

FlashTextFX

User Guide revision 1.2
www.flashloaded.com

Table of Contents

Installation	4
Getting started	5
Building a text animation sequence	8
Using the TextFXStep interface	9
Resizing the Component Inspector panel	9
Steps, preview and settings	10
Saving and loading preset settings	11
FlashTextFX Component Inspector Parameters	12
TextFXStep Component Inspector Parameters	13
Timing settings	13
Parts settings	13
Tween settings	14
Position settings	14
Properties settings	15
Filter settings	15
Changing the position and scale of parts	16
Using the bezier tool	18
Working with links	20
To animate linked text in mouse over:	20
Link actions:	20
Link events	21
XML text sequence	22

Displaying RSS and Atom feeds	23
Editing the display of RSS titles and descriptions	24
Editing the display of Atom titles and descriptions	25
Flash player security instructions for RSS & Atom feeds	26
ActionScript properties	28
ActionScript methods	31
ActionScript events	32
Help	35

Installation

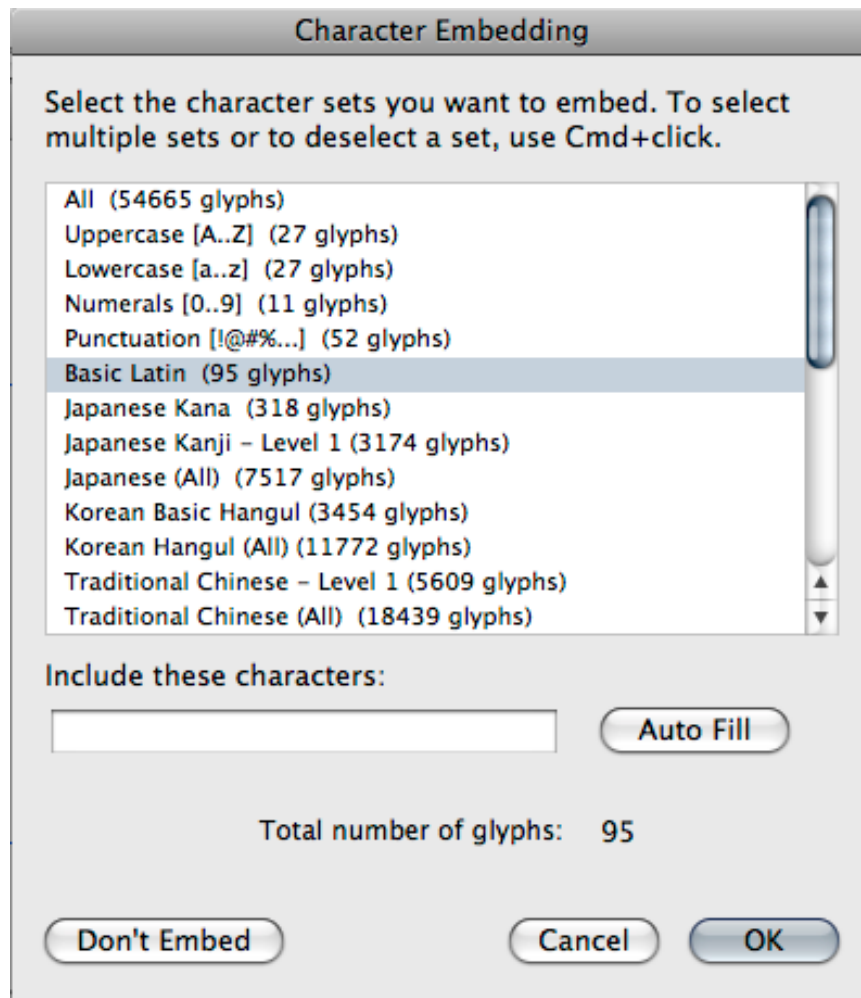
You will need Adobe Extension Manager in order to install this component. Extension Manager should have been installed by default when you installed Flash. You may download the latest version of Extension Manager for free from the [Adobe website](#).

1. Ensure that Flash is closed before installing the FlashTextFX component.
2. Unzip/extract the FlashTextFX.zip file that you downloaded. You will find a file called FlashTextFX.mxp. Double click on this file in order to install the component using Extension Manager.

The FlashTextFX should now be installed in your Flash Components Panel.

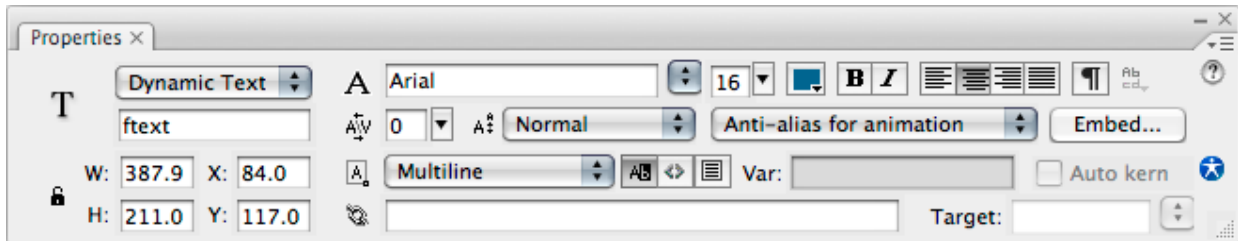
Getting started

1. Having installed the FlashTextFX using the Adobe Extension Manager, start a new Flash ActionScript 3 file and save it.
2. Place a textfield on the stage in which you would like the text to appear. The textfield must have the following settings:
 - a. Dynamic Text
 - b. Anti-alias for animation
3. Select your desired font, style, size, alignment and color for the text and choose whether the textfield should be single line or multiline.
4. Click on the *Embed* button, select the character sets to embed and press *OK*. For most latin languages, selected *Basic Latin* should be sufficient.

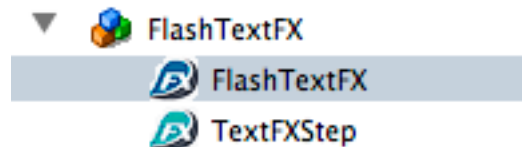


Note: It's recommended to avoid choosing to embed All characters as some fonts contain several hundred characters that you will not be using, which can cause your file to be excessively heavy.

5. Give the textfield an instance name. For example: *ftext*. Your textfield properties should now look something like this:



6. Drag and drop *FlashTextFX* component from the Components panel on top of the textfield that you just created:



Click on the *FlashTextFX* component that you just dragged onto the stage and open the Component Inspector panel (shift +F7) in order to set the parameters. The following basic parameters must be entered:

a. **target text field:** This is the instance name of the textfield that you created. This value should be completed automatically so there should be no need to change this.

b. **text sequence:** This is the sequence of sentences that you wish to appear in the textfield. The effects will be applied to each sentence and will be a way of transitioning between the sentences. To add a sentence to the sequence, double click in the value field and click the + symbol. Type the sentence or sentences and press OK when done.

The text sequence can also be entered in an external XML file. To use this option, enter the path and name of the XML file in the **xml path** parameter. Please refer to the [XML text sequence](#) section for instructions on using this option.

Note: The text sequence and xml path parameters can also be left blank if the text is entered directly into the textfield that's on the stage.

7. Drag and drop **TextFXStep** component from the Components panel anywhere onto the stage, ensuring that it is in the same frame as the textfield and the *FlashTextFX* component.
8. At this stage, you can already test the *FlashTextFX* with the default parameters, to ensure that you have set it up correctly. Press Ctrl+Enter (win) or Cmnd+Enter (mac) to test your movie. You should

see each sentence wipe in with the default effect settings.

9. For instructions on adding more effects and customizing the look, please see the next section on Building a text animation sequence. Please see the [Component Inspector parameters](#) section for a description on each parameter setting.

Note:

1. *In order for the animations to be smooth it's recommended to set your movie speed to 31 fps.*
2. *If you wish to have more than one instance of the FlashTextFX on the same stage, the textfield, FlashTextFX and corresponding TextFXStep components should be in separate movie clips for each instance.*

Building a text animation sequence

In order to create a text animation sequence, you would use multiple instances of the *TextFXStep* component. Each instance is defined with its own step number, indicating the order in which that effect will be executed.

For example:

Lets say that we want to create text effect that will fade in one sentence, keep it displayed for 2 seconds, fade it out and then fade in a second sentence, keep it displayed for 2 seconds, fade it out then repeat it again constantly.

We would create an animation with two steps - one for fade in and one for fade out. To do this, we use two *TextFXStep* components.

1. Drag and drop the first instance of the **TextFXStep** component onto the stage (you may already have once instance on the stage).
2. Set **End Pause** to 2 seconds.

This will cause the text to fade in and to pause for 2 seconds.
3. Drag and drop a second instance of the **TextFXStep** component onto the stage.
4. Set the step number to 2 (as this is the second step in the sequence of effects) and set the tween mode to be **B -> A** as this will cause the effect to go from the current state (B) to the fade out setting defined in the component parameters (A).
5. Press Ctrl+Enter (win) or Cmnd+Enter (mac) to test your movie. You should see each sentence wipe/fade in and wipe/fade out.

For an interesting variation, try set the position for the step 1 TextFXStep to **random**.

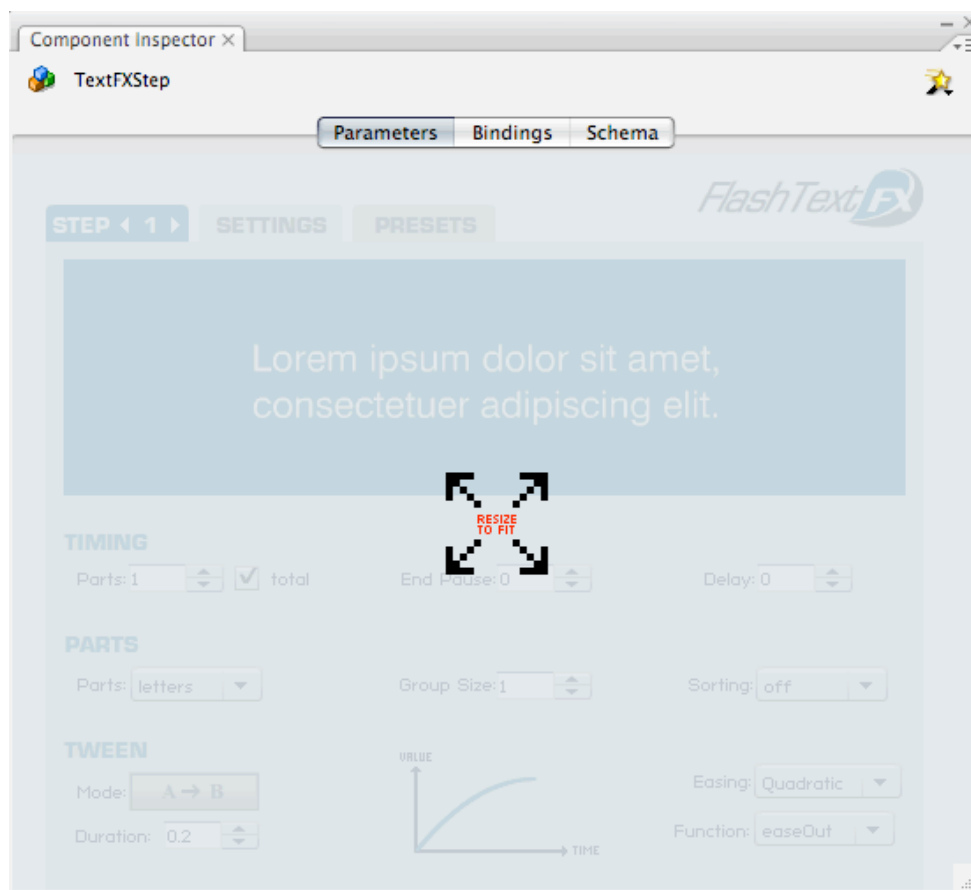
Experiment with different combinations of parameters in each of the TextFXStep components to see the different effects that you can create.

Using the TextFXStep interface

The TextFXStep component has a custom user interface which opens in the Component Inspector panel. All of the TextFXStep parameters are set in this interface.

Resizing the Component Inspector panel

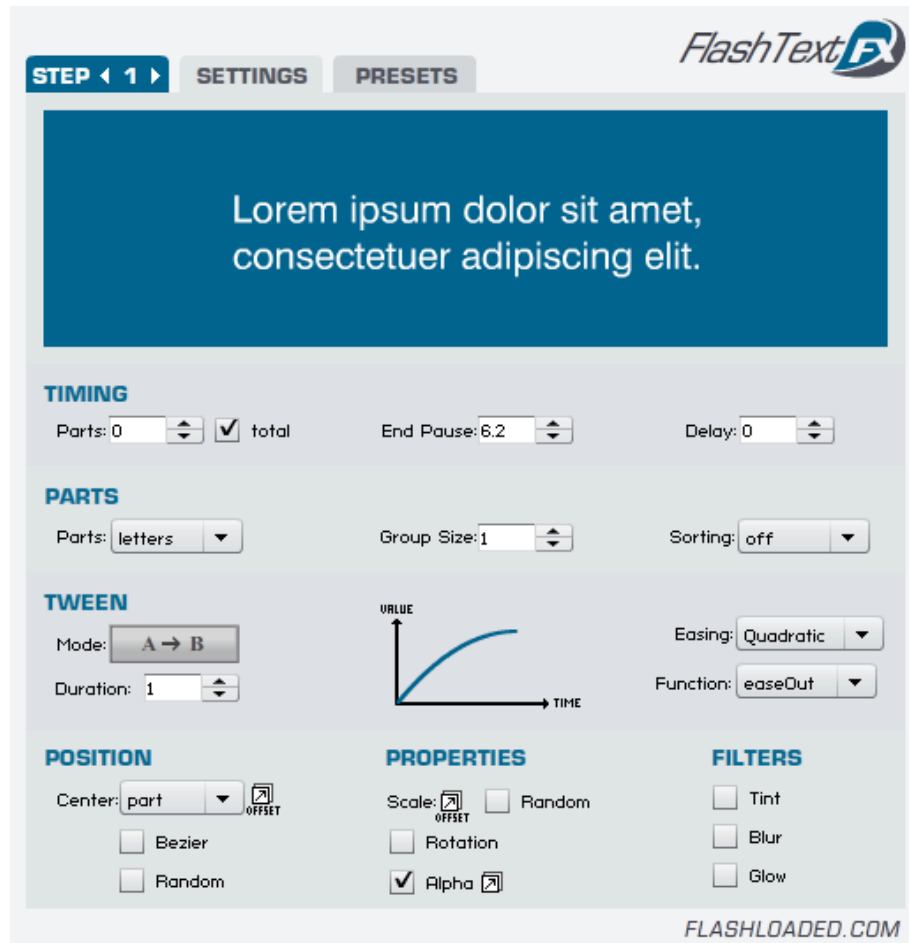
The TextFXStep interface requires a minimum amount of space for the Component Inspector panel. If you get the image below, simply resize the Component Inspector panel until the interface fits.



Steps, preview and settings

The step number for the TextFXStep component is set by clicking on the left and right arrows alongside the step tab in the top left of the TextFXStep interface.

All of the parameters are set on the settings tab. The preview at the top will change as each setting is changed.



Saving and loading preset settings

If you have created an effect that you may wish to reuse at some point, you can save the current effect by selecting the **presets** tab. Enter a name for this style of your choice and press save. The style will now be saved. To load a style, select it from the **presets** list and press the **load** button.

Preset styles can be deleted by selecting the style from the **presets** list and pressing the red X.



FlashTextFX Component Inspector Parameters

Parameter	Description	Example
Text sequence	The list of sentences to display and the order in which to display them.	
- xml path	The path and filename of the optional XML file containing the text sequence. See the XML text sequence section for instructions on creating an XML file. <i>Note: The path must be relative to the location of the HTML file in which the SWF is embedded.</i>	sequence.xml
- on end	Sets the action that should be performed upon completing the sequence: <i>stop, repeat, next frame</i>	TRUE
Link step	The animation can be set to appear over linked text only, when the mouse cursor hovers over it. This setting specifies the step number of the animation that should be applied when mousing over the linked text. See the Working with links section for more information on how to use the FlashTextFX component with links.	2
Link color	The color of linked text.	#0000FF
Link action	The action to perform when clicking on the linked text.	open url
- function name	The name of the function on the same timeline to execute. Object argument containing link properties is passed to the function.	
Delay	The delay (in seconds) before starting the animation sequence.	0
Pause	The pause (in seconds) after completing the animation sequence.	0
Target text field	The instance name of the dynamic textfield in which the animated text will appear. This parameter should appear automatically when dragging the FlashTextFX component onto the textfield.	ftext

TextFXStep Component Inspector Parameters

Timing settings

Parameter	Description
Parts	The number of parts to apply to the timing settings.
Parts - total	Checked = The total time that it will take for all of the text parts to appear Unchecked = The time for each part to appear
End pause	The number of seconds to pause after completing the animation in this step.
Delay	The number of seconds to wait before starting this animation step.

Parts settings

Parameter	Description
Parts	<p>Defines how the text is divided. The effect will occur on each part sequentially. For example, if set to <i>letters</i>, the effect will be applied letter by letter, according to the order specified in the <i>sorting</i> setting.</p> <p>Pixels: You will have the option to enter the bitmap width and height of the pixels. A small pixel size can impact on performance.</p> <p>Letters: The effects is applied sequentially letter by letter.</p> <p>Words: The effects is applied sequentially word by word.</p> <p>Line: The effects is applied sequentially line by line.</p> <p>Field: The effects is applied to the whole text field at once.</p> <p>Use previous: This option can only be used when there is a previous animation step defined. Using this option will use the same parts setting as defined in the previous step.</p>
Group size	The parts can be animated in groups. This setting determines the size of each group. For example, if parts is set to <i>letters</i> and groups is set to 2, the animation will occur in groups of 2 letters at a time.
Sorting	The order in which the parts animate/appear. Setting this to <i>off</i> will cause the parts to animate at the default left to right. <i>Reverse</i> , causes the text to animate from right to left and <i>random</i> will animate each part at a random position.

Tween settings

Parameter	Description
Mode	<p>This setting determines whether the text should animate from the current state to the settings specified for this step or whether it should start at the step settings and tween into the current state.</p> <p>A represents the state defined for this step in the component parameters B represents the current state</p> <p>A->B would usually only be chosen for the first step as this builds the text. B->A would usually be used for each subsequent step. B->A->B is used for a yo-yo effect.</p> <p>The result is this setting is shown using a graphical representation.</p>
Duration	The duration of the tween for each text part.
Easing	The style of easing to use for the animation. The result is this setting is shown using a graphical representation.
Function	Sets whether the easing should be set to <i>easeOut</i> , <i>easeInOut</i> or <i>easeIn</i> . The result is this setting is shown using a graphical representation.

Position settings

Parameter	Description
Center	<p>Specifies the center coordinate from which the text part appears. The X and Y coordinates can be set.</p> <p>Part: The coordinate system is referenced relative to the position of each part Fixed: The coordinate system is fixed relative to the text field position</p>
Offset	The X and Y settings to offset the parts. The offset can be set for each part number by clicking on the up/down arrows in the offset settings window.
Bezier	The text parts are scattered along a bezier. The shape of the bezier can be customized.
Random	The text parts are scattered randomly. The range for the random area can be customized.

Properties settings

Parameter	Description
Scale offset	The X and Y settings to offset the parts. The offset can be set for each part number by clicking on the up/down arrows in the offset settings window.
Scale random	The parts can be scaled randomly while animating within the specified X and Y range.
Rotation	The parts can be rotated randomly within the specified range while animating.
Alpha	Sets the text to fade in or fade out by changing the mode in the alpha settings to $A>B$ or $B>A$. The amount of alpha can be set.

Filter settings

Parameter	Description
Tint	Sets the text to tween into the defined tint color. When this option is checked, an icon will appear to the right for more custom options for this setting.
Blur	Sets the text to blur according to the blur settings. When this option is checked, an icon will appear to the right for more custom options for this setting.
Glow	Sets the text to glow according to the glow settings. When this option is checked, an icon will appear to the right for more custom options for this setting.

Changing the position and scale of parts

The x/y position from which each part of the text animates can be set through the TextFXStep component interface. The parts can also animate in scale. The simplest way to understand this concept it by following this short tutorial:

1. Start a new Flash ActionScript file and follow all of the steps in the [Getting started](#) section. This should get you to the point where you have the textfield on the stage, with the FlashTextFX component and one instance of the TextFXStep component.

If you test your movie at this stage, you should see the text fade in from left to right.

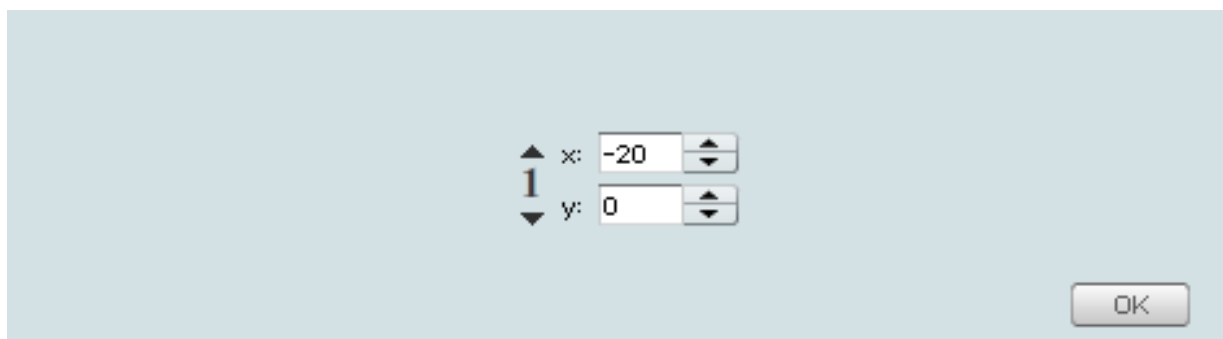
2. Click on the TextFXStep component that's on the stage and check that **parts** is set to **letters** like this:



3. Ensure that the **center** position is set to **part** and click on the **offset** icon.



4. The position offset panel will now open at the top of the interface, where the preview text normally appears. Change the x value to -20. This will cause the text to animate in, 20 pixels from the left. Press OK when done.

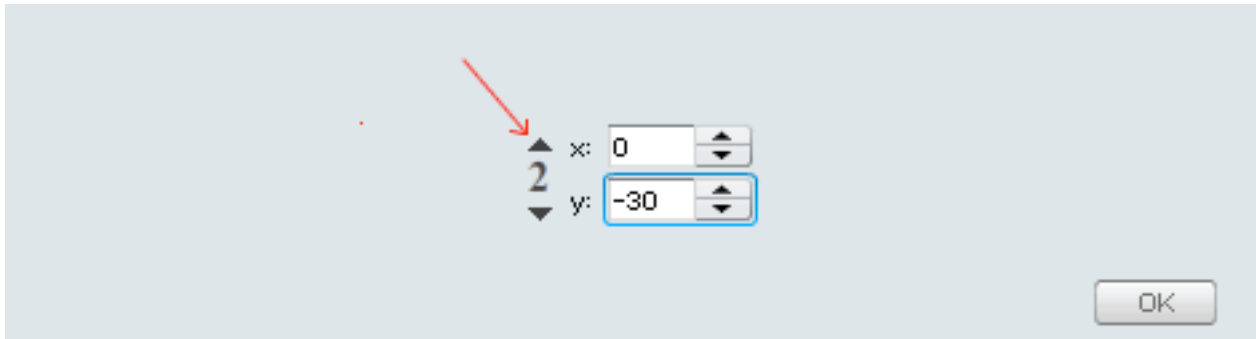


Test your movie (ctrl+enter or cmd+enter) to see how this looks. You will notice that each letter animates in 20 pixels from the left.

5. Click on the offset icon again to open the offset panel. This time, we're going to change the change the offset for the second part. As the parts has been set to letters, this change will have an effect on every second letter.

To do this, click on the up arrow that appears above the number 1 that's on the left to change it to number 2. Set the y value to -30. Press OK when done.

Test your movie (ctrl+enter or cmd+enter) to see how this looks. You will notice that every odd letter animates in 20 pixels from the left and every even letter animates in 30 pixels from the top. For

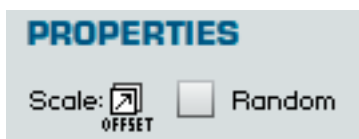


example, in the word FLASH:

- F animates in 20px from the left
- L animates in 30px from the top
- A animates in 20px from the left
- S animates in 30px from the top
- H animates in 20px from the left

You can expand on this by adding a different offset for each letter. The pattern will repeat based on the number of parts.

6. You can now follow the same principle to animate the scale of the parts (in this case, each letter). Click on the offset icon in the properties scale setting:



7. Clicking on this will open the scale settings panel which is similar to the position offset panel. Check the **constrain** checkbox to constrain the x and y proportions while scaling.
8. After changing these settings, experiment with the tween mode setting (A>B, B<A, B>A>B) to see the different results that can be achieved.



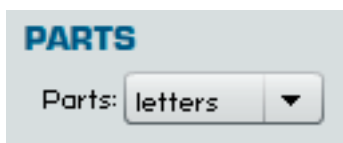
Using the bezier tool

The bezier tool can be a very powerful way to create advanced text animations. This short tutorial will give you a good idea as to how to use this.

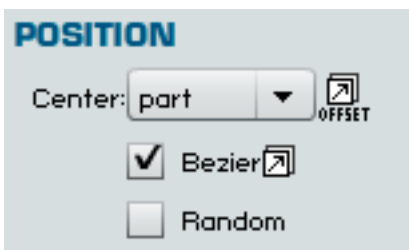
1. Start a new Flash ActionScript file and follow all of the steps in the [Getting started](#) section. This should get you to the point where you have the textfield on the stage, with the FlashTextFX component and one instance of the TextFXStep component.



If you test your movie at this stage, you should see the text fade in from left to right.

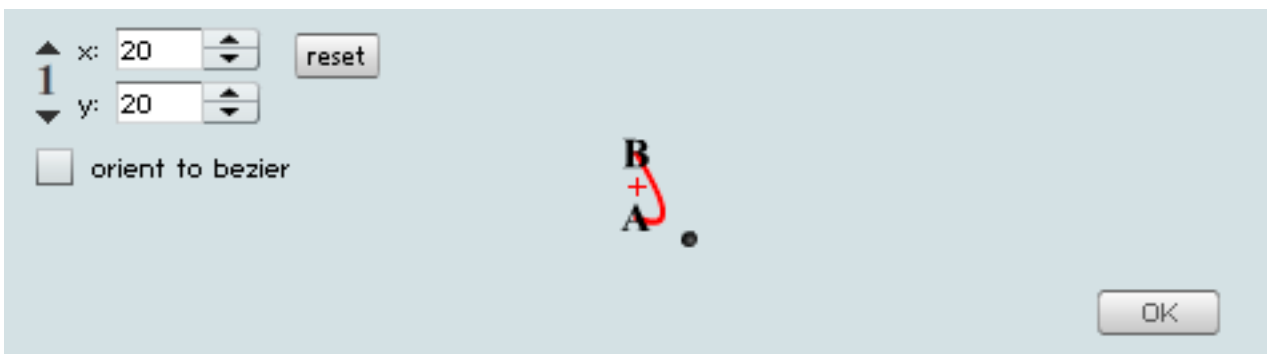
2. Click on the TextFXStep component that's on the stage and check that **parts** is set to **letters** like this:




3. Check the **bezier** checkbox in the position settings.



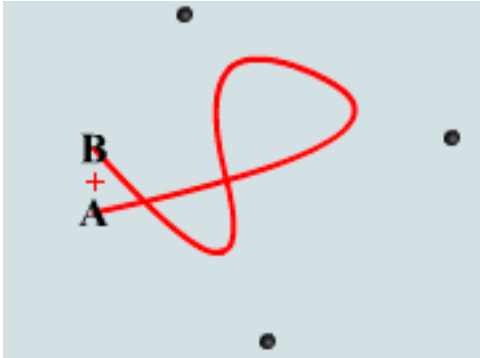
4. Click on the  icon which appears to the right of the ticked bezier checkbox. This will open the bezier settings  panel.



The bezier is represented by the red line between points A and B. The A and B points are draggable and they do not represent the actual path by default. Tick the orient to bezier checkbox to rotate the letters to the line of the bezier.

5. Click and drag the dot  to change the shape of the bezier. More points (more dots) can be added by clicking on the red line of the bezier at the desired point and dragging the point to form a new shape for the bezier. Remember to drag the A and B letters as well.

For example:



6. Press OK when done and test your movie (ctrl+enter or cmd+enter).

Note: Unchecking the bezier checkbox will cause all of the bezier settings to be reset.

Working with links

Hyperlinks defined in text using the FlashTextFX component can have rollover and rollout events. You can also set the animation to occur on the linked text only on mouse over.

To animate linked text in mouse over:

1. Ensure that the textfield is defined to *render text as HTML*.
2. Use the HTML href tags to define links without your text. For example:

```
<a href="http://www.flashloaded.com">Click here</a> to visit the  
Flashloaded website
```

3. Set the animation to use by entering the step number of the desired TextFXStep animation in the **link step** parameter.

When the animation reaches the step that is defined as the *link step*, it will stop and the step settings will apply only to the hyperlinked portion of the text when the mouse is over it.

Link actions:

You can choose one of the predefined actions to execute when a link is clicked or subscribe to the link events in order to create custom link actions through ActionScript. To use a custom ActionScript function, set the **link action** parameter to *none* and add a listener for the link events in ActionScript, or choose the *execute function* option and enter the name of the function to execute (which is on the same timeline). The object will be passed to the executed function, containing the link properties.

Link events

Events are called whenever the FlashTextFX performs the specified action. The following link events are available:

TextFXEvent.SEQUENCE

Triggered at the start and end of the sequence

TextFXEvent.STEP

Triggered at the start and end of a step

TextFXEvent.LINK_OVER

Triggered when the mouse is over hyperlinked text

TextFXEvent.LINK_OUT

Triggered when the mouse moves off hyperlinked text

TextFXEvent.LINK_CLICK

Triggered when hyperlinked text is clicked

Example using a link event:

```
import com.flashloaded.textfx.events.*;

ftx.addEventListener(TextFXEvent.LINK_CLICK, linkEvent);

function linkEvent(event:TextFXEvent):void
{
    trace(event.link.url)
}
```

XML text sequence

The sequence of sentences and their corresponding links can be added using an external XML file. This makes it possible to edit the content without having to republish the SWF. A sample XML file is included with your download in the *examples* folder.

1. Open your favorite plain text editor (for example Notepad on Windows or TextEdit on Mac) and start a new file. *Note: If you are using TextEdit on Mac, choose Format > Make Plain Text*
2. Begin your file with the following line:

```
<?xml version="1.0" encoding="utf-8"?>
```

This is the standard xml declaration.

3. Add the following lines to your xml file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8"?>  
<sequence>  
</sequence>
```

4. Add the text tags to your XML file (the bold lines are the new additions)

```
<?xml version="1.0" encoding="utf-8" ?>  
<sequence>  
  <text url="http://www.google.com" target="_self">Visit Google</text>  
  <text url="http://www.flashloaded.com">Visit Flashloaded</text>  
  <text url="http://www.fontsforflash.com">Pixel fonts</text>  
</sequence>
```

Note: The url is optional. You may omit this if you do not want the text to be links.

5. Save the XML file and enter the name and path to the XML file in the **xml path** parameter of the **FlashTextFX** component.

Displaying RSS and Atom feeds

The FlashTextFX can be used to display RSS and Atom feeds with text animations. This is how you would do this:

1. Follow the [getting started](#) tutorial in order to set up your textfield with the FlashTextFX component and your desired number of TextFXStep components.
2. Save the .fla and copy the class file called RSS2.as (for RSS feeds) or Atom.as (for Atom feeds) into the same folder as your .fla. You can locate these files in the RSS classes folder in the download.
3. Give the FlashTextFX component that's on the stage an instance name, for example: *ftx*
4. For RSS feeds, add the following line of ActionScript code to the timeline - in the same frame as the FlashTextFX component:

```
RSS2.load(ftx, "http://www.your-rss-feed.rss");
```

Note:

*In the above code, replace **ftx** with the instance name of the FlashTextFX component that's on your stage and replace <http://www.your-rss-feed.rss> with the URL of your RSS feed.*

5. For Atom feeds, add the following line of ActionScript code to the timeline - in the same frame as the FlashTextFX component:

```
var feed:String = "http://your-atom-feed.xml";  
Atom.load( ftx, feed );
```

Note:

*In the above code, replace **ftx** with the instance name of the FlashTextFX component that's on your stage and replace <http://www.your-atom-feed.rss> with the URL of your Atom feed.*

6. Press Ctrl+Enter (win) or Cmnd+Enter (mac) to test your movie.

IMPORTANT:

Please refer to the section called [Flash player security instructions for RSS & Atom feeds](#).

Editing the display of RSS titles and descriptions

The RSS2.as class that is supplied with the component has been set to display the titles and descriptions of the feeds in the same font, with the description underneath the title. You can edit this class in Flash in order to change the look or to display the titles or descriptions only. We have made a few options in the class already which just need to be uncommented.

To do this, open the RSS2.as file in Flash and locate the following code:

```
// show title only
//ftx.add(prop.title, prop.link)

//show title and description
ftx.add(prop.title + '<BR/>' + prop.description,prop.link);

//show title and description reformatted
//ftx.add('<BR/><FONT COLOR="#ff0000">' + prop.title+'</FONT><BR/>
<FONT SIZE="16">' + prop.description + '</FONT>',prop.link );
```

In this example, the **purple** line is the currently active style. To deactivate it and to active a different style, simply comment it out by adding // at the beginning of the line and remove the comments from another style. For example:

```
// show title only
//ftx.add(prop.title, prop.link)

//show title and description
//ftx.add(prop.title + '<BR/>' + prop.description,prop.link);

//show title and description reformatted
ftx.add('<BR/><FONT COLOR="#ff0000">' + prop.title+'</FONT><BR/>
<FONT SIZE="16">' + prop.description + '</FONT>',prop.link );
```

This code uses regular HTML formatting which you can change to suit your needs. Save the RSS2.as file when done.

Editing the display of Atom titles and descriptions

The Atom.as class that is supplied with the component has been set to display the titles and descriptions of the feeds in the same font, with the description underneath the title. You can edit this class in Flash in order to change the look or to display the titles or descriptions only. We have made a few options in the class already which just need to be uncommented.

To do this, open the Atom.as file in Flash and locate the following code:

```
// show title only
//ftx.add( prop.ATOM_NS::title, prop.ATOM_NS::link.@href );

//show title and description
ftx.add( prop.ATOM_NS::title + '<BR/>'
+prop.ATOM_NS::summary,prop.ATOM_NS::link.@href );

//show title and description reformatted
//ftx.add( '<BR/><FONT COLOR="#ff0000">' + prop.ATOM_NS::title + '</
FONT><BR/><FONT SIZE="16">' + prop.ATOM_NS::summary + '</
FONT>',prop.ATOM_NS::link.@href );
```

In this example, the **purple** line is the currently active style. To deactivate it and to active a different style, simply comment it out by adding // at the beginning of the line and remove the comments from another style. For example:

```
// show title only
//ftx.add( prop.ATOM_NS::title, prop.ATOM_NS::link.@href );

//show title and description
//ftx.add( prop.ATOM_NS::title + '<BR/>'
+prop.ATOM_NS::summary,prop.ATOM_NS::link.@href );

//show title and description reformatted
ftx.add( '<BR/><FONT COLOR="#ff0000">' + prop.ATOM_NS::title + '</
FONT><BR/><FONT SIZE="16">' + prop.ATOM_NS::summary + '</
FONT>',prop.ATOM_NS::link.@href );
```

This code uses regular HTML formatting which you can change to suit your needs. Save the Atom.as file when done.

Important note:

Some Atom feeds contain two links for each item. In such cases, the property **prop.ATOM_NS::link.@href** should be changed to **prop.ATOM_NS::link.@href[0]**. You will know if this needs to be done if the URL is incorrect when clicking on an item.

Flash player security instructions for RSS & Atom feeds

For security purposes, the Flash player does not allow SWF's to read XML (or RSS) files that are located on a server other than the server that is hosting the SWF file. This will probably be the scenario in most of your cases as generally, RSS feeds reside on external websevers. You will still be able to view the RSS or Atom contents using the FlashTextFX when viewing your SWF locally, however this will not work when you upload the file to a webserver.

The solution is to upload a proxy file to your webserver for each RSS or Atom URL that you wish to read using the FlashTextFX. This proxy file contains the URL of the RSS/Atom feed to read. This file must reside on the same webserver as the SWF file and you would specify the name and URL of the proxy file in your ActionScript code, instead of the URL of the RSS/Atom feed.

For further instructions and to download the proxy file most suited to you, please see the following Adobe link: http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_16520&sliceId=1

The sections that you should read is called *Server-side proxy method*. If you have PHP installed on your webserver, we recommend using the PHP solution. If you only have ASP then you should go with the ASP option. If you have PHP 3 installed on your webserver and you select the PHP option, you must install that the file name ends in ".php3" instead of ".php".

Example using the PHP proxy file:

In this example, we'll be loading accessing the RSS file of the Flashloaded website. The URL for this RSS feed is: <http://www.flashloaded.com/rss/flashloaded.rss>

1. Download the PHP file from the adobe website. You can download this file by clicking [here](#).
2. Open the PHP file in any text editor (for example, Notepad or Textedit). You should see the following:

```
<?php
    $dataURL = "http://www.macromedia.com/desdev/resources/
macromedia_resources.xml";

    //note that this will not follow redirects
    readfile($dataURL);

?>
```

3. Go to the line starting with ***\$dataURL =*** and replace the URL with the URL of the RSS or Atom file that you would like read. In our example, the results should look like this:

```
<?php
    $dataURL = "http://www.flashloaded.com/rss/flashloaded.rss";

    //note that this will not follow redirects
    readfile($dataURL);

?>
```

4. Save this PHP file under a new name, we'll call it: flashloadedrss.php
5. Upload this new PHP file to your webserver.
6. Go back to your Flash file and edit your ActionScript code to refer to the proxy file that you uploaded to your webserver, for example:

```
RSS2.load(ftx, "flashloadedrss.php");
```

ActionScript properties

FlashTextFX.delay

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.delay:Number
```

Description

The delay (in seconds) before the start of the animation sequence.

Example

The following code defines delay.

```
myTextFX.delay = 1.2;
```

FlashTextFX.endPause

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.endPause:Number
```

Description

The pause (in seconds) after the end of animation sequence.

Example

The following code defines end pause.

```
myTextFX.endPause = 1.7;
```

FlashTextFX.pause

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.pause:Boolean
```

Description

Property; pauses/resumes the animation

Example

The following code will pause the animation.

```
myTextFC.pause = true;
```

FlashTextFX.sequence

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.sequence:Array
```

Description

Property; an array containing the HTML text of the text sequence.

Example

The following code defines text sequence.

```
myTextFC.sequence = ["First sentence", "second sentence"];
```

FlashTextFX.styleSheet

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.styleSheet:StyleSheet
```

Description

Property; the stylesheet to apply to the animated text.

Example

The following code defines style of the linked text:

```
import flash.text.StyleSheet;
var style:StyleSheet = new StyleSheet();
var anchor:Object = new Object();
anchor.fontWeight = "bold";
anchor.color = "#FF0000";
style.setStyle("a", anchor);
myTextFX.styleSheet = style;
```

ActionScript methods

FlashTextFX.restart

Availability

Flash Player 9
ActionScript 3

Usage

```
myTextFX.restart()
```

Parameters

None.

Returns

Nothing.

Description

Method. Restarts the animation.

Example

The following code will start animation from the beginning:

```
myTextFX.restart();
```

ActionScript events

FlashTextFX.sequence

Availability

Flash Player 9
ActionScript 3

Description

Event; Broadcast to all registered listeners when the sequence starts or ends.

Example

The following code will output "Start" and "End" text when sequence starts and ends.

```
import com.flashloaded.textfx.events.TextFXEvent
myTextFX.addEventListener( TextFXEvent.SEQUENCE, onSequence );

function onSequence( event:TextFXEvent ):void
{
    if ( event.start ) trace("Start");
    else if ( event.end ) trace("End");
}
```

FlashTextFX.step

Availability

Flash Player 9
ActionScript 3

Description

Event; Broadcast to all registered listeners when animation step starts or ends.

Example

The following code will output "Start" and "End" text when step starts and ends.

```
import com.flashloaded.textfx.events.TextFXEvent
myTextFX.addEventListener( TextFXEvent.STEP, onStep );
```

```
function onStep( event:TextFXEvent ):void
{
    if ( event.start ) trace("Start");
    else if ( event.end ) trace("End");
}
```

FlashTextFX.linkOver

Availability

Flash Player 9
ActionScript 3

Description

Event; Broadcast to all registered listeners when mouse is over hyperlinked text. Event object will contain object property 'link' with following parameters:

text:String - text string that is hyperlinked

url:String - value of href attribute of 'a' tag

target:String - value of the target attribute of 'a' tag

Example

The following code will output value of the link.

```
import com.flashloaded.textfx.events.TextFXEvent
myTextFX.addEventListener( TextFXEvent.LINK_OVER, onLinkOver );

function onLinkOver( event:TextFXEvent ):void
{
    trace( event.link.url );
}
```

FlashTextFX.linkOut

Availability

Flash Player 9
ActionScript 3

Description

Event; Broadcast to all registered listeners when mouse is out of hyperlinked text area. Event object will contain object property 'link' with following parameters:

text:String - text string that is hyperlinked

url:String - value of href attribute of 'a' tag

target:String - value of the target attribute of 'a' tag

Example

The following code will output value of the link.

```
import com.flashloaded.textfx.events.TextFXEvent
myTextFX.addEventListener( TextFXEvent.LINK_OUT, onLinkOut );

function onLinkOut( event:TextFXEvent ):void
{
    trace( event.link.url );
}
```

FlashTextFX.linkClick

Availability

Flash Player 9
ActionScript 3

Description

Event; Broadcast to all registered listeners when hyperlinked text is clicked. Event object will contain object property 'link' with following parameters:

text:String - text string that is hyperlinked

url:String - value of href attribute of 'a' tag

target:String - value of the target attribute of 'a' tag

Example

The following code will output value of the link, when linked text is clicked.

```
import com.flashloaded.textfx.events.TextFXEvent
myTextFX.addEventListener( TextFXEvent.LINK_CLICK, onLinkClick );

function onLinkOut( event:TextFXEvent ):void
{
    trace( event.link.url );
}
```

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:

[FlashTextFX 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.