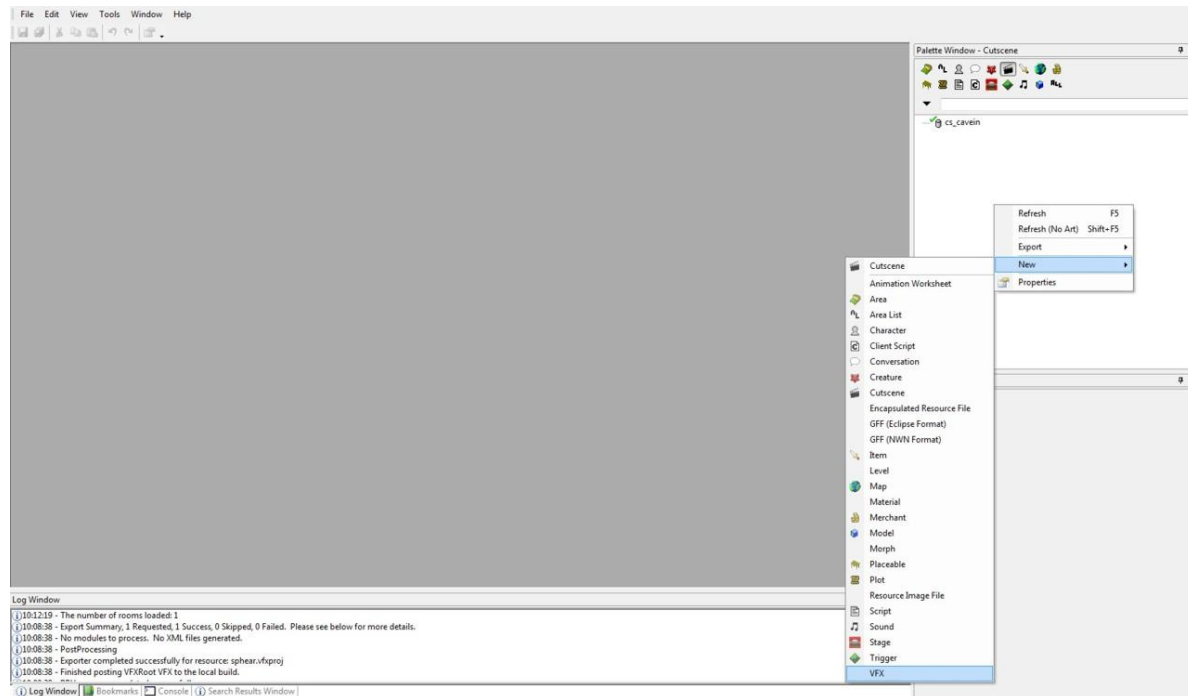


## VFX Tutorial #1 version 1.2 – The Basics

This tutorial is to get you started in creating your own visual effects to be used in the DA toolset. Tutorial #1 will cover the basics of creating, editing, and implementing the visual effect.

To create a new VFX, right click on the palette window and select “NEW > VFX”



The VFX Editor will open up showing a square grid and a “Select VFX” window in the middle. For this tutorial we will be doing a Placeable effect, so select “Placeable” and click “OK”.

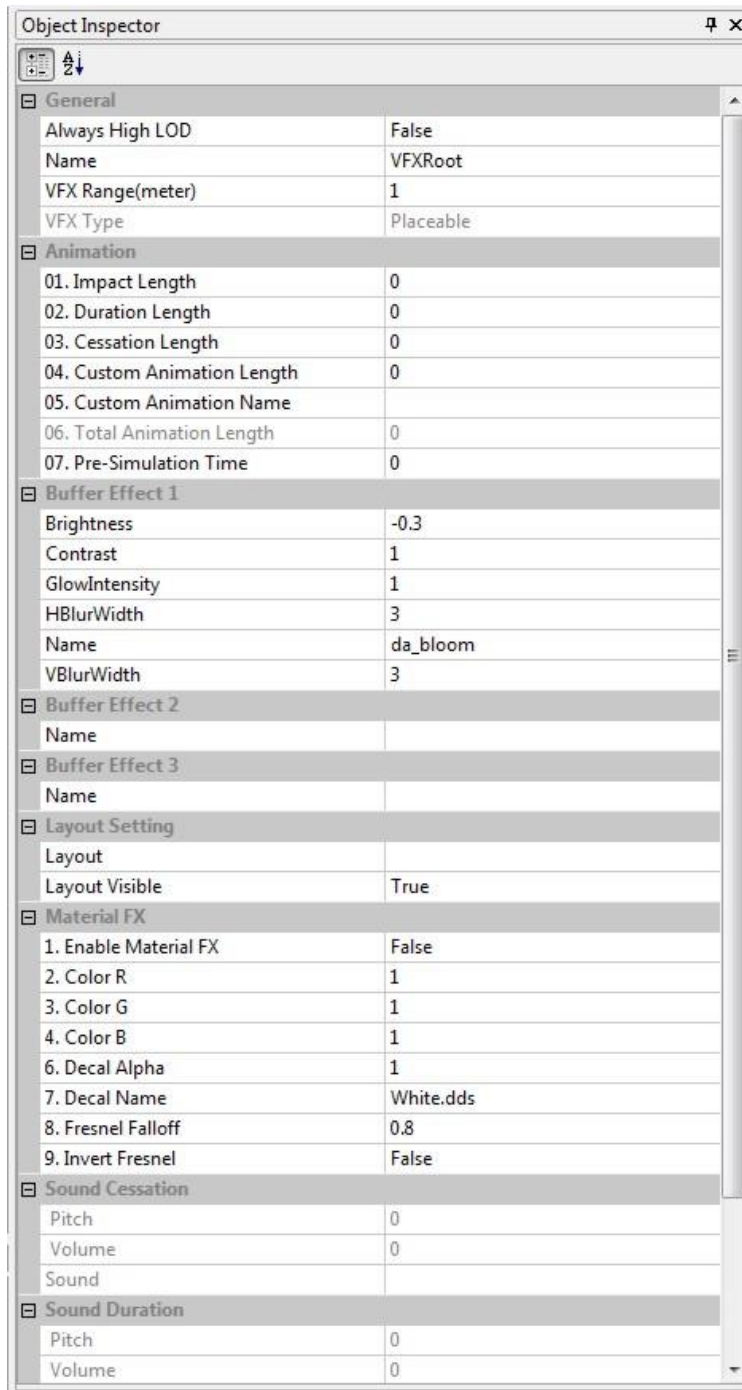
On the top left, you should see a little globe called “VFXRoot” and in the Object Inspector you should see the properties of VFXRoot.

At this point, save your work by selecting “File > Save As” or clicking the Save Icon in the upper left. You will be prompted for a file name. I like to keep all of my VFX files in one place so in My Documents/Bioware/Dragon Age folder I created a VFX folder to save my files.

For this tutorial we will be creating a Bon Fire type of effect and to stay with Dragon Age naming convention, we will use a 3 letter prefix, underscore, and then the file name. Name the file “fxp\_bonfire\_jp”

The “fxp” is for “Visual Effect Placeable” (all VFX files start with fx\*) “bonfire” tells me what the effect is, and the “jp” on the end lets me know that it is my effect and not someone else’s, you can add whatever suffix you want. Click “Save” when done.

You can close the toolset at any time and return to this tutorial. You can open your effect up again later by selecting “File > Open” and navigate to and select the file you just saved.



**VFXRoot Properties:** I'll just cover the ones that we will be using.

**Name:** This is the resource name and is what will be used when we Post to Local. Change it to match the file name.

**Impact Length:** Time in seconds for the stating phase of the effect. Example, a Runner starting off from the start line reaching max speed.

**Duration Length:** Time in seconds for the duration of the effect. Example, the time it takes the runner to reach the finish line after reaching max speed.

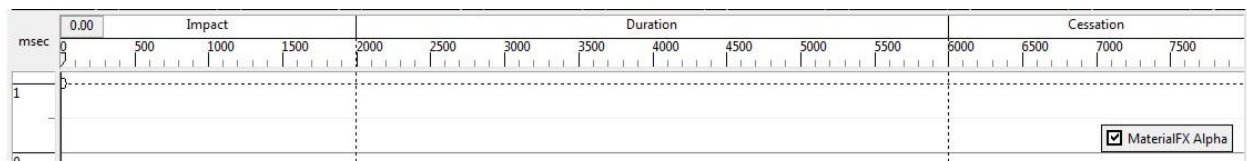
**Cessation Length:** Time in seconds for the ending of the effect. Example, the runner slowing down after crossing the finish line.

We are making a bon fire effect, so we want the fire to start off small, grow in size, then burn it's self out. So set the Impact Length to 3 seconds, this will start the fire and grown to it's max size. Set the Duration Length to 3 seconds. This will allow us plenty of time to vary the coloration of the flames. Set the Cessation time to 3 seconds as well, as the fire burns out.

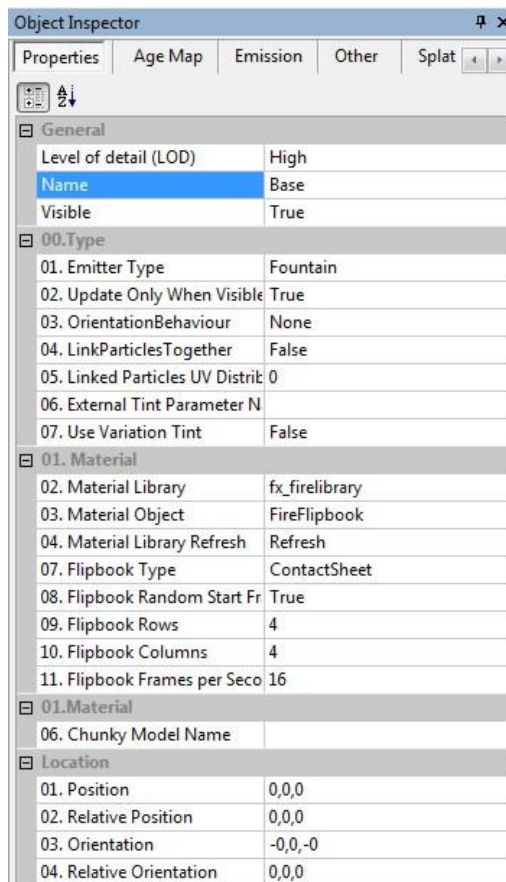
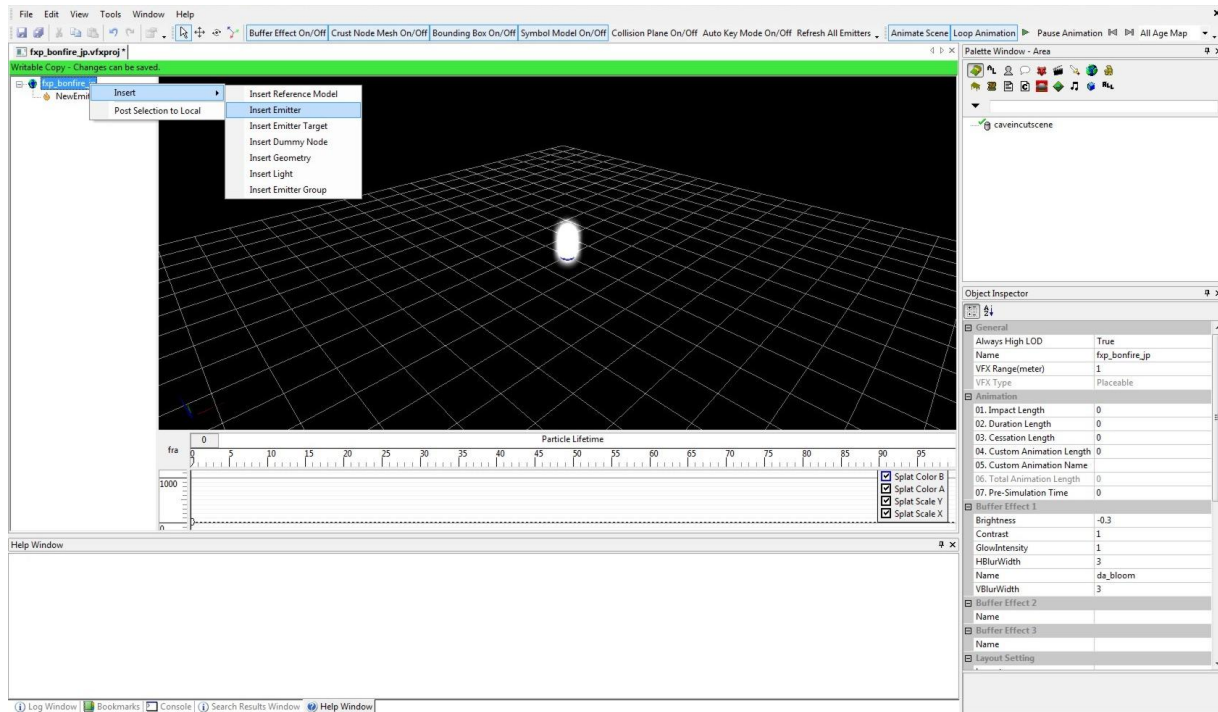
You may be thinking that 9 seconds is a very short fire. When the effect is set in a script, the duration time is set within

the function ApplyEffect\*. This duration time causes the duration time of the animation to loop over and over. So for example, if this bon fire effect is applied to an object for 6 min, the duration part of the effect will loop 120 times, 3 second duration \* 120 loops = 360 seconds = 6 minutes. Then the Cessation time play and the effect is done.

After setting these times, you should now see a time line something like this, More on this in part 2.



Next we need an emitter. Right click on the window and select “Insert > Insert Emitter”. You should get a pulsating white capsule in the center of the grid and a “NewEmitter” object added to the list on the left.



Select the NewEmitter object and you will see the properties in the Object inspector window.

### Properties Window:

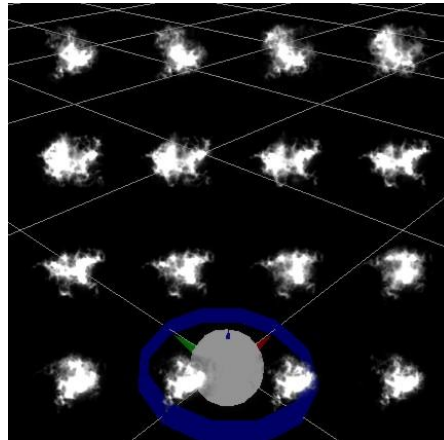
**Name:** Change the name of the emitter to “Base” this will be the main base of the fire. Naming the emitter will help sort them out when building an effect with several emitters.

**Material Library:** This is the library that we will be selecting our particle material from. The DA Effect Libraries start with the prefix fx\_ Click on the Material Library box to open the resource window. Select the fx\_firelibrary and click OK.

**Material Object:** This is the martial that we will use from the above selected Library. For this select “FireFlipbook”.

Before we continue, save your work and let’s talk about Flipbooks.

A flipbook is a particle that consists of columns and rows of varying images. The animation is created by the effect rapidly flipping through each image, much like a cartoon animation. At this point you should see 4 columns of flickering flames. Note that not all particles are in flipbook form, some are a static image. A simple way to see if the particle is a flipbook, and to see how many columns and rows there are is to set the Birth Rate to 1. Click on the “Emission” tab in the Object Properties and change the top parameter “Birth Rate” to 1. You should now see an image that consists of 4 rows and 4 columns of small flame images. This is a single particle, 4x4 flipbook. Set the Birth Rate back to 10, and return to the “Properties” tab of the Object Inspector, more on Birth Rate later.



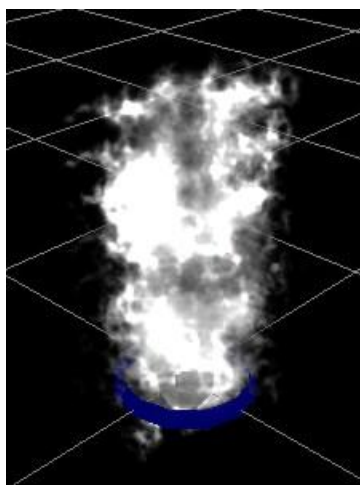
**Flipbook Type:** Select “ContactSheet”.

**Flipbook Random Start Frame:** This selects a random frame within the flipbook to start with on each emission. Set this to TRUE. This will help give us diversity and make the flame look more “real”.

**Flipbook Rows:** This is how many Rows there are in the flipbook, set this number to 4.

**Flipbook Columns:** This is the number of Columns in the flipbook, set this to 4 also.

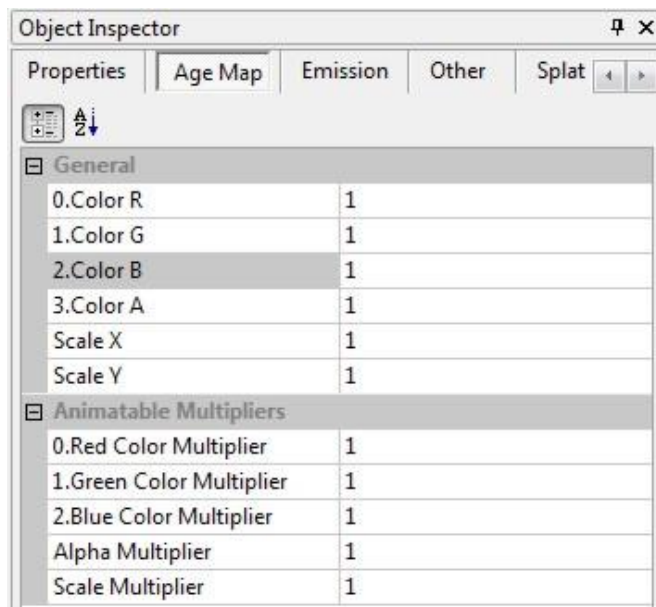
**Flipbook Frames per Second:** The particle image has 4 rows and 4 columns, there are 16 total images to flip through. This value should be a factor of Rows and Columns, so  $4 \times 4 = 16$ . Set this value to at least 16 flips per second.



You should now see a single column of what nearly looks like fire.

Right now the flames flicker a bit too slow for me, so set the Flipbook Frames per Second to 32. This makes the flames flicker twice as fast because it is now flipping through all 16 images twice per second instead of once.

Save your work.



In the Object Inspector click the “Age Map” Tab.

Here we can manipulate the color and the scale of the effect.

Color values are in percentage of 0 to 100% on a scale of 0 to 1, so 50% Blue would be 0.5.

Most normal fires aren’t white, so set Color B (blue) to 0. Your flame should now be yellow. Set Color G (green) to 0.4. We now have a orangey-red color.

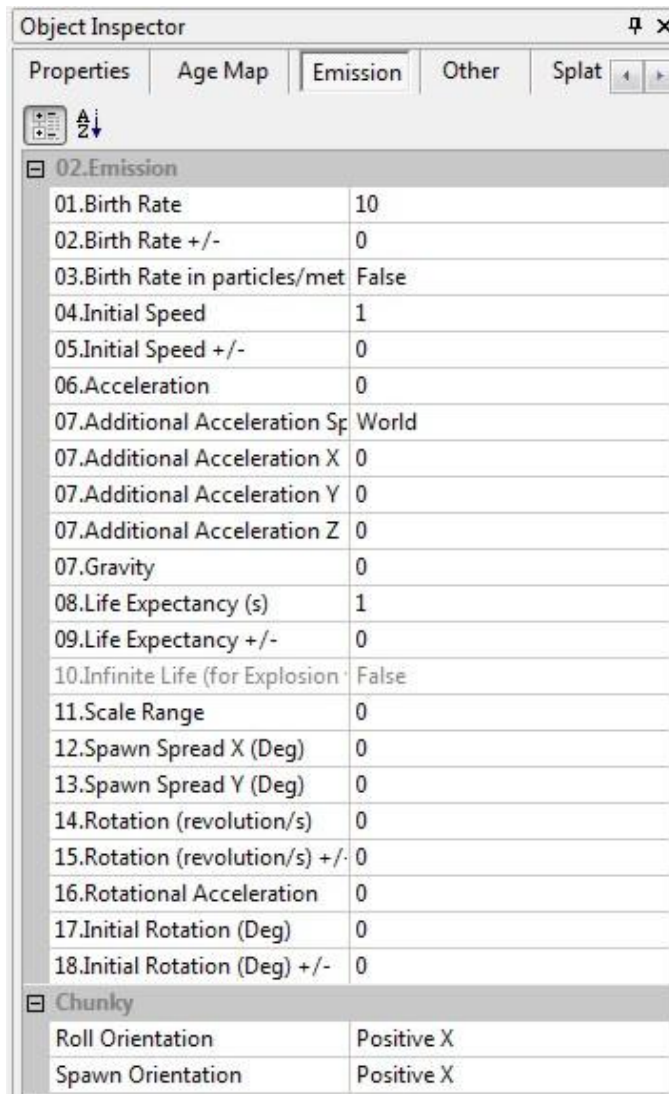
Color A (alph) is the flames color intensity setting this to 0.5, the fire is half as intense. Set it where you like.

The scale X and Y values can be any number from 0 to infinity. Setting the scale to 1 is the particles normal size. Because the particle image is a 2d image, there is no Z scale. Changing Scale X will affect the X (width) direction of the flame and Scale Y will affect the Y (Height) direction of the particle. For now we will leave them alone.

The corresponding multipliers simply multiply the values in the general section. For example, if you set Color G back to 1, then set Green Color Multiplier to 0.4 you will have the same result.  $1 * 0.4 = 0.4$

There is a notable difference between the 2 however, the General settings affect the animation as a hole over the length of the effect time while the Multipliers affect the partials’ over the life time of that particle. This will be explored further in Part 2. Go ahead and play with the color settings until you have it the way you want it, and then save your work.





Now click on the “Emission” tab these settings control the particle emissions.

**Birth Rate:** This is how many particle emit per second. The default is 10, which works out for the fire pretty good. However the flicker is pretty static. So we will change that in a minute.

**Birth Rate +/-:** This setting will cause the particle Birth Rate to randomly vary by a positive number or a negative number. If we set this value to 1, the Birth Rate then will vary between 9 and 11 particles per second. Since a 10 Birth Rate seems to be a good number for the most particles per second, set the +/- value to 2, and change the Birth Rate to 8. This will make the max 10 and the min 6 particles per second.

**Initial Speed:** This is how fast the particles travel when emitted from the emitter. Changing this value appears to change how long the particles last. However this is not the case, they simple move further before the particle life has expired.

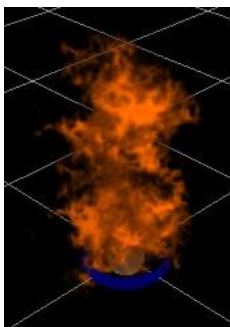
**Initial Speed +/-:** This setting works the same as the Birth Rate +/- setting but instead affects

the Initial Speed. Set this value to 0.5 which will now give the particles a varying speed between 0.5 and 1.5.

**Life Expectancy (s):** This is how long each particle last, in seconds, before vanishing.

**Life Expectancy +/-:** Again is a random variance to the value set in Life Expectancy. Set this value to 0.5, making the particle last longer and shorter as the fire burns.

**Scale Range:** This adds a random variance to the size of the particle much like the other +/- settings. Set this to 0.6. You should now see the flame vary in size as it emits from the base emitter.



Play with these settings a bit to get familiar with them. In the end you should have a raging fire animation. When you are happy, save your work.

**Getting the effect in game:** Right click on the Root and select “Post to Local”. Now your effect should be available to cut scenes, level editor and scripting etc.



**Notes:** Because an effect is Art work, at present will not be included with a Builder to Builder Create and must be sent separately. Scripting examples to come later once there is a client to test for accuracy.

So now you have a raging fire. In part 2 of this tutorial, we will make this fire dance and sing. Part 2 will introduce using the time line settings to vary coloration, scale, size and other more advanced settings as well as explore some of the settings introduced here a bit deeper. Keep the bon fire effect you created with this tutorial, we will need it in the second part. The bon fire effect can also be downloaded from the social site.