colors etc, that's totally more than just 3 things!", and the answer to that is, layers!
To put it easily, layers are "stuff" you put on your weapon. By "stuff" I mean the pretty shiny things you actually see in-game, which include: 3 base coat colors, a pattern layer, a decal layer and a reflective layer, and - here's the cool thing - each one of these layers has its own specific channel definition, which means that for every single one of them you can decide where you want it to be visible on the weapon. Let's say, you want a pistol that has a wooden handle and a metal barrel: in that case you would set your shiny reflective to be on the metal part - so they look like metal - but not on the wooden parts. Or, let's say you want a sweet Dahl camo pattern on your sniper rifle but of course not just painted on literally everything because that looks super lame, but instead you just want it on the body of the gun for example; in that case you can simply go to your pattern's channel and set it to only cover one of the 3 sections of the gun.
To use an analogy, think of traffic: layers are shiny colorful cars driving around, and channels are signs telling them where they need to go. Signs cannot be changed, however every single car can decide which sign to follow.

3 - Getting started

3.1 - Structuring your code

Now that we have a basic understanding of how things are structured, we can finally start writing things, wohoo!
If you've made it this far, well congrats on going through actual pages of theory (and hey thanks for still being here, it means this thing is actually working and you're willing to learn this stuff. Go get yourself a cookie, you deserve one).

Remember at the start where I showed you what a basic blank 'skin file' looks like? Yes, this one right here:

```
############ SKIN STRUCTURE #################

##set <ID> Parent <ID>

set <ID> TextureParameterValues (

(ParameterName="p_NormalScopesEmissive",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Nrm',ExpressionGUID=(A=-1743872746,B=1126171774,C=119496871,D=-1952271718)),
(ParameterName="p_Diffuse",ParameterValue=Texture2D'Common_GunMaterials.CompTextures.Weap_LauncherShotgunPistol_Comp',ExpressionGUID=(A=1757607260,B=1326539502,C=-317215581,D=-565807463)),
(ParameterName="p_Masks",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Comp',ExpressionGUID=(A=-330624041,B=1167251458,C=-335191907,D=-166684394)),
(ParameterName="P_SimpleReflect",ParameterValue=Texture2D'Common_GunMaterials.Env.Gold',ExpressionGUID=(A=-858148940,B=1327945772,C=148462268,D=1899047224)),
(ParameterName="p_Pattern",ParameterValue=Texture2D'Common_GunMaterials.Patterns.Patter_MaliwanFire',ExpressionGUID=(A=534250533,B=1202550002,C=1578302861,D=-717876416)),
(ParameterName="p_Decal",ParameterValue=Texture2D'Common_GunMaterials.Patterns.Pattern_JakobsEpic_SpaltedMaple',ExpressionGUID=(A=-923052711,B=1309861752,C=584229786,D=-1229888527))
)

set <ID> VectorParameterValues (

(ParameterName="p_AColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143)),
(ParameterName="p_AColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473504356,B=1338058895,C=824823046,D=864253013)),
(ParameterName="p_AColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590041,B=1156405294,C=-1015192900,D=687313410)),
(ParameterName="p_BColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170714760,B=1132476783,C=-275668290,D=655702143)),
(ParameterName="p_BColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473594356,B=1338758895,C=824823946,D=864253813)),
(ParameterName="p_BColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590341,B=1156435294,C=-1015192901,D=687313413)),
(ParameterName="p_CColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170714760,B=1132476783,C=-275668290,D=655702143)),
(ParameterName="p_CColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=473594356,B=1338758895,C=824823946,D=864253813)),
(ParameterName="p_CColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=-429590341,B=1156435294,C=-1015192901,D=687313413)),
(ParameterName="p_DColor",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=696455109,B=1155878830,C=-1741888361,D=802120528)),
(ParameterName="p_EmissiveColor",ParameterValue=(R=0.000000,G=0.000000,B=0.000000,A=1.000000),ExpressionGUID=(A=-2074486426,B=1296399582,C=-2021314681,D=-350758005)),
(ParameterName="p_ReflectColor",ParameterValue=(R=16.000000,G=16.000000,B=16.300000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976)),
(ParameterName="p_ReflectionChannelScale",ParameterValue=(R=0.450000,G=0.60000,B=0.900000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976)),
(ParameterName="p_PatternColor",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=676539706,B=1125682796,C=1871983293,D=-2049503601)),
(ParameterName="p_PatternScalePosition",ParameterValue=(R=12.000000,G=8.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=-2005018406,B=1132497243,C=-39915121,D=208423616)),
(ParameterName="p_PatternChannelScale",ParameterValue=(R=0.500000,G=0.050000,B=1.000000,A=0.000000),ExpressionGUID=(A=439432319,B=1091149893,C=-1991909502,D=1816944627)),
(ParameterName="p_DecalColor",ParameterValue=(R=0.800000,G=1.600000,B=1.600000,A=1.000000),ExpressionGUID=(A=1691998600,B=1239094551,C=2074257317,D=1844701893)),
(ParameterName="p_DecalScalePosition",ParameterValue=(R=17.100000,G=24.100000,B=0.400000,A=.500000),ExpressionGUID=(A=395540170,B=1243133493,C=-1264190552,D=123075385)),
)


set <ID> ScalarParameterValues (

(ParameterName="p_HighlightsIntensity",ParameterValue=0.450000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173)),
(ParameterName="p_ShadowsIntensity",ParameterValue=1.000000,ExpressionGUID=(A=437293753,B=1205147708,C=-775723903,D=1480014964))
(ParameterName="p_ReflectColorScale",ParameterValue=1.000000,ExpressionGUID=(A=1875785607,B=1186033550,C=-1822263113,D=-1465755701)),
(ParameterName="p_ReplacePattern",ParameterValue=1.000000,ExpressionGUID=(A=-2084339847,B=1096440125,C=439008937,D=45433490))
(ParameterName="p_DecalRotate",ParameterValue=0.000000,ExpressionGUID=(A=-276527909,B=1298581551,C=856978878,D=743944047)),
(ParameterName="p_ReplaceDecal",ParameterValue=1.900000,ExpressionGUID=(A=85863466,B=1257609701,C=-728575820,D=1337098176)),
(ParameterName="p_UseFullColorDecal",ParameterValue=1.000000,ExpressionGUID=(A=-1064329812,B=1077705328,C=339664807,D=1869745420))
)
```

Go grab that, cause we're gonna need it right now. If you need a link to it again go back to page 2, it's right there.

As the title of this chapter suggests, this part is all about structure. Having a well-structured file helps you find stuff more easily and quickly. If you've done any coding at all, even at a very basic level, you probably understand how important having tidy code is; if stuff breaks it's also way easier to find typos and mistakes you might have sneaking around.

First of all, you can write comment lines with #. This means, any line that starts with a # sign will be ignored by the game; this is very useful to label stuff or write notes to yourself without the game getting confused.
Now that this is out of the way, let's start looking at our actual commands and start doing the fun stuff.

3.2 - Syntax

Since we're substantially writing code, syntax (a.k.a. "how you write stuff") is extremely important. If you forget a bracket, a comma, a space, anything at all, your code simply will not work. Luckily for us, this stuff is very intuitive and straight forward to write. The basic command structure the engine wants us to use is the following:

set <ID> Attribute ((parameter1),(parameter2),(parameter3),(parameter#))

Quick note, there's no limit to the number of parameters you can use. If you want something simple, you can just use a couple. If you want to go overkill, grab all the ones you like. The only thing you need to make sure is to put a comma between one parameter and the next one. Also, ALWAYS double check your brackets!
You might notice that in the base file I provided you with stuff looks a bit different: I go to a new line after each parameter. That is just a momentary 'unpacked' state I use to see things more clearly. After I'm sure everything is in order and nothing is missing I then proceed to delete all unnecessary spaces and pack everything back into a single line.
To make this super painfully clear:

"Unpacked" form:

set <Material ID> Attribute (
(parameterName=Parameter1, ParameterValue=1),
(parameterName=Parameter2, ParameterValue=2),
(parameterName=Parameter3, ParameterValue=3)
)

After being packed back up becomes:

set <Material ID> Attribute ((parameterName=Parameter1, ParameterValue=1),(parameterName=Parameter2, ParameterValue=2),
(parameterName=Parameter3, ParameterValue=3))

Easy enough, right?

3.3 - Parameters list

Let's just jump into this. Here's a quick first overview of all parameters we can use and a super brief explanation of what they do. I will use this as pretty much an index to explain things, going over it once again and explaining everything in detail.

First off, as briefly mentioned at the start all parameters are divided in 3 major sections: TextureParameterValues, VectorParameterValues and ScalarParameterValues. Here's a list of everything each of them contains:

TextureParameterValues — These are all your base textures addresses

p_NormalScopesEmissive]
p_Diffuse — These 3 are texture maps and need to *ALWAYS* be there,
p_Masks]
P_SimpleReflect — Reflective layer texture,
p_Pattern — Pattern layer texture,
p_Decal — Decal layer texture,

VectorParameterValues — These control colors and channels

p_AColorHilight — Primary highlight's color,
p_AColorMidtone — Primary midtone's color,
p_AColorShadow — Primary shadow's color,
p_BColorHilight — Secondary highlight's color,
p_BColorMidtone — Secondary midtone's color,
p_BColorShadow — Secondary shadow's color,
p_CColorHilight — Tertiary highlight's color,
p_CColorMidtone — Tertiary midtone's color,
p_CColorShadow — Tertiary shadow's color,
p_DColor — Accent color,
p_EmissiveColor — Glow color,
p_ReflectColor — Controls the reflective texture's color,
p_ReflectionChannelScale — Controls which channels the reflective layer will be on,
p_PatternColor — Controls the pattern texture's color,
p_PatternScalePosition — Controls the pattern layer's size and position,
p_PatternChannelScale — Controls which channels the pattern layer will be on,
p_DecalColor — Controls decal color,
p_DecalScalePosition — Controls decal size and position,
p_DecalChannel — Controls which channels the decal will be on,

ScalarParameterValues — These are single value parameters

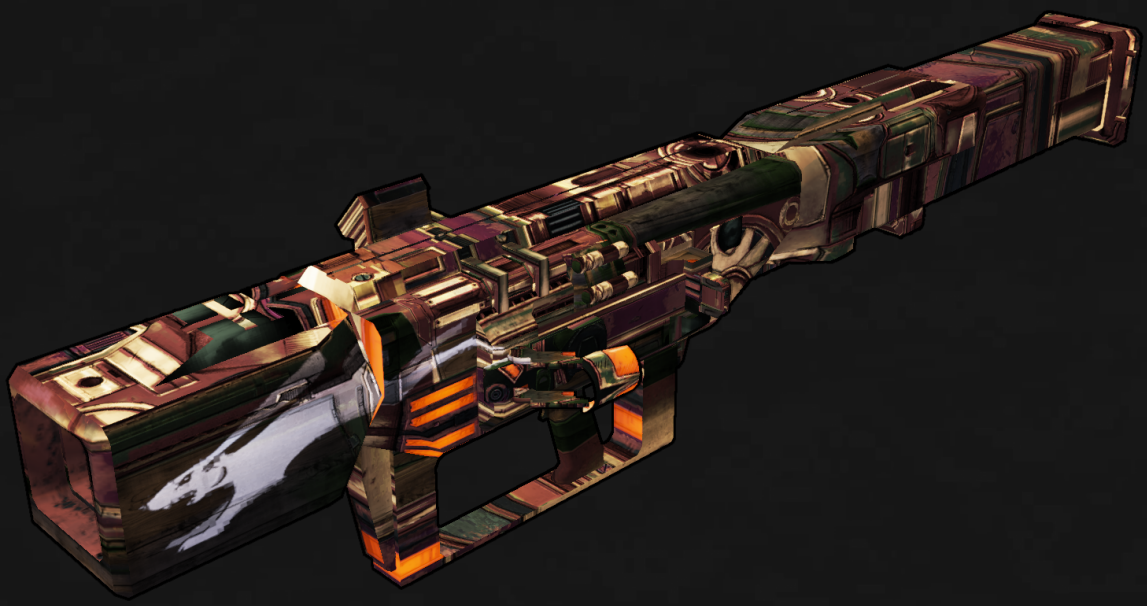
p_HighlightsIntensity — Controls the intensity of the Material's highlights,
p_ShadowsIntensity — Controls the intensity of the Material's shadows,
p_ReflectColorScale — Controls the reflective layer's intensity,
p_UseFullColorDecal — Determines if the decal uses its full colors or not,
p_DecalRotate — Determines if and how much the decal layer is rotated,
p_ReplaceDecal]
p_ReplacePattern — These two control how Decal and Pattern are overlayed.
p_EmissiveScale — This changes your elemental glow intensity.

So there you go. These are all the ingredients you need to craft any weapon skin you dream of. Well, within the limitations of the game, that is.
Now that we have a list down, let's go through it in detail.

4 - TextureParameterValues

As mentioned, this section is where you pick your base textures. First off, maps:

Why correct mapping is extremely important



Bunny launcher with an incorrectly mapped
Vladof assault rifle skin

Some of you might remember before quality checks where a thing, when 'Black rarity weapons' were popular. Many of them had ultra awful skins like the one pictured above. The reason for that is very simple: incorrect mapping. To put it in simple words, every weapon type (launchers, pistols, shotguns, etc.) have type-specific textures that make sure everything is mapped correctly and placed on the right parts. Every single weapon skin mod you'll ever do will *ALWAYS* start with the Holy Texture Trinity:

★★★ p_NormalScopesEmissive, p_Diffuse and p_Masks ★★★

Since these are so vital, I've included a quick copy-paste table at the bottom of the base text file. Here's an example of what these actually look like all together:

```
set <ID> TextureParameterValues (

(ParameterName="p_NormalScopesEmissive",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Nrm',ExpressionGUID=(A=-1743872746,B=1126171774,C=119496871,D=-1952271718)),
(ParameterName="p_Diffuse",ParameterValue=Texture2D'Common_GunMaterials.CompTextures.Weap_LauncherShotgunPistol_Comp',ExpressionGUID=(A=1757607260,B=1326539502,C=-317215581,D=-565807463)),
(ParameterName="p_Masks",ParameterValue=Texture2D'Weap_Pistol.Tex.Weap_Pistols_Comp',ExpressionGUID=(A=-330624041,B=1167251458,C=-335191907,D=-166684394)),
)
```

In this example (taken from the base blank file), textures are being mapped to a pistol. If you're working on any other gun type, find the 3 ones you need in the list and copy/paste.
THESE ARE ALWAYS MANDATORY! If not used, textures will also unload from your gun while in-game. Adding these 3 lines will also fix any existing unloading issue.

Now that we're done with the obligatory stuff, we can move on to the magic creative part. From now on, all this stuff is *optional*, meaning you can decide if you want to use it or not. If you don't, the skin will retain its default properties. Also, once again mod files *CAN* be executed via console while in-game, even while inspecting the weapon you're modifying! So, let's continue going through our parameters:

— **p_SimpleReflect** —Here's what it looks like in text form:

(ParameterName="P_SimpleReflect",ParameterValue=Texture2D'Common_GunMaterials.Env.GlossyD',ExpressionGUID=(A=-858148940,B=1327945772,C=148462268,D=1899047224))

—Here's what it does:

If added, this line places a reflective texture on your weapon, which will look like a glossy-like finish in most cases. You can *technically* use any texture for this, however there are a few ones specifically made for the job so they will obviously look best. I've included a small list of the main ones at the very bottom of the base text file. If you want to edit it, simply swap the texture name (highlighted in cyan in the example above) with anything you like.

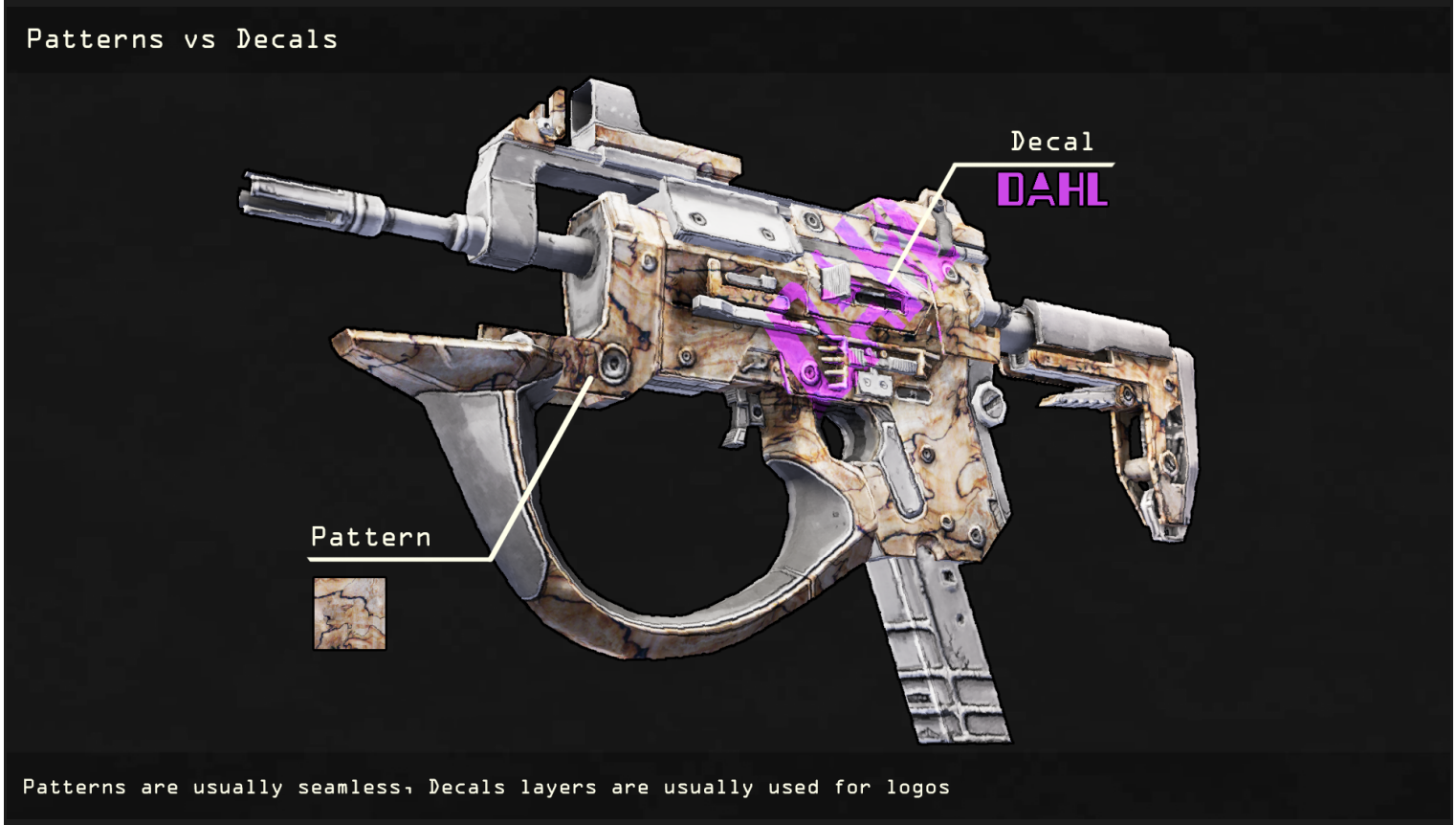
— **p_Pattern** and **p_Decal** —Here's what they look like in text form:

(ParameterName="p_Pattern",ParameterValue=Texture2D'Common_GunMaterials.Patterns.Pattern_DahlLegendCamo1',ExpressionGUID=(A=534250533,B=1202550002,C=1578302861,D=-717876416))

(ParameterName="p_Decal",ParameterValue=Texture2D'Common_GunMaterials.Logos.Logo_TorgueHyperionDahl_Comp',ExpressionGUID=(A=-923052711,B=1309861752,C=584229786,D=-1229888527))

—Here's what they do:

If added, these lines place respectively a pattern texture and a decal on your weapon. Now, this is where stuff gets fun: this is where you can choose your Dahl camos, your Jakobs wood textures, your Moxxi logos, everything of that sort. Once again, changing the name highlighted in cyan will change your texture.



Patterns are usually seamless, Decals layers are usually used for logos

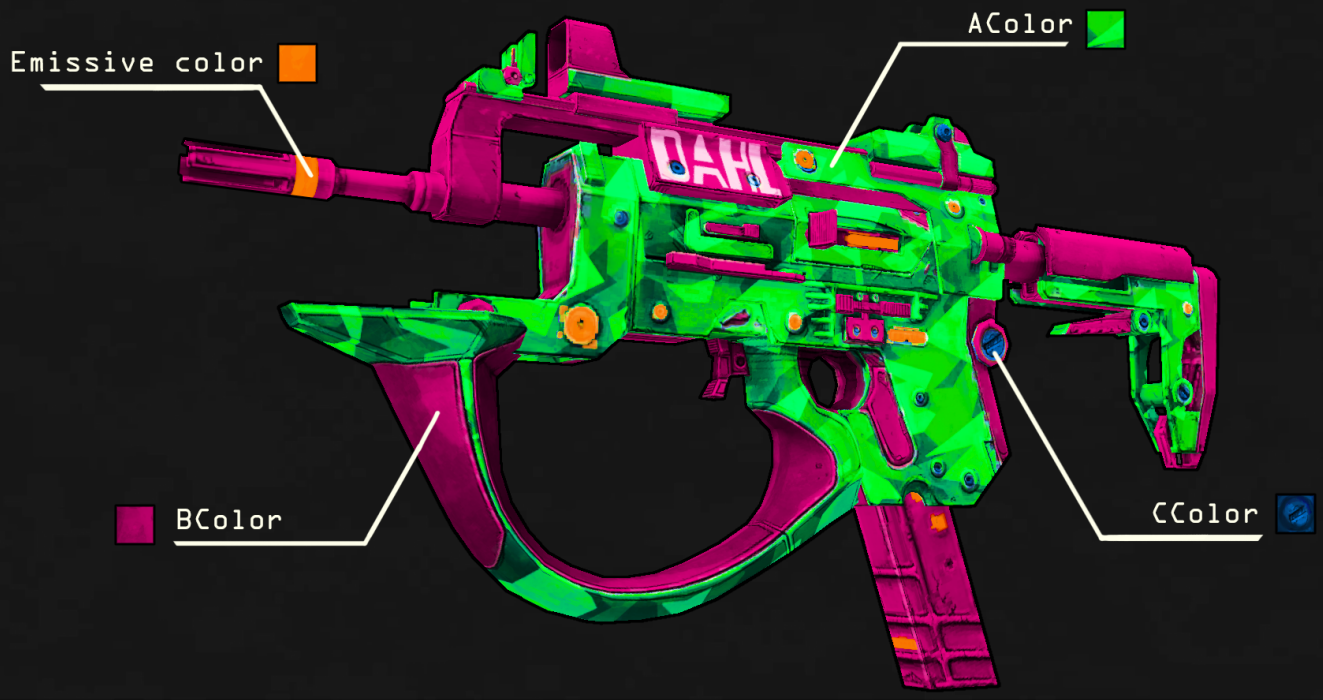
Usually, this is how patterns and decals are utilized, however, *in some cases you can use another pattern texture on your decal layer and end up with two patterns on the same gun*. A very important note must be said about this though: a lot of decals and patterns (but especially decals) have inherent unique properties that cannot be changed and sometimes will straight up not work. Sometimes weird visual errors can be salvaged (see p_UseFullColorDecal, p_ReplaceDecal and p_ReplacePattern later in the guide), but other times it just isn't possible.

Now you might be thinking, "Hey that's cool, but where do I get a list of all textures I can use for th-" -Don't worry buddy ol' pal, I got you covered. Here's a handy link to all textures in the game. Once again I suggest you save that somewhere so you can have it at hand at all times. Now, you can use *most* of these, however sometimes using texture from non-guns stuff will confuse the game and even crash it. You can still do stupid stuff like put Piro Pete's face on a gun, however make sure to test it to ensure it actually works!

5 - VectorParameterValues

Have I mentioned this next part is fun? Because this next part is fun. The almighty Vector section is where you'll be spending most of your time. This is where you set properties such as colors, channels and texture sizes/placements.
Let' start with base colors.

Main body colors



Now, this will already look like a deja-vu, and that's because it pretty much is. Let's ignore the Emissive color for a moment - don't worry, we'll get to that immediately after this. Just like channels, guns have 3 base colors that paint those same exact areas: those are named AColor, BColor and CColor. AColor is usually the main body of the gun (colored green in the example), BColor is usually assigned to metal parts and/or grips and accessories (colored magenta in the example) and lastly CColor is usually assigned to minor parts or color accents (colored blue in the example). If there's no pattern or decal covering them, base colors will show unaltered in all their glory. Here's how to edit them:

— **p_AColorHilight**

— **p_AColorMidtone**

— **p_AColorShadow** —Here's what they look like in text form:

(ParameterName="p_AColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_AColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_AColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))

—Here's what they do:

If added, these three lines modify your AColor highlights, mid-tone and shadow's colors. As previously mentioned, AColor is usually the base coat color assigned to the main body of your gun. To edit those, simply change the R G B values in the line. If you don't know how RGB (Red-Green-Blue) colors work, there's a bunch of RGB color pickers online, however it's not too difficult to figure out. If you want a nice red color for example, set R to a nice integer number and turn G and B off by setting them to zero. If you want a darker color, go into the decimals. For example, a mid-tone R set to 0.5 or lower gives a deep crimson red. Generally you'll find yourself tinkering with these 3 lines back and forth a lot until you nail down that perfect hue you like, so be prepared for a bunch of alt-tabbing.

And as you might imagine, BColor and Ccolor work in exactly the same way.
For the sake of consistency, I'll write these down too:

— p_BColorHilight

— p_BColorMidtone

— p_BColorShadow —Here's what they look like in text form:

(ParameterName="p_BColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_BColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_BColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))

—Here's what they do:

If added, these lines modify your BColor highlights, mid-tone and shadow's colors. Just like AColor, they use R G B values which can be modified very easily!

— p_CColorHilight

— p_CColorMidtone

— p_CColorShadow —Here's what they look like in text form:

(ParameterName="p_CColorHilight",ParameterValue=(R=6.000000,G=6.000000,B=6.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_CColorMidtone",ParameterValue=(R=4.000000,G=4.000000,B=4.000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))
(ParameterName="p_CColorShadow",ParameterValue=(R=2.000000,G=2.000000,B=2000000,A=1.000000),ExpressionGUID=(A=170014760,B=1132076783,C=-275608290,D=650702143))

-Here's what they do:

If added, these lines modify your CColor. Yes I know, you wouldn't have guessed it by now. Let's move on!

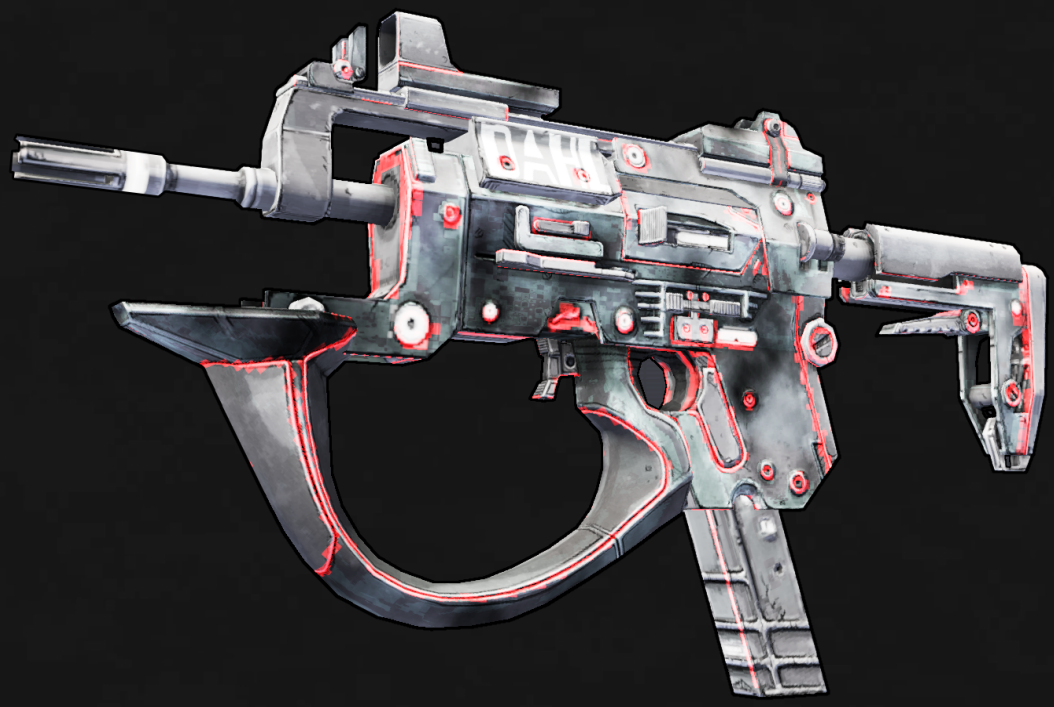
— p_DColor —Here's what they look like in text form:

(ParameterName="p_DColor",ParameterValue=(R=2.000000,G=2.000000,B=2.000000,A=1.000000),ExpressionGUID=(A=696455109,B=1155878830,C=-1741888361,D=802120528))

-Here's what they do:

Contrarily to what you might expect, Dcolor is not as 'in-depth' like its A B C siblings. DColor only uses a single line (once again with R G B values) and defines accent colors such as sights colors (sometimes), highlight lines and things like that. DColor is more of a detail and it's rarely even used at all, however it can still have really neat applications, so play around with it and see! Here's DColor on Dahl SMGs:

DColor example



"p_DColor",ParameterValue=
(R=250.000000,G=1.000000,B=1.000000,A=1.000000)

DColor can be a real hero sometimes.

— p_EmissiveColor —Here's what it looks like in text form:

(ParameterName="p_EmissiveColor",ParameterValue=(R=0.000000,G=0.000000,B=0.000000,A=1.000000),ExpressionGUID=(A=-2074486426,B=1296399582,C=-2021314681,D=-350758005))

-Here's what it does:

If added, this line modifies the "elemental" glowing color on your gun.
IMPORTANT NOTE: this will only change the glow color if the gun has no element. For example, if your Emperor SMG is corrosive, the glowing color will be green no matter what you set this value to. Think of the Bekah and Godfinger: they both have a cool unique magenta glow, but only because they have no element. If elements were allowed on those weapons their glow color would simply match their element.

— p_ ReflectColor —Here's what it looks like in text form:

(ParameterName="p_ReflectColor",ParameterValue=(R=8.000000,G=8.000000,B=8.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

If added, this line modifies the color of your reflective texture. If you want your guns to have a cool reflective shine then - you guessed it - change your R G B values to whichever color you like. Higher R G B values (depending on what texture you're using as your reflection layer - see P_SimpleReflect again if you need a reminder) will make the color brighter and more intense.

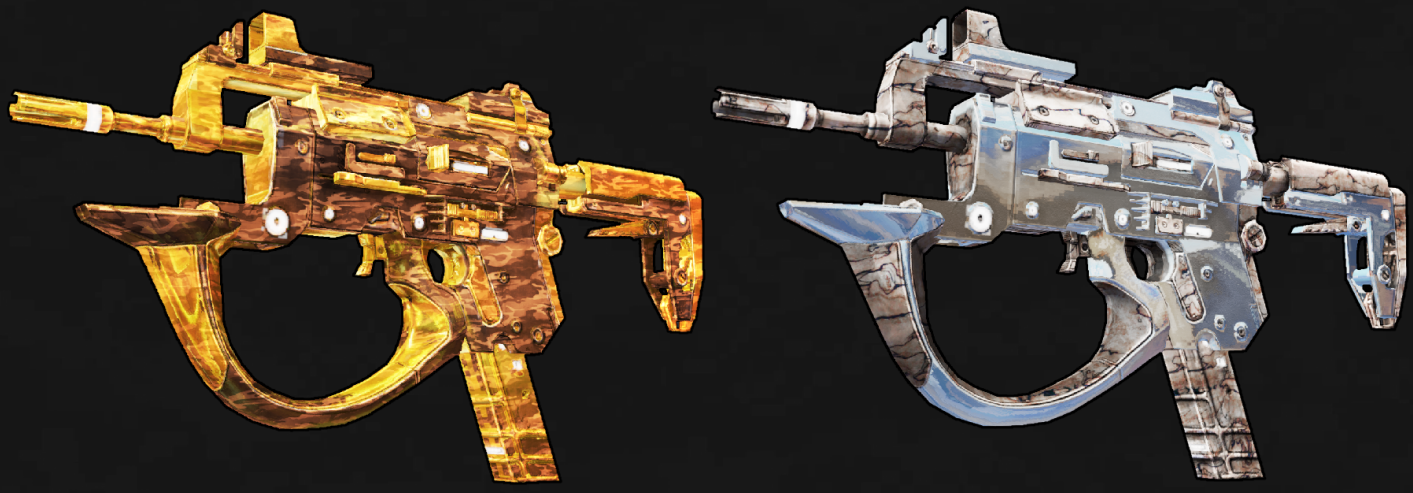
— p_ ReflectionChannelScale —Here's what it looks like in text form:

(ParameterName="p_ReflectionChannelScale",ParameterValue=(R=1.000000,G=1.000000,B=1.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

If added, this line defines which *channels* your reflective layer will appear on.
Remember the whole channels thing? Here's how we utilize that.

Reflective layer



Reflective layers also use channels independently

Reflective color scale defines how 'shiny' the layer is

Env.Gold set on all 3 channels, set to be semi-transparent on channel R

Env.Chrome set only on R channel, overlaying Decal layer

In this case, your R G B values set to 1.000000 or 0.000000 will turn their channels on and off. As you can see in the picture above, the weapon on the left has all three R G B channels enabled and is therefore all shiny. The one to the right only has its R channel set to 1, so only its main body will be shiny.
You can also set these values to smaller decimals to decide how transparent you want your layer to be: for example the SMG on the left has its G channel set to 0.5, and as a result it's only semi-transparent compared to the others.

— p_ PatternColor —Here's what it looks like in text form:

(ParameterName="p_PatternColor",ParameterValue=(R=3.000000,G=3.000000,B=3.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

If added, this line defines what color your Pattern (defined in your p_Pattern entry located in your TextureParameterValues section - see related picture from earlier) texture will be. Note that some patterns retain their original color when used on certain guns. For example, the Dahl rare camo (the blue one) natively has a blue hue, so if you want a different color you have to balance it out.

— p_PatternScalePosition —Here's what it looks like in text form:

(ParameterName="p_PatternColor",ParameterValue=(R=10.010000,G=10.010000,B=10.010000,A=10.010000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

If added, this line can be used to control your pattern's size and position. Your R and G values control the size: *reducing* their value will *increase* width and height. Always make sure to change these two in tandem so you can keep good proportions.
To move your pattern around, change your B and A values. Increasing and reducing these two will move your pattern up and down/left and right.
IMPORTANT: these values *are very sensitive*. Start increasing or decreasing your values in very small intervals (0.01 intervals are a good place to start) to see exactly how your pattern moves around. Also additional side-note, setting scale values to negative will flip the pattern vertically or horizontally, depending on which one you invert.

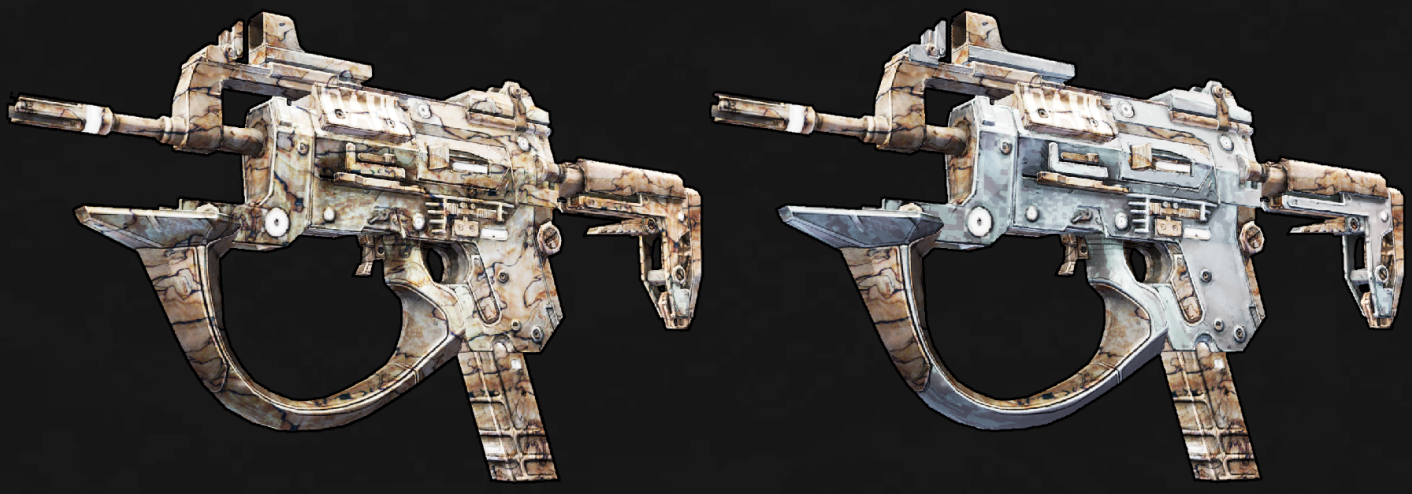
— p_PatternChannelScale —Here's what it looks like in text form:

(ParameterName="p_PatternChannelScale",ParameterValue=(R=1.000000,G=1.000000,B=1.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

Much like the previously explained reflection channel parameter, this works pretty much the same way but with patterns: it controls which channels your pattern will be on.
Here's a visual example to show patterns behavior on different channels:

Patterns channels behavior



All channels selected

"p_PatternChannelScale",ParameterValue=
(R=1.000000,G=1.000000,B=1.000000,A=1.000000)

A channel disabled,
B and C selected

"p_PatternChannelScale",ParameterValue=
(R=0.000000,G=1.000000,B=1.000000,A=1.000000)

Once again, your R G B values set to 1.000000 or 0.000000 will turn your channels on and off. In the picture above, the weapon on the left has all three R G B channels enabled and therefore shows the pattern everywhere. The one on the right only has its G and B channels set to 1, and as a result only parts of the handle, the mag and other secondary areas are covered by the pattern.
Also, just like the other Channels parameters, if you set these values to non-integers numbers between 0 and 1 (like 0.5 for example) the pattern will be semi-transparent.

— **p_DecalColor** —Here's what it looks like in text form:

(ParameterName="p_DecalColor",ParameterValue=(R=3.000000,G=3.000000,B=3.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

Yup, as you might expect this is pretty self explanatory. Change this value if you want to modify your decal color, simple as that.
NOTE: some decals have fixed colors which are retained on certain guns, and other times they just look glitchy and don't work at all as you'd expect. Sometimes this can be salvaged (see p_UseFullColorDecal and p_RelaceDecal later on), but in most cases if decals look weird and don't react well to color changes, it's probably not a good sign.

— **p_DecalScalePosition** —Here's what it looks like in text form:

(ParameterName="p_DecalScalePosition",ParameterValue=(R=17.100000,G=24.100000,B=0.400000,A=0.500000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

-Here's what it does:

Much like its pattern counterpart, this line can be used to control your Decal's size and position. Your R and G values control the size: *reducing* their value will *increase* width and height. Always make sure to change these two in tandem so you can keep good proportions - also again, setting these to negative values will flip your decal. Once again, if you also want to move your decal around change your B and A values. Increasing and reducing these two will move it up and down/left and right.
IMPORTANT: once again and *ESPECIALLY* these values, they *VERY* sensitive. The values I provided in the example here (same in the blank base file) place your decal around the middle of your gun so you'll have an easier time locating it and moving it around. Just like patterns' scale and position, start changing your values in 0.1's intervals so you can easily keep track of where that thing goes. Trust me, decals move a lot more than you'd expect so start with *small* values changes and then move up when you see what's up.

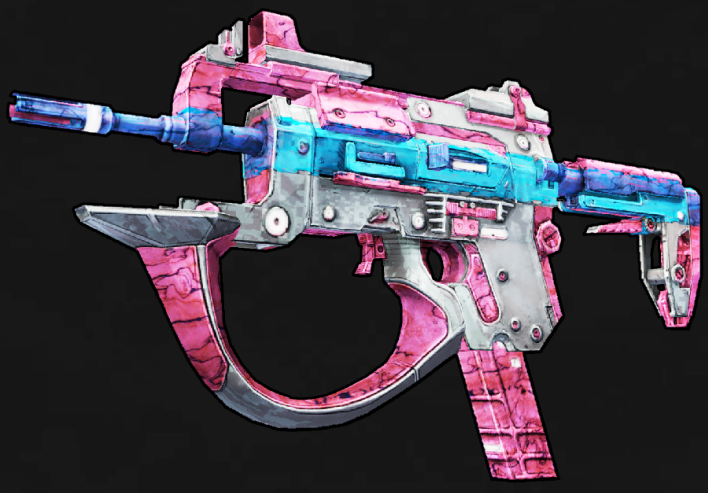
— **p_DecalChannel** —Here's what it looks like in text form:

(ParameterName="p_DecalChannel",ParameterValue=(R=1.000000,G=1.000000,B=1.000000,A=1.000000),ExpressionGUID=(A=295058103,B=1318551573,C=-2045449573,D=-547597976))

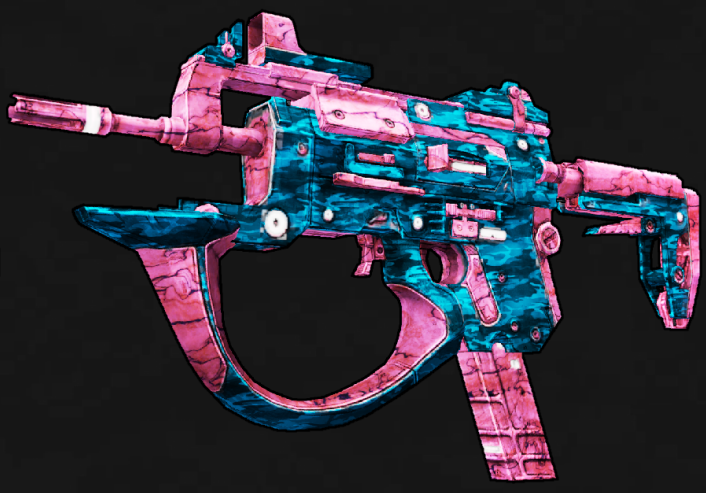
-Here's what it does:

You know this by now. This line is once again a channels control line and decides which ones your decal is gonna be on. Here's a picturee which pretty much summarizes everything you can do with Decal's channels, colors and scale/position:

Decals colors and channels



Decal and Pattern layers overlaying
on channels R and G (purple-ish color)



Decal and Pattern layers assigned
to non-overlaying channels

Common Hyperion stripe set on
Decal layer (Decal always goes on top)

Dahl pattern used as a decal
Pattern textures are automatically repeated

As for everything else, you'll have to go through a lot of trial and error until you get these just right like you want them. Again, you can exec skin mods in real time so no need to reload your game every time you change a number - which is really nice and saves a ton of time.

And this concludes Vectors! Onto our last parameters category:

6 - ScalarParameterValues

So here we are, the last batch of parameters.
These are mostly going to be used to tweak and fix things, but you generally won't spend a lot of time in here. ScalarParameterValues are parameters that accept a *single value input*, so no RGB shenanigans going on here: each parameter here has one single number per entry that does stuff and that's it.

Let's look at these, shall we?

— **p_HighlightsIntensity**

— **p_ShadowsIntensity** —Here's what they look like in text form:

(ParameterName="p_HighlightsIntensity",ParameterValue=0.450000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))
(ParameterName="p_ShadowsIntensity",ParameterValue=1.000000,ExpressionGUID=(A=437293753,B=1205147708,C=-775723903,D=1480014964))

—Here's what they do:

If added, these two can be used to tweak your highlights and shadows. A lower HighlightIntensity value can fix unwanted 'glow' if your colors are generally too bright, and a higher ShadowIntensity value can give them more depth and contrast. These are a last-minute kind of tweak so don't worry too much about them, you'll likely never even have to touch them.

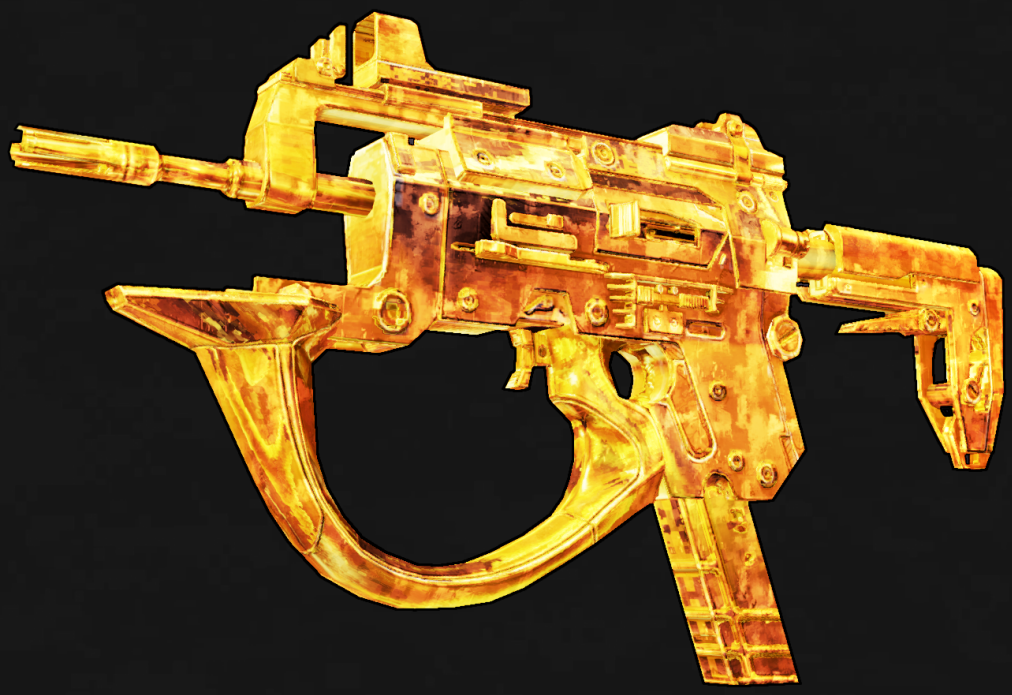
— **p_ReflectColorScale** —Here's what it looks like in text form:

(ParameterName="p_ReflectColorScale",ParameterValue=1.000000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))

—Here's what it does:

If added, this value changes 'how shiny' your reflective layer will be. *IMPORTANT:* this thing is *VERY* sensitive and can get really nasty really quick. You'll likely never have to touch this value or just leave it at a flat 1.000000. Don't go over 1. Please.
And since we're concluding our last puzzle piece about making things shiny, here's a quick PSA:

Also,



Do not do this. Nobody likes this. This is not ok.

See this? Don't do this. This is very ugly and literally not a single person on the entire planet thinks this is cool. I mean sure you can make gold stuff but please, PLEASE add in some metal or some black accents. Please. Please? Thank you.

Moving on.

— **p_UseFullColorDecal** —Here's what it looks like in text form:

(ParameterName="p_UseFullColorDecal",ParameterValue=1.000000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))

—Here's what it does:

So this is a bit of a specific thing. Some decals natively have multiple colors, and setting this value to 1.000000 will enabled them. However, this is an *extreme case-by-case* situation, as in, basically every decal has unique properties and changing this one single number to a one or a zero could potentially totally break everything and make everything look terrible because, well, perhaps that one single decal you chose wasn't supposed to have full color enabled.
On the other hand however, changing this value could also potentially fix everything if stuff looks awful! In other words, try and change it and see what happens. If stuff looks fine, then good job, leave this thing alone and move on.

— **p_DecalRotate** —Here's what it looks like in text form:

(ParameterName="p_DecalRotate",ParameterValue=0.000000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))

—Here's what it does:

I bet you can't figure out what this does.
Yup, it rotates your decal. I know, it's pretty incredible technology.
Literally the only thing you need to know about this is that a 1.0 increment here makes it do a full rotation. Wanna have your decal do half rotation? Set this value to 0.5! Amazing!

— **p_ReplaceDecal**

— **p_ReplacePattern** —Here's what they look like in text form:

(ParameterName="p_ReplaceDecal",ParameterValue=0.000000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))
(ParameterName="p_ReplacePattern",ParameterValue=0.000000,ExpressionGUID=(A=437293753,B=1205147708,C=-775723903,D=1480014964))

—Here's what they do:

I grouped them together because they kinda do the same thing: setting these two values to one or zero will change how your decal and pattern are overlayed, and by that I mean that simply *some* decals and *some* patterns look better with either a one or a zero in here, nothing else. So in other words, if your pattern looks totally lame try slapping a one or a zero in here and see if that does anything.
If it still looks totally lame then sorry, you probably chose something gross like any of the Bandit patterns anyway so I mean, that's your problem dude.

— **p_p_EmissiveScale** —Here's what it looks like in text form:

(ParameterName="p_EmissiveScale",ParameterValue=1.000000,ExpressionGUID=(A=-1257568432,B=1277066486,C=-723473993,D=-1144384173))

—Here's what it does:

So here it is, our last parameter. This is very simple: this value changes the intensity of your elemental glow, where 1.000000 represents full intensity and 0.000000 turns it off completely. Setting this to zero is different than simply changing your glow color to black because in case your gun has an element, even if you set its native color to black elemental colors will still overwrite it. Here however, setting this value to zero will turn off *ANY* glow, including elemental colors.
I generally wouldn't recommend turning this off every time because it can make it tricky to tell which element your gun is unless you look at the card or -obviously- shoot it, however it can be really awesome if you make multi-elements guns and simply do not want just one glow color to be shown on the model. Also, some skins just straight up look dope without any glow on them, so do play around with this, you might run into some super cool combination thanks to this thing alone.

And there you have it, this concludes our parameters list.

Now, one last recap on resources and a couple final words and we're done. Yes I'm aware we're at an awkward spot on the page and starting the final conclusive chapter here would be super weird so...

I'll just

do

this:

ʔ•ꈊ•ʔ

7 - Conclusions

So here you have it, this is pretty much everything you need to know. Always remember that like any other artistic thing, the best way to go about it is to not be afraid of experimenting. Try out ideas even if they sound wacky and stupid - you might accidentally make something totally amazing. And most importantly, don't give up if stuff isn't exactly working out. Art requires patience and dedication. (:

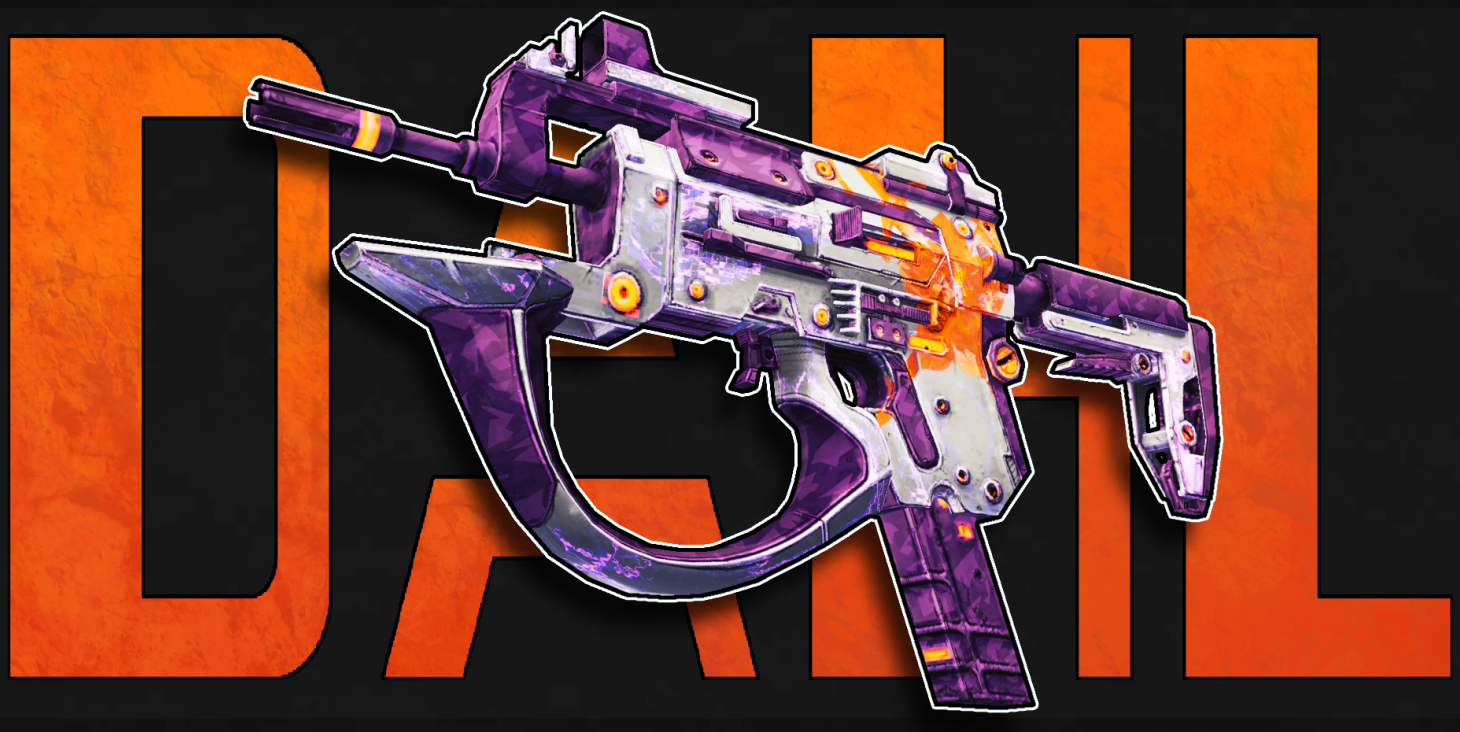
I'd like to personally thank you if you've made it this far. We're currently on page 16, can you believe it? I originally just wanted to make a quick guide after figuring these stuff out and now look at what it has become.
But hey, if even a single person finds value in this and uses it to make something dope then I'll consider it a success. After all, the whole point of this was to help people figure out this stuff so they could make personalized skins for their own guns instead of using the vanilla ones we've seen a billion times.

Before I forget, once again all the resources listed throughout this guide (the master blank file, textures IDs, guns IDs, etc.) can also be found in my *Resources* folder located inside my GitHub folder in the BL2 community directory: **[HERE]**.

So that's it. Now go make some cool shit.

-Tofu

~ THE END ~



Thanks for reading (:

Rabbit loves you.