

# CLASSES AND FLASH

## Sound Class

new Sound()

Creates sound object

new Sound (target)

Target reference either movieclip where sound is

### Properties

duration

Total length of sound (milliseconds)

position

Length of time the sound has played (milliseconds)

### Methods

attachSound()

associate sound from library with movieclip instance

getPan()

retrieve current sound pan

getVolume()

retrieve current sound volume

setPan()

set pan of sound

setVolume()

set volume of sound

start()

start playing of sound

stop()

silence sound

### Event Handlers

onLoad()

when sound finishes loading

onSoundComplete()

when sound finishes playing

## Color Class

new Color(target)

Target reference either movieclip where sound is

### Methods

getRGB()

retrieve current offset values for Red, Green, Blue

setRGB()

set current values of movieclip to Red, Green, Blue

## The anatomy of creating a class for sound and color

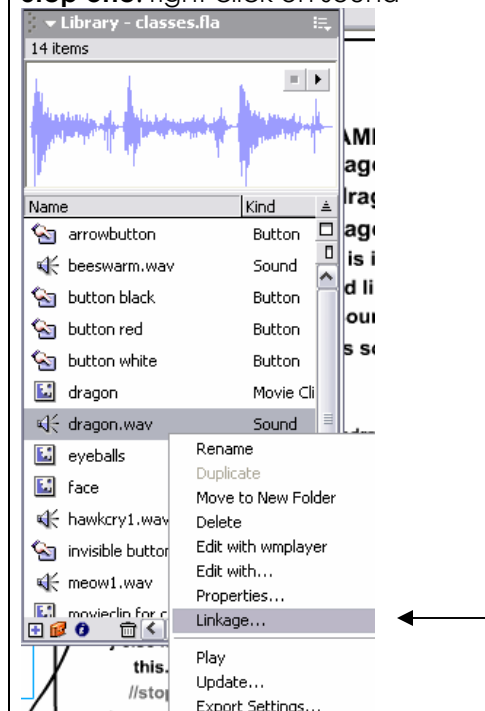
### Color

Placed in a frame	Comments
<pre>var circlecolor = new Color("mc");</pre>	<pre>//create a variable named circlecolor //stored color class information //is stored in circlecolor variable //target for any action is movieclip named 'mc'</pre>
Placed on button (in this case)	
<pre>on (release){     circlecolor.setRGB(0xFF0000); }</pre>	<pre>//note i am not putting instance name in script //hex values are used //0x signifies hex values</pre>

### Sound

Placed in a frame	Comments
<pre>makenewsound= new Sound(soundmc); makenewsound.attachSound("hawkcry");</pre>	<pre>//makenewsound is variable to control sound //hawkcry is identifier set up //changed linkage to sound in library //attachSound attached sound to movieclip on stage named 'soundmc'</pre>
Placed on button (in this case)	
<pre>on (release){     makenewsound.stop(); }</pre>	<pre>//starts sound playing</pre>

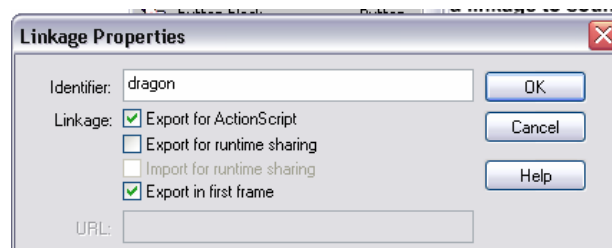
### Step one: right click on sound



### Below is the way you 'attachsound' from library

To be controlled by actionscript.

Step two: Choose linkage  
Choose export for actionscript  
Export in first frame



## TextField Class

**Must be a dynamic or input text box and have an instance name assigned to it.**

### properties

**\_alpha**  
if embedFonts used then sets opacity

**background**  
on or off

**backgroundColor**  
rgb number specifying background color (0x000000)

**border**  
on or off

**borderColor**  
rgb number specifying border color (0x000000)

**embedFonts**  
if embed characters is set before export renders font in box (true, false-aliased device)

**\_height**  
height of text field bounding box in pixels

**length**  
number of characters in box

**multiline**  
enables or disables multi-line

**\_name**  
instance name

**password**  
obscures user input

**\_rotation**  
in degrees

**selectable**  
enables or disables user select

**text**  
specifies characters to be rendered in field

**textColor**  
sets color for all letters in field

**textHeight**  
pixel height of all text in field

**textWidth**  
pixel width of longest line of text

**type**  
accepts user input (true, false value)

**variable**  
associates a variable with text

**\_visible**  
shown or hidden

**\_width**  
width of text box in pixels

**wordwrap**  
true or false – allows word wrap with long lines

**\_x**

horizontal position of field  
\_xscale  
width of text field as percentage  
\_y  
vertical location of box  
\_yscale  
height of text field as percentage

---

## examples

testme.text= "100";  
puts the number 100 in a text field instance named 'testme'

testme.embedFonts=true;  
testme.\_alpha=30;  
if fonts are embedded then embed fonts renders them antialiased and sets the alpha of the box to 30%

## How to Embed fonts for text field (input or dynamic only):

