

Installing the ExtendedExhaust specialization:

The first thing you need to do is copy the contents of this mod into the root folder of your existing mod.

The image shows two screenshots of a Windows File Explorer window. The top screenshot shows the 'Downloads' folder containing a subfolder named 'FS19_ExtendedExhaustSDK'. This subfolder contains three subfolders: '_sdk', 'exhaust', and 'scripts'. A table to the right of the folder list provides the following details:

Name	Date modified	Type
_sdk	12/20/2020 9:24 PM	File folder
exhaust	12/17/2020 4:34 AM	File folder
scripts	12/16/2020 3:20 AM	File folder

The bottom screenshot shows the 'FarmingSimulator2019 > mods > FS19_claasTorion1914DevMule' folder. A table to the right of the file list provides the following details:

Name	Date modified	Type	Size
<input checked="" type="checkbox"/> _sdk	12/20/2020 9:31 PM	File folder	
<input checked="" type="checkbox"/> exhaust	12/20/2020 9:31 PM	File folder	
<input checked="" type="checkbox"/> scripts	12/20/2020 9:31 PM	File folder	
sounds	12/20/2020 1:25 PM	File folder	
vehicles	12/20/2020 1:25 PM	File folder	
brand_claas.dds	7/24/2019 2:00 PM	IrfanView DDS File	33 KB
icon_claasTorion1914DevMule.dds	7/7/2019 2:07 PM	IrfanView DDS File	33 KB
modDesc.xml	12/20/2020 1:32 PM	XML Source File	2 KB

Next, you need to open your modDesc.xml file. Here you will be adding the specialization to the game and either creating a new vehicle type or adding it to an existing vehicle type. Some vehicles already have these entries in their modDesc files, more on this below.

If your vehicle already has a <specializations> section inside of it, simply add the highlighted line to the existing section.

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<modDesc descVersion="43">
  <author>GIANTS Software</author>
  <version>1.0.0.0</version>
  <multiplayer supported="true" />
  <title>
    <en>CLAAS TORION 1914 Dev Mule</en>
  </title>
  <description>
    <en>CLAAS TORION 1914 Dev Mule</en>
  </description>
  <iconFilename>icon_claasTorion1914DevMule.png</iconFilename>
  <specializations>
    <specialization name="extendedExhaust" className="ExtendedExhaust" filename="scripts/ExtendedExhaust.lua" />
  </specializations>
  <storeItems>
    <storeItem xmlFilename="vehicles/claas/torion1914DevMule/torion1914DevMule.xml"/>
  </storeItems>
  <brands>
    <brand name="CLAAS" title="CLAAS" image="brand_claas.dds"/>
  </brands>
  <brandColors>
    <color name="CLAAS_GREEN1" value="0.2384 0.3186 0.0116 1" />
    <color name="CLAAS_RED1" value="0.5310 0.0357 0.0112 1" />
    <color name="CLAAS_WHITE1" value="0.784 0.784 0.784 1" />
    <color name="CLAAS_DARKGREY1" value="0.0721 0.0765 0.0906 1" />
    <color name="CLAAS_DARKGREY2" value="0.068 0.068 0.068 1" />
  </brandColors>
</modDesc>
```

After that, you want to create/add a new vehicle type. This is also done one of two ways, depending on whether or not your vehicle already has a <vehicleTypes> section or not.

If your vehicle does **NOT** have a <vehicleTypes> section in the modDesc.xml, you will first want to open your vehicle xml file and take note of its vehicle type. This can be found at the top of that file.

```
1 <?xml version="1.0" encoding="utf-8" standalone="no" ?>
2 <vehicle type="tractor">
3   <annotation>Copyright (C) GIANTS Software GmbH, All Rights Reserved.</annotation>
4
5   <storeData>
6     <name>TORION 1914 Dev Mule</name>
7     <specs>
```

Now, copy the <vehicleTypes> section from modDescEntries.xml found in the _sdk folder to your modDesc xml file. Change OLD_VEHICLE_TYPE to the vehicle type you noted above. Change NEW_VEHICLE_TYPE to whatever name you want. Just remember the name you chose. I used "tractorXT" for this example.

If your vehicle already has a <vehicleTypes> section inside of modDesc.xml, you simply add the highlighted line to the existing vehicle type. You can also **SKIP** the next section.

```
1 <?xml version="1.0" encoding="utf-8" standalone="no" ?>
2 <modDesc descVersion="43">
3   <author>GIANTS Software</author>
4   <version>1.0.0.0</version>
5   <multiplayer supported="true" />
6   <title>
7     <en>CLAAS TORION 1914 Dev Mule</en>
8   </title>
9
10  <description>
11    <en>CLAAS TORION 1914 Dev Mule</en>
12  </description>
13
14  <iconFilename>icon_claasTorion1914DevMule.png</iconFilename>
15
16  <specializations>
17    <specialization name="extendedExhaust" className="ExtendedExhaust" filename="scripts/ExtendedExhaust.lua" />
18  </specializations>
19
20  <vehicleTypes>
21    <type name="tractorXT" parent="tractor" filename="$data5/scripts/vehicles/Vehicle.lua">
22      <specialization name="extendedExhaust"/>
23    </type>
24  </vehicleTypes>
25
26  <storeItems>
27    <storeItem xmlFilename="vehicles/claas/torion1914DevMule/torion1914DevMule.xml"/>
28  </storeItems>
29
30  <brands>
31    <brand name="CLAAS" title="CLAAS" image="brand_claas.dds"/>
32  </brands>
33
34  <brandColors>
35    <color name="CLAAS_GREEN1" value="0.2384 0.3186 0.0116 1" />
36    <color name="CLAAS_RED1" value="0.5310 0.0357 0.0112 1" />
37    <color name="CLAAS_WHITE1" value="0.784 0.784 0.784 1" />
38    <color name="CLAAS_DARKGREY1" value="0.0721 0.0765 0.0906 1" />
39    <color name="CLAAS_DARKGREY2" value="0.068 0.068 0.068 1" />
40  </brandColors>
41 </modDesc>
42
```

The next step is to change the vehicle type of your vehicle to the name you created above. Remember this step is **NOT** necessary if your vehicle already has a custom type specified in your modDesc.xml.

Back inside of your vehicle xml file, again at the top, change the vehicle type. Since I used "tractorXT" in the above example, this is what I change this value to.

```
1 <?xml version="1.0" encoding="utf-8" standalone="no" ?>
2 <vehicle type="tractorXT">
3   <annotation>Copyright (C) GIANTS Software GmbH, All Rights Reserved.</annotation>
4
5   <storeData>
6     <name>TORION 1914 Dev Mule</name>
7     <specs>
8       <power>224</power>
9       <maxSpeed>40</maxSpeed>
```

Some vehicles have a <vehicleTypeConfigurations> section inside of their vehicle xml file. If it does, you need to change the vehicleType entries there to the same name you just set above. This is **ONLY** for vehicles that have this section.

```
76 <vehicleTypeConfigurations title="Exhausts">
77   <vehicleTypeConfiguration name="Muffler W/ AirCleaner" price="0" vehicleType="tractorXT">
78     <objectChange node="0>22|0" visibilityActive="true"/>
79   </vehicleTypeConfiguration>
80   <vehicleTypeConfiguration name="Muffler W/ Short Cleaner" price="0" vehicleType="tractorXT">
81     <objectChange node="0>22|1" visibilityActive="true" />
82     <objectChange node="0>22|0" visibilityActive="false" />
83   </vehicleTypeConfiguration>
84   <vehicleTypeConfiguration name="Straight W/ Turbo II" price="0" vehicleType="tractorXT">
85     <objectChange node="0>22|2" visibilityActive="true" />
86     <objectChange node="0>22|0" visibilityActive="false" />
87   </vehicleTypeConfiguration>
88   <vehicleTypeConfiguration name="Muffler" price="0" vehicleType="tractorXT">
89     <objectChange node="0>22|3" visibilityActive="true" />
90     <objectChange node="0>22|0" visibilityActive="false" />
91     <objectChange node="0>14|16" visibilityActive="true" />
92   </vehicleTypeConfiguration>
93 </vehicleTypeConfigurations>
94
```

Now you need to, again in your vehicle xml file, find the <motorized> section. At the bottom of that section, you will usually find your <exhaustEffect> entries.

```
371         <transmission minForwardGearRatio="19.2" maxForwardGearRatio="300" minBackwardGearRatio
372         </motorConfiguration>
373     </motorConfigurations>
374
375     <exhaustEffects>
376         <exhaustEffect node="exhaustParticle1" filename="$data/particleSystems/shared/exhaust.i3d"
377     </exhaustEffects>
378
379     <motorStartDuration>2750</motorStartDuration>
380
```

Copy/paste one of the <extendedExhaust> variants from the vehicleEntries.xml file found in the _sdk folder to your vehicle xml file. Do this under the <exhaustEffect> section.

```
    </motorConfiguration>
</motorConfigurations>

<exhaustEffects>
    <exhaustEffect node="exhaustParticle1" filename="$data/particleSystems/shared/exhaust.i3d" minRpmColor="0.9 0.9 0.9 0.1" maxRpmColor="0
</exhaustEffects>

<extendedExhaust motorStartDuration="1000">
    <particleSystems>
        <particleSystem node="1>0|8" gain="0.00075" speed="0.001" normal="0.5" tangent="0.25" lifespan="4000" file="exhaust/exhaust.i3d">
            <startInfo emitScale="4.0" emitSpeed="0.004" tangent="0.1"/>
            <loadInfo emitScale="5.0" emitMin="0.75"/>
            <rpmInfo useThrottle="true">
                <speedDamping forward="0.5" reverse="0.25"/>
            </rpmInfo>
        </particleSystem>
    </particleSystems>
</extendedExhaust>
```

Note in the above example, the original exhaustEffect node is "exhaustParticle1". At present you **CAN NOT** use the name of a node for extended particles. For whatever reason, the vanilla particle loading code does not recognize it. To find the node index, scroll down to the <i3dMappings> section of your vehicle xml and find "exhaustParticle1". You could also search for this with Notepad++ or something similar.

```
741     <i3dMapping id="fuelNeedle" node="1>0|5|7|5|0" />
742     <i3dMapping id="tempNeedle" node="1>0|5|7|6|0" />
743     <i3dMapping id="interiorScreenLight" node="1>0|5|8" />
744     <i3dMapping id="articulatedAxisAiReverserNode" node="1>0|6" />
745     <i3dMapping id="exitPoint" node="1>0|7" />
746     <i3dMapping id="exhaustParticle1" node="1>0|8" />
747     <i3dMapping id="axisJoint" node="1>0|9" />
748     <i3dMapping id="exterior_decals" node="1>0|10|9" />
749     <i3dMapping id="exterior_decals3" node="1>0|10|14" />
750     <i3dMapping id="exterior_decals4" node="1>0|10|15" />
```

In the above image, the actual index for "exhaustParticle1" is "1>0|8". The index is the value you want to use for the node entry of your new particleSystem (shown above).

Some vehicles have multiple exhaust effects by default. To add more particle systems, copy the <particleSystem> lines from the first <particleSystem> section as many times as you like. Change each node to the entry of the corresponding <exhaustEffect>.

The following is an example of a vehicle with multiple exhaust nodes:

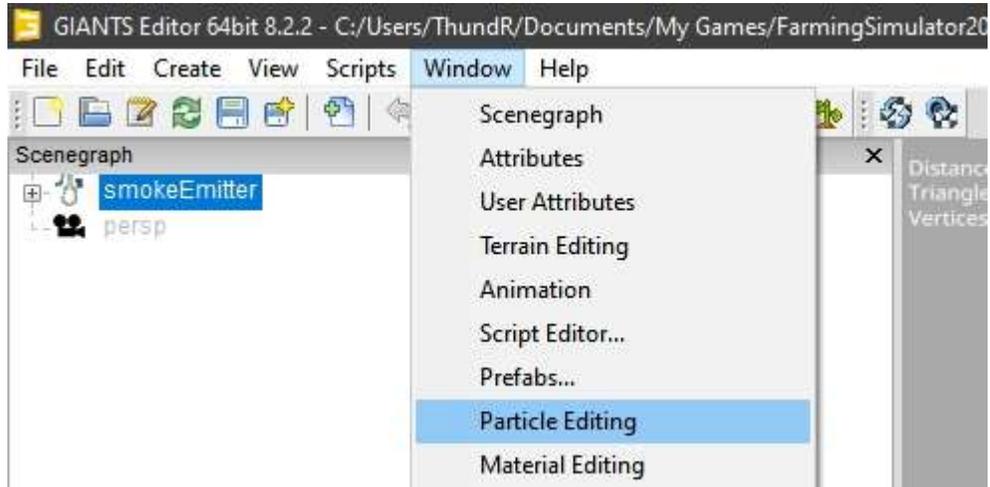
```
<extendedExhaust motorStartDuration="1000">
  <particleSystems>
    <particleSystem node="0>0|12|0" gain="0.00075" speed="0.001" normal="0.5" tangent="0.25" lifespan="4000" file="exhaust/rollCoal.i3d">
      <startInfo emitScale="1.0" emitSpeed="0.004" tangent="0.1"/>
      <loadInfo emitScale="1.5" emitMin="0.80"/>
      <rpmInfo emitScale="2.5" emitMin="0.5" emitMax="0.85" useThrottle="true">
        <speedDamping forward="0.35"/>
      </rpmInfo>
    </particleSystem>
    <particleSystem node="0>0|12|1" gain="0.00075" speed="0.001" normal="0.5" tangent="0.25" lifespan="4000" file="exhaust/rollCoal.i3d">
      <startInfo emitScale="1.0" emitSpeed="0.004" tangent="0.1"/>
      <loadInfo emitScale="1.5" emitMin="0.80"/>
      <rpmInfo emitScale="2.5" emitMin="0.5" emitMax="0.85" useThrottle="true">
        <speedDamping forward="0.35"/>
      </rpmInfo>
    </particleSystem>
  </particleSystems>
</extendedExhaust>
```

If you want to remove the default exhaust effects from the vehicle, you can simply comment out the original <exhaustEffects> section as follows:

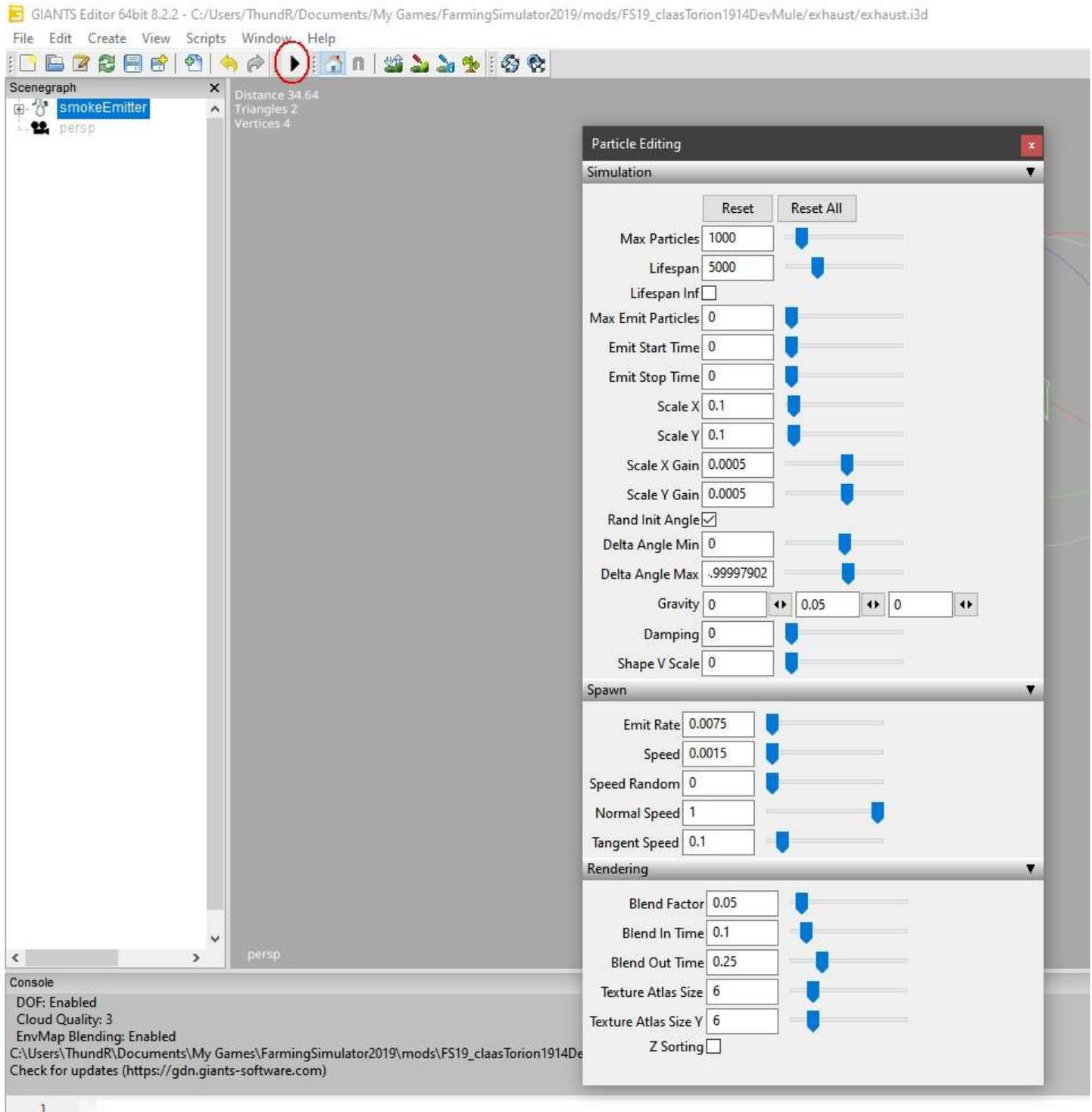
```
</motorConfiguration>
</motorConfigurations>
<!--
<exhaustEffects>
  <exhaustEffect node="exhaustParticle1" filename="$data/particleSystems/shared/exhaust.i3d" minRpmColor="0.9 0.9 0.9 0.1" maxRpmColor="0
-->
<extendedExhaust motorStartDuration="1000">
  <particleSystems>
    <particleSystem node="1>0|8" gain="0.00075" speed="0.001" normal="0.5" tangent="0.25" lifespan="4000" file="exhaust/exhaust.i3d">
      <startInfo emitScale="4.0" emitSpeed="0.004" tangent="0.1"/>
      <loadInfo emitScale="5.0" emitMin="0.75"/>
      <rpmInfo useThrottle="true">
        <speedDamping forward="0.5" reverse="0.25"/>
      </rpmInfo>
    </particleSystem>
  </particleSystems>
</extendedExhaust>
```

Adjusting the particle i3d for your vehicle:

To adjust the particle effect, open exhaust.i3d or rollCoal.i3d located in the exhaust folder of your mod. Inside of GE, select “smokeEmitter” and open the particle editor.



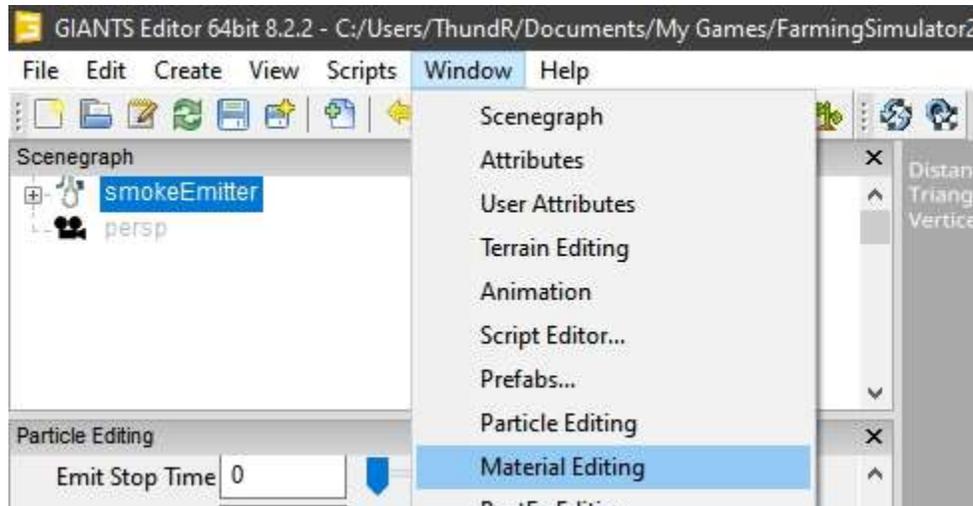
This will open the particle editing window. You can also hit the play button to view your smoke effect real time.



The main values you can set are described below:

- * Max Particles refers to how many “particles” can be emitted at once. Seems to be mostly a performance tweak. For example, having a high emit rate and lifespan with low max particles will make the exhaust effect appear to stop and restart. This is due to running out of particles part way through the animation.
- * Lifespan is simply the emit time in milliseconds. Larger values make the smoke “linger” for a longer amount of time but could affect performance. This can be set in the vehicle xml file as well.
- * Scale X and Scale Y are the size of the particle at the base or start. This value can be set in the vehicle xml file as well.
- * Scale X Gain and Scale Y Gain are the amount the particle spreads out as it travels or how wide it becomes. This value can be set in the vehicle xml file as well.
- * Gravity is how heavy the smoke is. The higher the number, the quicker the smoke will lift. Setting this to zero basically creates a straight line effect.
- * Emit Rate is how “thick” the smoke appears. This value can be set in the vehicle xml file as well. In addition to the initial value, you can also set rates for motor startup, load, and rpm.
- * Speed is how fast the smoke travels. This can also be adjusted inside of the vehicle xml. You can adjust initial rate, as well as the rate on motor startup.
- * Normal Speed works in conjunction with speed. I am not quite sure how to explain this one. While speed is the rate at which the particles emit, normal speed is more of a scaling factor, working on the normal map of the particle. The range is 0 to 1. This value can be set in the vehicle xml file as well.
- * Tangent Speed is how much the smoke “dissipates” as it moves. Higher speed values increase the effect. This can be adjusted in the vehicle xml as well.
- * Damping causes the smoke to “bunch up” after a period of time. The higher the number, the sooner it happens. This must be set in the i3d file.
- * Shape V Scale appears to make the smoke have less friction. In other words, the smoke will “chase” you more at higher values. It will also travel more easily in the direction it was “thrown”. Looks decent at high speed but does make the exhaust effect appear choppy if set too high. This also must be set in the i3d file.
- * Blend Factor is the value most people will want to set. This determines how “opaque” or black the smoke is. I could not find a way to adjust this via lua, so you will have to set the base darkness of your smoke effect inside of the i3d file.
- * Blend In Time is how long it takes (in percentage of lifespan) that the particle takes to fully blend in. Used for making the exhaust effect appear to “start” further from the actual pipe. This also must be set in the i3d file.
- * Blend Out Time is the point (in percentage of lifespan) that the smoke begins to fade out. This also must be set in the i3d file.

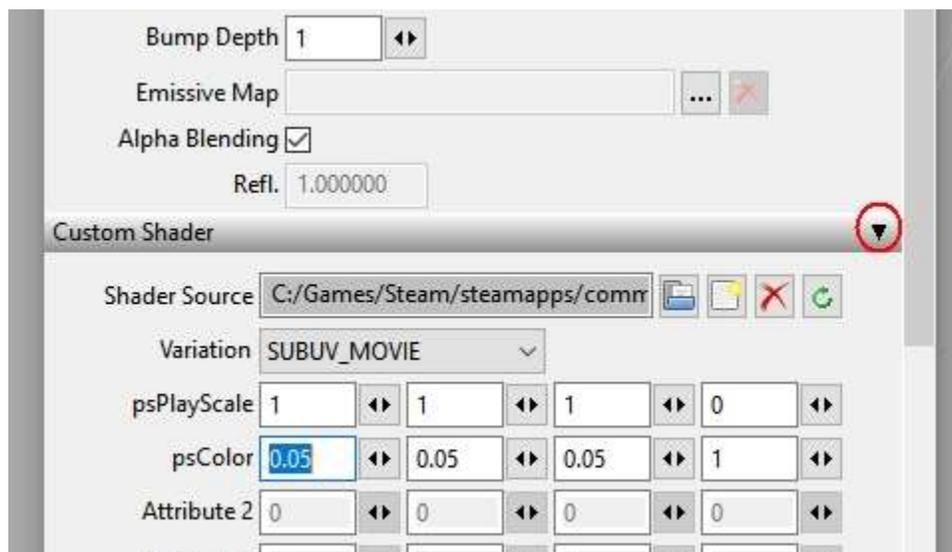
You can also change the color of the smoke by opening the Material Editor in GE.



Here you need to scroll to the bottom of the material editing window, click the little arrow on the right side of the “Custom Shader” section, and change the first three values under “psColor”. The values translate to the R, G, and B values of the smoke in percentage format. The last value is A or “alpha”, which is normally the transparency. 0 is invisible while 1 is opaque. For this shader, however, that value is not used. Simply leave it as 1.

To convert 0-255 values to percentage values, just take the value you want and divide by 255. For example, 10, 10, 10 in RGB would translate to approximately 0.04, 0.04, 0.04 in percentage format. White smoke would be 1, 1, 1 while black smoke would be 0, 0, 0 in other words.

Also note, in the below image, Variation is set to SUBUV_MOVIE. The different variations determine how the psPlayScale values are used. Currently, I leave those values set to what you see below and use the SUBUV_BY_LIFE variant instead. Really, you could even select “none” if you wanted. I couldn’t find any way to set those values that looked good to me, so I just left them as 1.



Editing the vehicle xml ExtendedExhaust section:

You can set your vehicle up a variety of different ways via xml. I have included 5 different variants in the vehicleEntries.xml file, located in the _sdk folder. There are two exhaust i3d files as well. One is for a more “normal” exhaust effect, while the other is darker and designed for more of a “roll coal” effect.

Below is a vehicle xml entry with all of the different values shown, as well as their default values:

```
<extendedExhaust motorStartDuration="1000">
  <particleSystems>
    <particleSystem node="1>0|8" scale="0.25" emitScale="1.0" gain="0.001" speed="0.001" normal="0.5" tangent="0.25" lifespan="4000" file="exhaust/exhaust.i3d">
      <startInfo emitScale="0" emitSpeed="0.001" tangent="0.25"/>
      <loadInfo emitScale="0" emitMin="0" emitMax="1.0"/>
      <rpmInfo emitScale="0" emitMit="0" emitMax="1.0" useThrottle="false">
        <speedDamping forward="0" reverse="0"/>
      </rpmInfo>
    </particleSystem>
  </particleSystems>
</extendedExhaust>
```

* **motorStartDuration** is the amount of time in milliseconds the *startInfo* section plays for when the motor first starts. You can use the startInfo section to create a darker puff of exhaust on startup, for example. The default value for this is 0, meaning no startup effects will be used, regardless of what is in startInfo.

The <particleSystem> line is where you set all of the initial values for your exhaust effect. Not all i3d values are available, but most of them are.

* **scale** is the size of the smoke particle at the start of the animation or at the pipe. Sets the initial *Scale X and Scale Y* for the particle. This defaults to the value set in the particle i3d file.

* **emitScale** is the default multiplier used for determining Emit Rate (see above). This defaults to 1.0.

* **gain** is the size of the smoke particles as they travel along the animation. Makes the smoke effect appear wider as it travels. Sets the initial *Scale Gain X and Scale Gain Y* for the particle (see above). This defaults to the value set in the particle i3d file.

* **speed** is the rate at which the particles travel. Also affects how pronounced things like damping and tangent speed are. Sets the initial *Speed* value for the particle (see above). This defaults to the value set in the particle i3d file.

* **normal** is a speed factor (described above). Sets the initial *Normal Speed* value for the particle (see above). This defaults to the value set in the i3d file.

* **tangent** determines how much the smoke dissipates. Sets the initial *Tangent Speed* value for the particle (see above). This defaults to the value set in the particle i3d file.

* **lifespan** determines the time the animation plays in milliseconds. Higher numbers mean longer trails of smoke. Sets the initial *Lifespan* value for the particle (see above). This defaults to the value set in the particle i3d file.

* **file** is the relative path to the particle i3d file.

The <startInfo> section contains the values used when the motor is first starting and lasts until *motorStartDuration* is finished. This entire section and any of its variables can be omitted.

* **emitScale** is a multiplier to the default emit scale for when the engine is starting. Defaults to 0 if omitted, meaning no startup effect will be displayed.

* **emitSpeed** is the speed the particles emit while the engine is starting. Uses the default emission rate if omitted.

* **tangent** is the dispersion rate while the engine is starting. Sometimes needed to make the initial puff of smoke look more realistic. Uses the default tangent speed if omitted.

The <loadInfo> section contains the values used when the motor is under load. This entire section and any of its variables can be omitted.

* **emitScale** is a multiplier to the default emit scale when the motor is under load. Emission begins at default rate when motor load percentage reaches **emitMin** and increases to the emitScale value when motor load percentage reaches **emitMax**. Defaults to 0 when omitted, meaning no motor load effects will be displayed.

* **emitMin/emitMax** values are described above. Smoke begins emitting when motor load percentage reaches emitMin and emits at the value set to **emitScale** when motor load percentage reaches emitMax. These values default to 0 and 1 respectively if omitted, meaning smoke begins emitting at 0% load and reaches max emission at 100% load.

The <rpmInfo> section contains the values used when motor rpm percentage increases. Can optionally be linked to throttle pressure. The rpm value used by the specialization can be lessened at higher speeds by with the <speedDamping> sub section. Can also be used with the <loadInfo> section to have throttle based exhaust in tandem with motor load (explained below). This entire section and any of its variables can be omitted.

* **emitScale** works just like the motor load version, except it is based on percentage of rpm instead of percentage of motor load. Defaults to 0 if omitted, meaning no rpm based effects will be displayed (with some exceptions, see below).

* **emitMin/emitMax** values work just like the motor load version, except they are based on motor rpm percentage instead of motor load percentage. If **useThrottle** is set to true, rpm is modified by throttle pressure. For example, letting off the throttle in game does not always lower the rpm. With useThrottle, rpm would be 0 if the throttle is not pressed, for example. These default to 0 and 1 respectively if omitted, meaning smoke begins emitting at 0% rpm and reaches max emission at 100% rpm.

* **useThrottle** is a special flag that serves a couple purposes. The main one is to add a throttle effect to the rpm value used by the specialization. As described above, simulates “hitting the gas” for vehicles that work that way. This value defaults to false if omitted, meaning the actual rpm will be used regardless of throttle pressure.

The <speedDamping> sub section of <rpmInfo> is for damping the rpm at higher speeds. This can be used to lessen the smoke effect when travelling at high speed, for example. The two values, *forward* and *reverse* are used to set the damping for when the vehicle is travelling in that direction. This entire section and its variables can be omitted.

* **forward** this is the speed damping effect while moving forward. Setting this value to 0 means no damping, and full rpm is used regardless of speed. Setting this value to 1 would be the equivalent to full rpm at idle and 0 rpm at max speed. This defaults to 0 if omitted, meaning no damping effect while travelling forward.

* **reverse** is the same as forward, except for when the vehicle is travelling backwards. This also defaults to 0 if omitted, meaning no damping effect while travelling in reverse.

The value set to forward or reverse is the percentage that is deducted from rpm percentage at maximum speed. Setting one of these to 0.25 would, for example, result in the calculated rpm being 75% of the actual value at maximum speed (100% rpm – 25% speed = 75% rpm).

Setting forward or reverse to a value **greater** than 1 results in dropping the calculated rpm percentage to 0 **before** reaching maximum speed. For example, setting forward to 2 would drop the calculated rpm percentage to 0 at 50% speed (100% rpm – (50% speed * 2) = 0% rpm). You could use something like this to make smoke stop appearing (or minimally appear if *emitMin* is set to 0) at lower speeds regardless of rpm.

The speedDamping section of rpmInfo, along with *useThrottle* can also be used with the <loadInfo> section to create a throttle based motor load setup. This only works if the *emitScale* value in <rpmInfo> is omitted or set to 0. In this setup, smoke will display according to motor load as usual. However, when you hit the throttle, the calculated load is increased. This effect can be lessened at higher speeds using speed damping.

I added this feature to compensate for the odd ways the internal rpm and motor load values behave in a vanilla game sometimes. For example, in a vanilla game motor load sometimes takes a long time to increase even though you have the gas pressed all the way to the floor. It could just be the way the vehicles I was using are set up, but it did not seem right to me. So I added these optional config values to adjust for that. It may seem confusing at first, but its easy to pick up once you mess with the values a little.

One last thing to mention. If both <rpmInfo> and <loadInfo> values are specified, the section currently emitting the most smoke is used. For example, if motor load is emitting at a scale of 2.5 while rpm is emitting at scale of 1.2, the emission scale would be 2.5 (motor load value).

Looking at the different variants in the vehicleEntries.xml file in the `_sdk` folder will show you how these values can be used.

There are also two i3d files for exhaust. The first, exhaust.i3d uses a more transparent texture and is for a more “natural” exhaust type effect. The second file, rollCoal.i3d uses a less transparent texture and a higher blend rate for emitting bigger darker clouds of smoke.

You can use any particle effect you want. All you have to do is point to it in the <particleSystem> line **file** variable of your vehicle xml file. I have been messing with the particle systems for a while, but I still have some learning to do. The ones included with the sdk are the best I could come up with for the time being.

When you are done, save your vehicle xml and you should be all set! You can delete the _sdk folder if you want, or you can leave it there for future reference if you like as well. Should not hurt anything by being there.

