

# Flash MX Audio Workshop

## Using Audio in Flash MX

In this workshop, you will learn how to use Macromedia Flash to stream audio over the Internet.

You will learn how to:

- Create a Sound object;
- Import audio files and link them to the Sound object;
- Create playback controls for the Sound object;
- Export the Shockwave Flash file so that it can be viewed in a browser.

You can follow the same steps for both a horizontal or vertical volume slider. There is only one difference: The document size. Other than that everything is exactly the same. Once finished you simply rotate the vertical slider to convert it to a horizontal slider. None of the ActionScript or other details need to be changed.



### Step one: Setting Up the Document

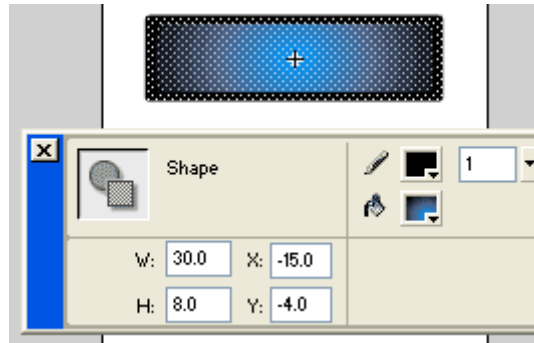
1. Open a new Flash file.
2. Go to: **Modify > Document**
3. Set the size to: **40 x 140 pixels**
4. Click: **OK**

This is a suggestion. You can actually set the document to any size it is most appropriate for your movie.

### Step two: Creating the Slider

First we will create the horizontal volume control bar: 

1. Go to: **Insert > New Symbol**
2. Name the symbol: **Control Bar MC**
3. Behavior: **Movie Clip**
4. Click: **OK**
5. Using the Rectangle tool: **Draw a rectangle** 
6. When you have finished drawing the rectangle return to your Arrow tool: 
7. If the Property Inspector is closed, Open it: **Window > Properties**
8. In the Property Inspector select any colour you may want for the stroke and fill. Mine has a black outline with a gradient fill.
9. In the Property Inspector set the size as follows: **30 x 8 pixels**
10. Center the rectangle: **x = -15 y = -4**

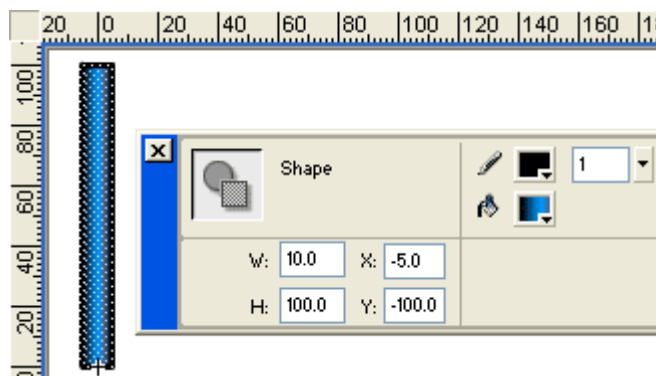


1. *Details in the Property Inspector.*

It is now time to create the vertical slider bar:

2. Go to: **Insert > New Symbol**
3. Name the symbol: **Slider MC**
4. Behavior: **Movie Clip**
5. Click: **OK**
6. Using the Rectangle tool: **Draw a rectangle**
7. When you have finished drawing the rectangle return to your Arrow tool:
8. In the Property Inspector select any colour you may want for the stroke and fill.
9. In the Property Inspector set the size to: **10 x 100 pixels**
10. Center the rectangle on the horizontal x axis: **x = -5**
11. On the vertical y axis the bottom of the slider must be located on the registration point: **y = -100**

It is -100 because Flash measures from the top of the rectangle down.



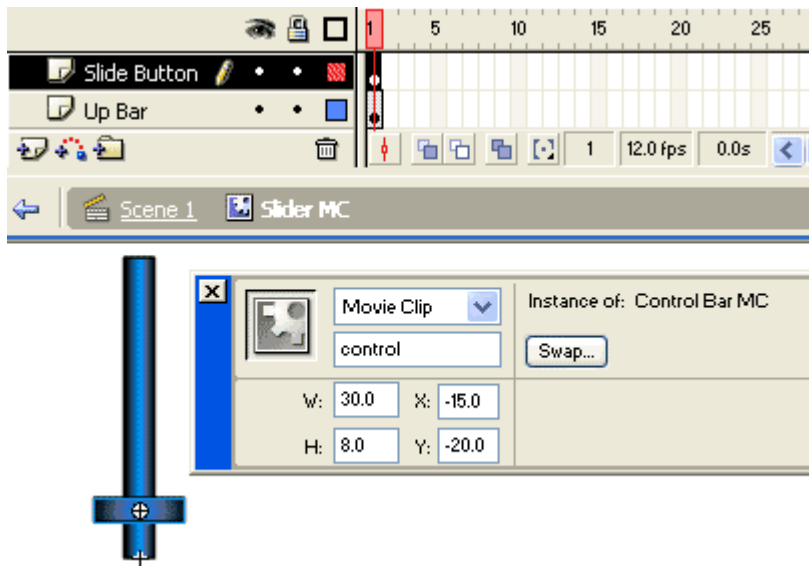
*Details in the Property Inspector. Note at the bottom of the slider the position of the registration point: +*

It is now time to add the Control Bar MC.

12. Re-name Layer 1: **Up Bar**
13. In the Timeline click the Insert Layer Button:
14. Name the new layer: **Slide Bar**
15. Open the Library: **Window > Library**
16. From the Library drag onto Stage the symbol: **Control Bar MC**
17. Center the symbol on the horizontal x axis: **x = -15**

**Note:** The position of the symbol on the y axis is not important as it is controlled by ActionScript.

18. In Property Inspector give the symbol, Control Bar MC an Instance Name: **control**



*The completed Slider.*


The slider is now completed. Go back to the main Stage by clicking on the Tab:



### Step Three: The Main Stage - Slider

The Main stage should still be empty. We need two buttons and the slider on stage.

1. If the Library is closed, open it: **Window > Library**
2. Drag onto Stage the Symbol: **Slider MC**
3. In the Property Inspector give the symbol an Instance Name: **slider**

**Note:** If the slider is not the right size you can simply resize it on the main Stage with the Free Transform tool: 

**For Horizontal Sliders only:**

- a. Go to: **Window > Transform**
- b. In the Rotate box type: **90°**
- c. On your keyboard press: **Enter** (Mac Return)




*After the Slider is rotated.*

## Step Four: The Main Stage - Stop Button

**Note:** If you do not wish to make your own buttons use some buttons from the Common Library. You will still need to give them an Instance Name (step 4.9 and step 5.14): **Window > Common Library > Buttons**

If you take a button from the Common Button Library avoid Knobs, Faders and Component buttons as they work differently.



1. Use the Rectangle tool to: **Draw a square** ■  
My square is: 12 x 12 pixels
2. When you have finished drawing the rectangle return to your Arrow tool: 
3. Select any colour you may want for the stroke and fill.
4. Double click in the center to select the rectangle.
5. With the square selected right click (Mac Ctrl click) and select: **Convert to Symbol**
6. Name the symbol: **Stop**
7. Behavior: **Button**
8. Click: **OK**
9. In the Property Inspector give the button an Instance Name: **stopButton**

## Step Five: The Main Stage - Play Button

The following are only suggestions to test your buttons are working. Please, create original buttons (nicer than those ones!)

The easiest way to draw a triangle is to create a square and convert it so a triangle.

Draw a square that is slightly larger than the previous one.  
Mine is 15 x 15 pixels.

1. When you have finished drawing the rectangle return to your Arrow tool: 
2. Select any colour you may want for the stroke and fill.
3. With the Sub-selection tool  go to the rectangle and click on the: **top right hand corner**




*Top right corner is selected.*

4. On your keyboard press: **Delete**



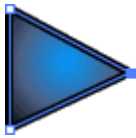
*You should now have a right angled triangle.*

5. With the Sub-selection tool  go to the rectangle and click on the: **bottom right hand corner**




*Bottom right corner selected.*

6. On your keyboard press the up arrow until the bottom corner is centred.



*Completed triangle.*

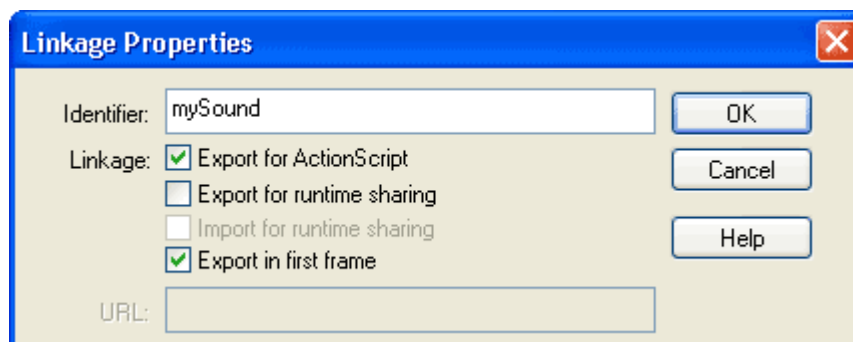
7. Go back to the Arrow tool: 
8. Double click in the centre to select the triangle.
9. Right click and select: **Convert to Symbol**
10. Name the symbol: **Play**
11. Behaviour: **Button**
12. Click: **OK**
13. In the Property Inspector give the button an Instance Name: **playButton**

All the objects on Stage are now complete.

### Step six: The Sound file

You now need a short sound file to loop. It can be an: MP3, WAV or AIFF file.

1. Go to: **File > Import to Library...**
2. Select the file of your choice.
3. If the Library is closed, open it: **Window > Library**
4. Right click on the sound file and select: **Linkage**
5. For Linkage select: **Export for ActionScript**
6. For Identifier type: **mySound**
7. Click: **OK**



*The Linkage settings.*

Everything is now done except for the ActionScript in frame one.

## The ActionScript

Place the following ActionScript in frame 1:

```
myMusic = new Sound(this);
myMusic.attachSound("mySound");
myMusic.start(0, 1);
slider.control._y = -50;
slider.control.onEnterFrame = function() {
    myMusic.setVolume(0-this._y);
}
slider.control.onPress = function() {
    startDrag(this, false, this._x, -100, this._x, 0);
}
slider.control.onRelease = function() {
    stopDrag();
}
stopButton.onRelease = function() {
    myMusic.stop();
}
playButton.onRelease = function() {
    myMusic.start(0, 1);
}
```

You will want to test your movie as it is now all done.

## ActionScript Explained

```
myMusic = new Sound(this);
```

Makes a new sound object in the timeline. This sound object does not yet contain any sound yet. It is like a CD player with out a CD.

```
myMusic.attachSound("mySound");
```

This attaches the sound from the Library which we called "mySound".

```
myMusic.start(0, 1);
```

This is an auto-start. It instructs the sound file to start to play. (The files at the top of this page do not have this - on my files you must hit the play button). Once the file starts to play it will loop 1 time.

```
slider.control._y = -50;
```

Sets the initial volume to 50% full volume. 0 is off and -100 is full volume. Remember that -100 is the top of the slider scale (step 2.20 above).

```
slider.control.onEnterFrame = function() {
    myMusic.setVolume(0-this._y);
}
```

Sets the volume to be what every the y value of slider control. As we slide the controller up and down the volume goes according to the y value. The(0-this.\_y)converts all the negative numbers to positive so the volume scale is actually 0 to 100 (not 0 to -100).

**Note - Rotation:** What is strange is that if you rotate the slider on the main stage to make a horizontal control - the symbols nested inside are not considered to have been rotated. This means that the y value still works! You do not need to convert the ActionScript to an x value.

**Note - Size:** This same is also true of size. If you change the size of the controller on the main Stage the `y` scale inside the re-sized controller is still 0 to -100. The sizing on Stage makes no difference to measurements of objects nested inside a symbol.

```
slider.control.onPress = function() {  
    startDrag(this, false, this._x, -100, this._x, 0);  
}
```

This makes the slider move when it is dragged with the mouse. Its movement is restricted between 0 and -100.

```
slider.control.onRelease = function() {  
    stopDrag();  
}
```

When you let go of the mouse the slider will stop moving.

```
stopButton.onRelease = function() {  
    myMusic.stop();  
}
```

When you click on the stop button it will stop the sound from playing.

```
playButton.onRelease = function() {  
    myMusic.start(0, 1);  
}
```

When you click on the play button the sound file will play once.

That's it! You should have a great volume control.