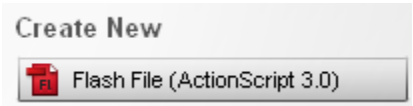
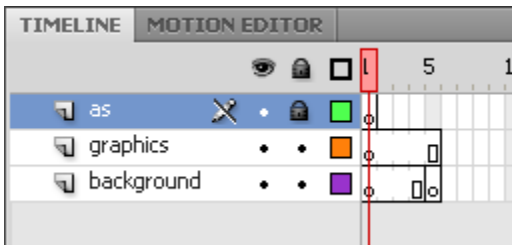
Creating Basic Action Script Buttons with Different Sections

Step 1 – Create a new Action Script 3.0 Document



Step 2 – Set up your timeline

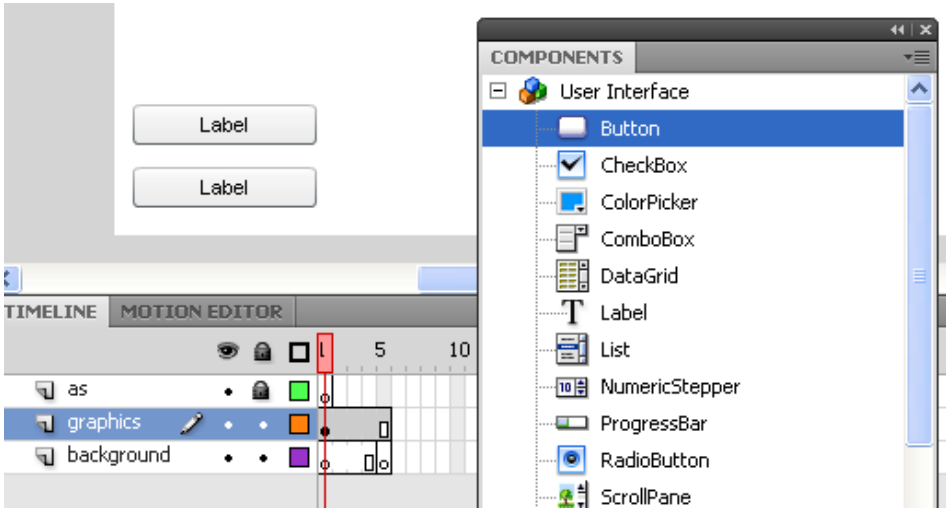


Layer 1 is for Action Script, lock this layer so you don't accidentally put any graphics on it.

Layer 2 will contain your graphics. ie: buttons, images, input fields

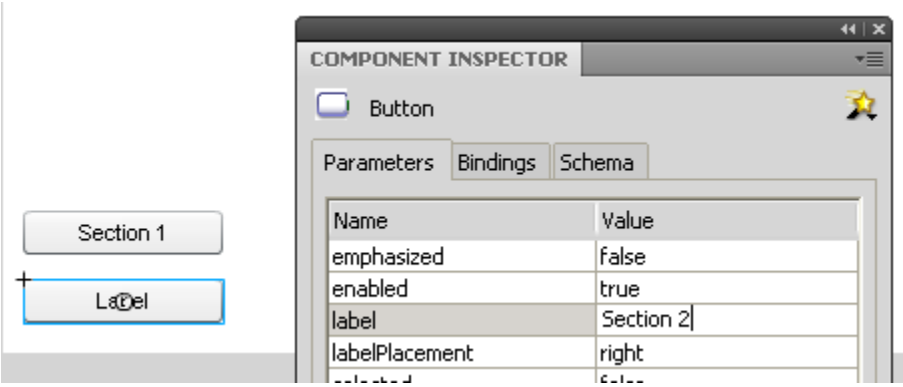
Layer 3 will contain your background image or color

Step 3 – Bring buttons onto your stage



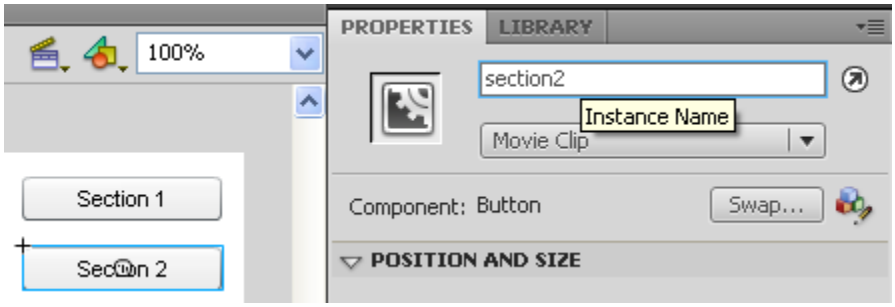
Using your Components (Control + F7 or Window > Components) drag out 2 buttons to the stage on the graphics layer. When finished, pull up your Components Inspector (Shift + F7 or Window > Components Inspector).

Step 4 – Label your buttons



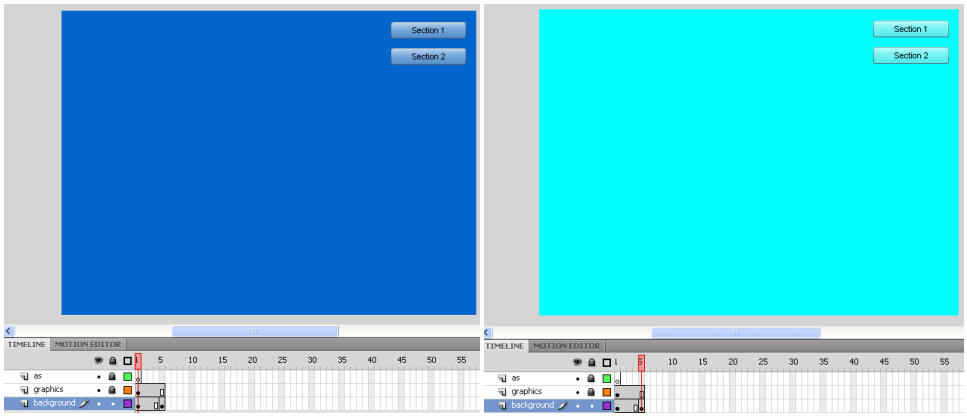
Using the Components Inspector (see step 3 to open window) change the label name of the button to “Section 1” and “Section 2”. First, click on the button. Then click in the Value section next to the “label” Name and change the text.

Step 5 – Assign instance names



After selecting your button, click on the Properties Tab. Then change the Instance Name field to “section1” and “section2”, respectively. This instance name will be referenced in Action Script to assign an action to the buttons.

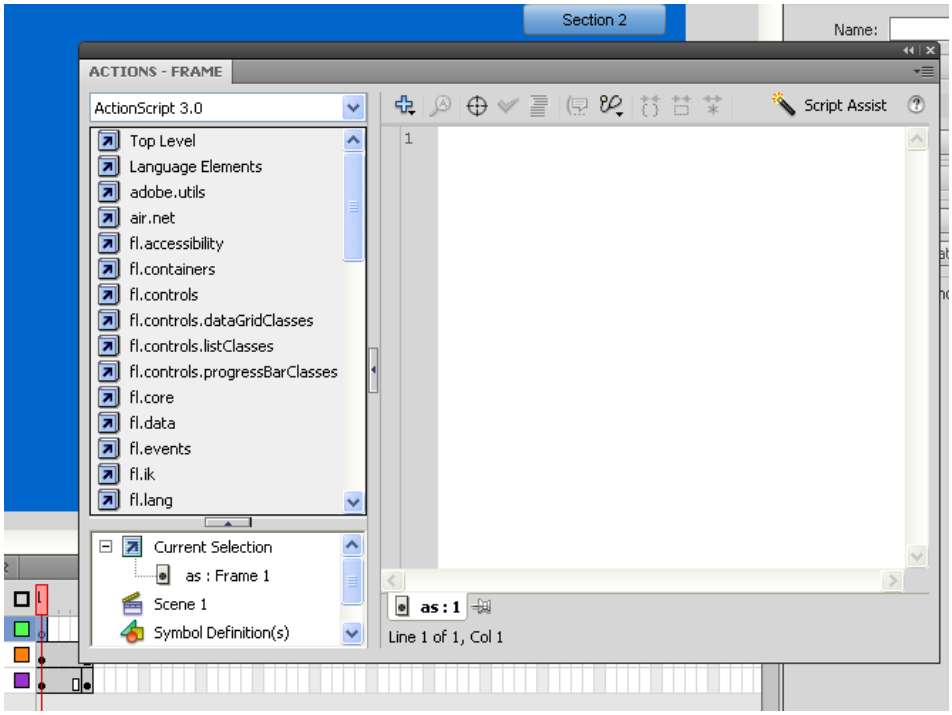
Step 6 – Set up different background colors



Using the rectangle tool, draw out two different background colors. The first on Frame 1 of the Background layer, then navigate to Frame 5 and draw a different color on the keyframe.

Step 7 – Add in Action Script 3.0

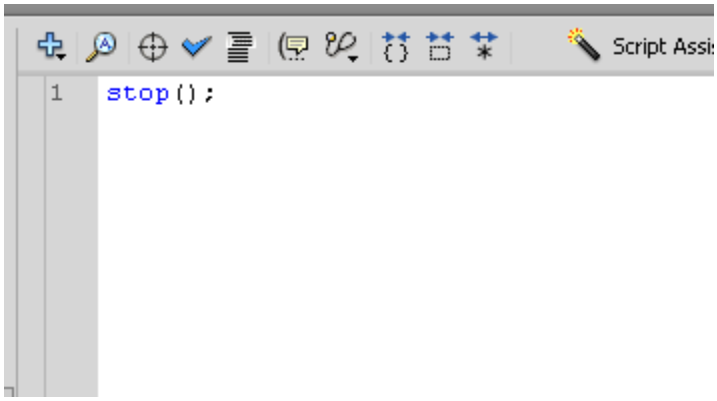
Select Frame 1 of the as layer then hit F9 on your keyboard to open the Actions Panel.



First we will type “stop();” into the Actions Panel. This will tell Flash to stop on the first frame and not continue playing through the Timeline.

Step 8 – Add button interaction

Your Actions Panel should look like this:



To get the buttons interacting with the stage we will need to first reference the Instance Name we assigned in Step 5. On the first button we called it “section1”, so the AS to reference that button would look like this:

section1.addEventListener();

“.addEventListener();” tells Flash we are waiting for an event to happen to section1. What event that is needs to be defined inside of the parenthesis. For this particular button we will use the following:

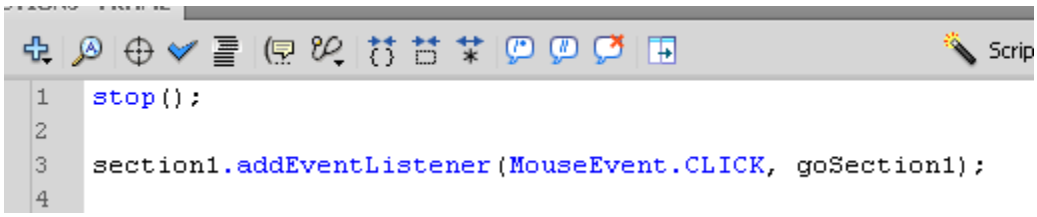
section1.addEventListener(MouseEvent.CLICK, goSection1);

“MouseEvent” tells Flash we will be utilizing a mouse functionality, in this case “CLICK” (when the user clicks their mouse). Other events include MOUSE_OVER, MOUSE_DOWN, etc…

“goSection1” is a function we will set up. Functions tell the mouse event what they need to do.

Step 9 – Define a function

Your Actions Panel should look like this:



To define a function you first need to declare the function and give it a name. In Step 9 we are looking for the function named goSection1, so we will set up a function with that name:

function goSection1(e.MouseEvent):void {

}

“function” tells Flash we are defining a function, the following text “goSection1” is the name we give the function and can be anything just as long as it matches the name we are referencing in our event listener.

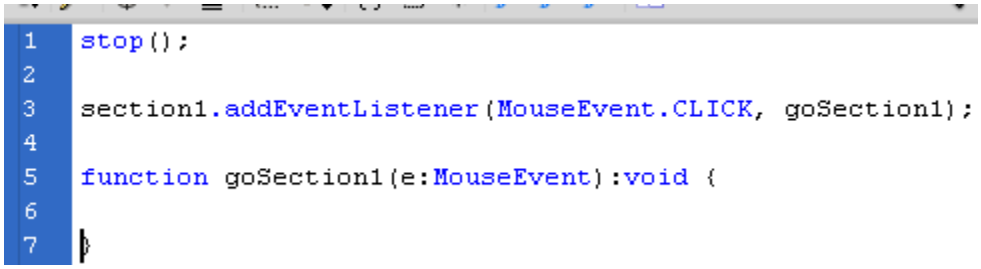
“(e:MouseEvent)” tells Flash we are referencing a mouse event in our event listener (as defined in Step 8)

“:void” is pretty useless. It simply tells Flash we’re not passing any information and is required for the function to work. We will need this after the open/close parenthesis.

“{ }” (open and close brackets) Everything inside of these will run when the function is called. We will utilize this function to go to another frame on the timeline and stop at that frame.

Step 10 – Adding an event

Your Actions Panel should look like this:



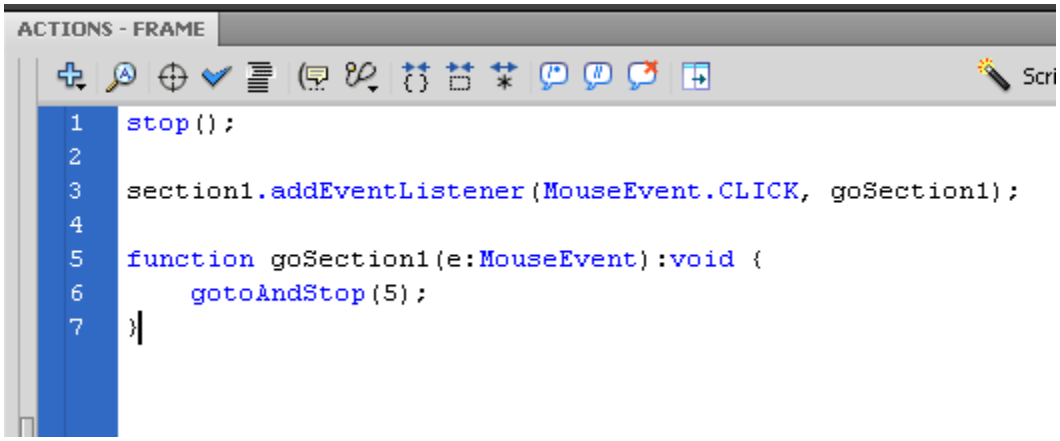
Inside of the open/close brackets add the line:

“gotoAndStop(5);”

This will send your playback head on the timeline to frame 5 and will stop there when the section1 button is clicked.

Step 11 - Moving forward

Your final Actions Panel should look like this:



From here you can repeat Steps 8-10 to set up any additional buttons. Simply swap out the function name and create a new function for each button and tell it to go to a different frame.