

AuroraFX

User Guide revision 1.1
www.flashloaded.com

Table of Contents

Installation	3
Getting Started	4
Component Inspector Parameters	5
Advanced Tutorial	6
Creating an Instance Dynamically	9
Actionscript 2 method	9
Actionscript 3 method	9
Customizing dynamic instances with AS2	10
Customizing dynamic instances with AS3	10
How to create an AuroraFX effect on a motion guide tween	11
Creating an AuroraFX component effect on a motion guide	11
Actionscript Reference	12
Help	16

Installation

You will need Adobe Extension Manager in order to install this component. The Extension Manager should have been installed by default when you installed Flash. You may download the latest version of Extension Manager for free from the [Adobe website](#).

1. Ensure that Flash is closed before installing the AuroraFX component.
2. Unzip/extract the aurorafx.zip file that you downloaded. You will find a file called either aurorafx.mxp or aurorafx_as3.mxp depending on which version you've downloaded.

Double click on this file in order to install the component using Extension Manager. The AuroraFX should now be installed in your Flash Components Panel.

Getting Started

This example shows you the most basic use of the AuroraFX component.

1. Open Flash and start a new file. *If you are intending to use the actionscript 2 version of the component ensure that you start an actionscript 2 file, likewise if you wish to use the actionscript 3 version you must start an actionscript 3 file. There is no difference between the two versions.*
2. Press Ctrl+F3 (Windows) or Cmnd+F3 (Mac) to open the properties panel and set the background color of your swf file to black. *It is important that you set the background to a dark shade in order for the AuroraFX component to be clearly visible, the effect will not be seen on a white background.*
3. Press Ctrl+F7 (Windows) or Cmnd+F7 (Mac) to open the Components panel.
4. If you're using Actionscript 2 locate the folder named 'AuroraFX', if you're using Actionscript 3 locate the folder named 'AuroraFX_AS3'.
5. Drag a copy of the AuroraFX component out of the folder and onto your stage.
6. Press Ctrl+s (Windows) or Cmnd+s (Mac) to save your file.
7. Press Ctrl+Enter (Windows) or Cmnd+Enter (Mac) to test it.

Component Inspector Parameters

Parameter	Description	Example
After Burn	The time it takes the emitter to disappear.	0.25
Auto Emit	If set to true the component starts as soon as it is added to the stage, if set to false the component only begins emitting when the start method is called.	TRUE
blendMode	The type of blending to apply to the effect. For a complete description of each type of blend refer to Flash's documentation. Possible values are "add", "darken", "difference", "hardlight", "invert", "lighten", "multiply", "normal", "overlay", "screen", "subtract".	add
Burn Strength	The strength of the effect.	35
Fizzle Time	The time it takes the effect to appear.	1
Life Time	The time the effect should last for.	0.5
Particle Type	the type of particle to use. Values can be 1, 2, 3, 4, 5, 6, 7 or 8.	1

Advanced Tutorial

We're now going to take a look at the parameters that can be set to customize the AuroraFX.

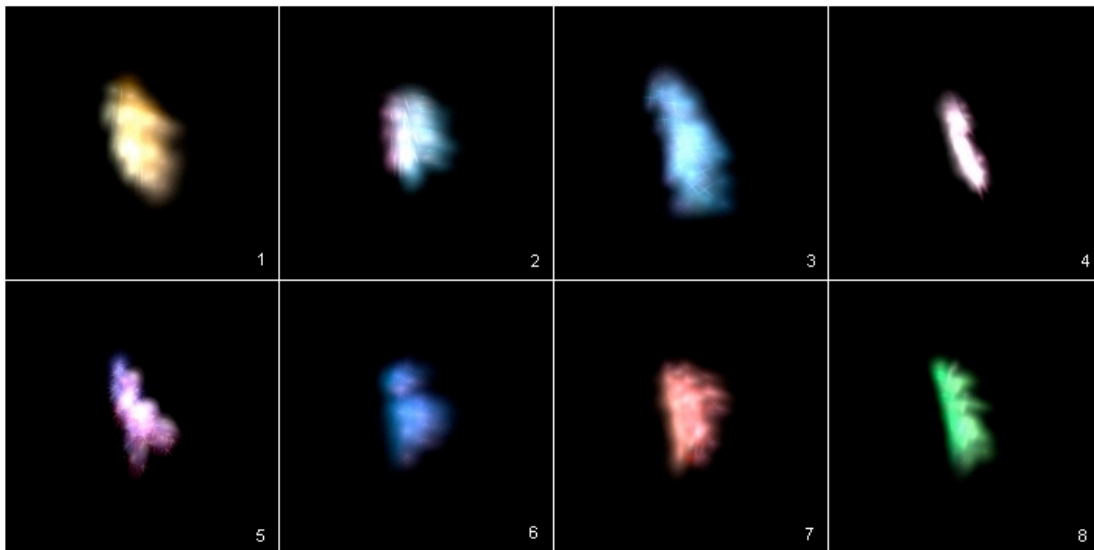
In this tutorial we'll be using the file you created in the previous tutorial 'Getting Started'.

Changing the Particle Type

The Particle Type is the type of particle used by the component.

1. Open the file you created in the 'Getting Started' tutorial.
2. Select the AuroraFX component on the stage.
3. With the AuroraFX component still selected press Shift+F7 to open the component inspector.
4. Locate the 'Particle Type' parameter and change its value from the default (which is 1) to 2.
5. Press Ctrl+F7 (Windows) or Cmnd+F7 (Mac) to test the file. You should see the kind of particle used has changed.

There are eight different particles to choose from, they are as follows:



Customizing the particle strength

Burn Strength determines the intensity of the visual appearance of the particle. You can change the Burn Strength of a particle following these steps:

1. Open the file you created in the 'Getting Started' tutorial.
2. Select the AuroraFX component on the stage.
3. With the AuroraFX component still selected press Shift+F7 to open the component inspector.
4. Locate the 'Burn Strength' parameter and change its value from the default (which is 35) to 95. *The 'Burn Strength' parameter accepts values from 1 to 100, the higher the value the stronger the effect.*
5. Press Ctrl+F7 (windows) or Cmnd+F7 (Mac) to test the file. You should see that the strength of the effect has increased.

Customizing the particle timing

The length of time a particle emits for is determined by three parameters, 'Fizzle Time', 'Life Time' and 'After Burn'. Alter these parameters like so:

1. Open the file you created in the 'Getting Started' tutorial.
2. Select the AuroraFX component on the stage.
3. With the AuroraFX component still selected press Shift+F7 to open the component inspector.
4. Locate the 'Fizzle Time' parameter and change its value from 1 to 2. *The 'Fizzle Time' determines how quickly a particle disappears and is measured in seconds.*
5. Locate the 'Life Time' parameter and change its value from .5 to .25. *The 'Life Time' determines how long a particle should exist before dissapating and is measured in seconds.*
6. Locate the 'After Burn' paramter and change its value from .25 to 0. *The 'After Burn' determines how long a particle takes to vanish once the 'Fizzle Time' has been expended and is measured in seconds.*
7. Press Ctrl+Enter (windows) or Cmnd+Enter (Mac) to test the file. You should see that the timing of the effect has changed.

Skinning a particle (Actionscript 2)

Skinning a particle lets you change its appearance. In order to skin the particle you must first load it into your library.

1. Open the file you created in the 'Getting Started' tutorial.
2. Press Ctrl+L (windows) or Command+L (mac) to open your library.
3. Select Window > Common Libraries > AuroraFX Assets to open the assets library.
4. Locate the particle skin you wish to edit. Each particle is inside its own folder, for example the particle that corresponds to particle type 1 is named 'particle1' and is inside the 'particle 01' folder.
5. Drag the particle you wish to skin across to your own library. Then close the Assets library.
6. In your own library double-click the particle symbol and edit its contents as desired.
7. Press Ctrl+Enter (windows) or Cmnd+Enter (Mac) to test the file. Note: if the file does not reflect the changes you have made make sure that you have skinned the particle that corresponds to the particleType.

Skinning a particle (Actionscript 3)

The Actionscript 3 version of the component takes advantage of the .fla based component clip to enable skinning on the timeline. To skin the Actionscript 3 version of the component follow these steps:

1. Open the file you created in the 'Getting Started' tutorial.
2. Double-click the AuroraFX component on the stage.
3. Unlock the skins layer and double-click on the skin you wish to edit.
4. Make your desired changes.
5. Press Ctrl+Enter (windows) or Cmnd+Enter (Mac) to test the file. Note: if the file does not reflect the changes you have made make sure that you have skinned the particle that corresponds to the particleType.

Creating an Instance Dynamically

Actionscript 2 method

You can add an instance of the AuroraFX component dynamically using Actionscript 2. To do so follow these steps:

1. Open Flash and start a new Actionscript 2 file.
2. Press Ctrl+F3 (Windows) or Cmnd+F3 (Mac) to open the properties panel and set the background color of your swf file to black. *It is important that you set the background to a dark shade in order for the AuroraFX component to be clearly visible.*
3. Press Ctrl+F7 (Windows) or Cmnd+F7 (Mac) to open the Components panel.
4. Locate the folder named 'AuroraFX'.
5. Drag a copy of the AuroraFX component out of the AuroraFX folder and onto your stage.
6. Delete the instance from the stage. *Adding the instance to the stage then deleting it adds the component to your file's library so that you can access it with script.*
7. Select the keyframe on your timeline and press Alt+F9 to open the actions panel. Enter the following Actionscript:

```
attachMovie("AuroraFX", "myAuroraFX", this.getNextHighestDepth());
```

8. Press Ctrl+s (Windows) or Cmnd+s (Mac) to save your file and Ctrl+Enter (Windows) or Cmnd +Enter (Mac) to test the file. You'll see the AuroraFX attached in the top left of the movie.

Actionscript 3 method

To add an instance of the AuroraFX component to the stage using Actionscript 3 follow these steps:

1. Open Flash and start a new Actionscript 3 file.
2. Press Ctrl+F3 (Windows) or Cmnd+F3 (Mac) to open the properties panel and set the background color of your swf file to black. As with the AS2 version the AS3 version will not show on a white background.
3. Press Ctrl+F7 (Windows) or Cmnd+F7 (Mac) to open the Components panel.
4. Locate the folder named 'AuroraFX_AS3'.

5. Drag a copy of the AuroraFX component out of the folder onto the stage.
6. Delete the instance from the stage. *Adding the instance to the stage then deleting it adds the component to your file's library so that you can access it with script.*
7. Select the keyframe on your timeline and press Alt+F9 to open the actions panel. Enter the following Actionscript:

```
import com.flashloaded.as3.AuroraFX;
var myAuroraFX:AuroraFX = new AuroraFX();
addChild(myAuroraFX);
```

8. Press Ctrl+s (Windows) or Cmd+s (Mac) to save your file and Ctrl+Enter (Windows) or Cmd+Enter (Mac) to test the file. You'll see the AuroraFX attached in the top left of the movie.

Customizing dynamic instances with AS2

The attachMovie method allows you to pass an object to the symbol you're attaching so it is possible to specify x and y coordinates for the component like so:

```
attachMovie("AuroraFX", "myAuroraFX", this.getNextHighestDepth(), {_x:50, _y:200});
```

It is also possible to specify a value for any of the component's properties. So for example you could specify that the component should use particleType 8:

```
attachMovie("AuroraFX", "myAuroraFX", this.getNextHighestDepth(),
{particleType:8});
```

Customizing dynamic instances with AS3

In order to specify properties for the AuroraFX component with Actionscript 3 we need to specify each property after the component is created like so:

```
import com.flashloaded.as3.AuroraFX;
var myAuroraFX:AuroraFX = new AuroraFX();
myAuroraFX.x = 50;
myAuroraFX.y = 200;
myAuroraFX.particleType = 8;
addChild(myAuroraFX);
```

How to create an AuroraFX effect on a motion guide tween

Creating an AuroraFX component effect on a motion guide

These steps must be followed exactly in order to create the desired effect. Any deviations from these exact steps will cause unexpected errors.

Make sure that you are using a minimum of 30 frames per second to ensure a smoother effect.

1. Create a motion path:

- a. Create the desired line on your stage and modify it until you are satisfied. This will be our guide.
- b. Convert the line to a movie clip and select its handler to enter its sub movie clip. Right click/ctrl click on the guide layer and convert the line layer into a guide type by selecting "guide".
- c. Insert a new layer and drag it underneath the guide layer so it associates to the guide.

2. Attach the AuroraFX component:

- a. Drag and drop an AuroraFX component from the component palette. You can access the component palette from Windows > Components or press CTRL+F7 (or Cmnd+F7 on Mac). Tweak the AuroraFX component through the component inspector. Align it to the start of the guide. You must give the AuroraFX component an instance name. Name it *myafx*.
- b. Insert a desired amount of frames for the end of the motion tween. You will generally want to insert as many frames to create an effect for at least 1 second. Press F6 in the last frame to create a key frame and drag the instance of the AuroraFX that is in the last keyframe to the end of the guide. Create the motion tween by selecting the keyframe and choosing to create a motion tween in the properties panel (note: do not right click/ctrl click on the timeline to create the motion tween).
- c. On the last frame of the motion tween, make sure you put in the following code:

```
stop();  
myafx.stop();
```

This will stop the animation and kill the AuroraFX from continuing to creating auroras.

Actionscript Reference

Events

emitterComplete

Availability

Flash Player 8

Description

Broadcast when all emitting has ceased.

Actionscript 2 Example

```
myAFX.addEventListener("emitterComplete", this);
function emitterComplete(){
    trace("Complete");
}
```

Actionscript 3 Example

```
myAFX.addEventListener("emitterComplete", "completeHandler");
function completeHandler(eo:Event):void {
    trace("Complete");
}
```

Methods

start

Availability

Flash Player 8

Description

Starts the AuroraFX emitting.

Example

```
myAFX.start();
```

stop

Availability

Flash Player 8

Description

Stops the AuroraFX emitting.

Example

```
myAFX.stop();
```

Properties

afterBurn

Availability

Flash Player 8

Description

Number: the time it takes the emitter to disappear.

Example

```
myAFX.afterBurn = 1;
```

autoEmit

Availability

Flash Player 8

Description

Boolean: If set to true the component starts as soon as it is added to the stage, if set to false the component only begins emitting when the start method is called.

Example

```
myAFX.autoEmit = true;
```

blendMode

Availability

Flash Player 8

Description

String: the type of blending to apply to the effect. The options are "add", "darken", "difference", "hardlight", "invert", "lighten", "multiply", "normal", "overlay", "screen", "subtract". Refer to Flash's documentation for a complete description of each option.

Example

```
myAFX.blendMode = "add";
```

burnStrength

Availability

Flash Player 8

Description

Number: the strength of the effect.

Example

```
myAFX.burnStrength = 35;
```

fizzleTime

Availability

Flash Player 8

Description

Number: The time it takes the effect to appear.

Example

```
myAFX.fizzleTime = 1;
```

lifeTime

Availability

Flash Player 8

Description

Number: The time the effect should last for.

Example

```
myAFX.lifeTime = 5;
```

particleType

Availability

Flash Player 9

Description

Number: the type of particle to use. Values can be 1, 2, 3, 4, 5, 6, 7 or 8.

Example

```
myAFX.particleType = 3;
```

Help

This component is fully supported by the Flashloaded support team through our support forum. You will also find tips and additional information in the forum as well as announcements for version updates: [AuroraFX Support Forum](#)

Note: In order to post a question in the forum, you will need to [register](#) by creating a username and password. This registration differs from your account login.