

Importing Graphics and Text with ActionScript 3

Step 1 – Set up your timeline

After you've created your new ActionScript 3.0 document, we are going to make our timeline look like this:

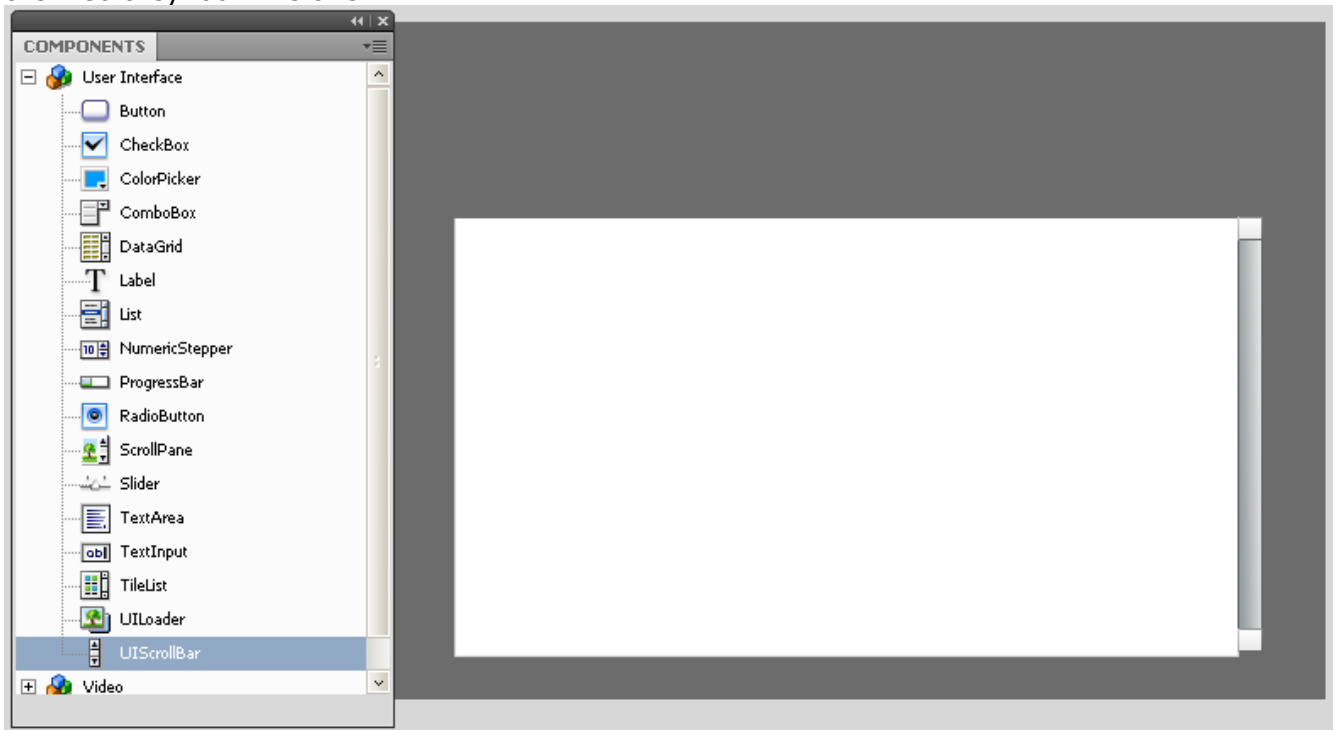


Layer 1 is for ActionScript. Lock this layer so you don't accidentally put any graphics on it.

Layer 2 will contain your textarea and scroll bar we will set up in Step 2.

Step 2 – Set up your stage

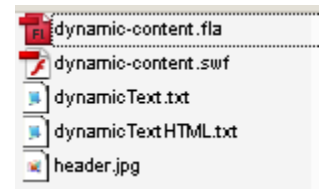
Using your Components (Control + F7 or Window > Components) drag out a TextArea and UIScrollBar onto the stage on Frame 1 of the text box layer. Use your Free Transform tool to resize them so they look like this:



(I made my stage background darker so you can see the graphics better)

Step 3 – Set up external files

You will need 3 different files for this tutorial. An image and 2 text files, 1 with plain text and 1 with HTML formatting. I have all of these files, plus the Flash source files available at: <http://school.smple.com/resources/IMED-1345-Week11.zip>. Be sure all of these files are in the same folder when working with this example.



Step 4 – Load in the graphic

The first thing in loading a graphic is to define a variable (or loader) to hold the image. Right now you're probably thinking, "???". Here's the best explanation I can give:

When you want to store something to use later or ship to a friend you put it in a box. Consider variables your box. We are basically going to store information in that box.

The first 2 lines of ActionScript will look like this:

```
// DEFINE A VARIABLE TO HOLD THE IMAGE  
var headerLoader:Loader = new Loader();
```

The first is a comment. Here is a break down of line 2:

"var" - Tells Flash we are creating a variable.

"headerLoader" - The name of our variable (can be anything).

":Loader" - Assigning the variable to a certain data type. Data types help tell Flash what the variable is supposed to contain and is used in debugging.

"= new Loader();" - Tells Flash this particular variable will be used to load in external information.

The next 2 lines are:

```
// CREATE A URL REQUEST TO POINT TO THE IMAGE  
headerLoader.load(new URLRequest("header.jpg"));
```

"headerLoader" - References the headerLoader variable we defined on line 2.

".load" - Tells Flash we will be loading in information to the variable headerLoader.

"(new URLRequest("header.jpg"))" - Tells Flash the new information we are loading in will call to an external URL. In this case the URL is header.jpg and will store that file in the headerLoader variable.

The next 2 lines are:

```
// ADD THE IMAGE TO THE STAGE  
addChild(headerLoader);
```

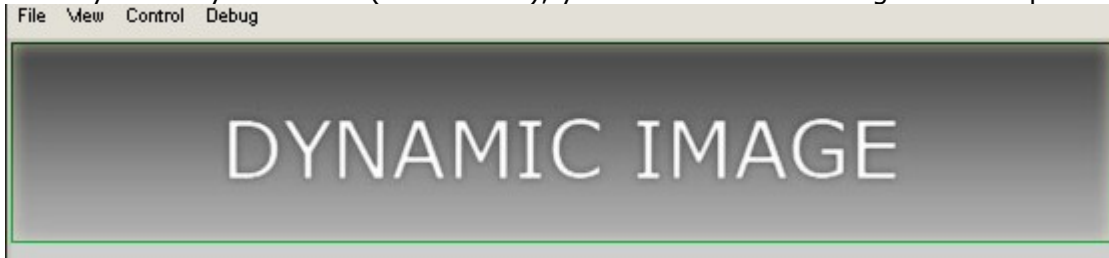
"addChild();" - Your stage is considered the "parent". When you add a child you are adding a dynamic element to the parent element, in this case, the stage.

"headerLoader" - The name of our variable that is storing the image header.jpg. When it is added to the stage using addChild(); it retrieves the information stored in it and puts it on the stage.

Right now your code should look like this:

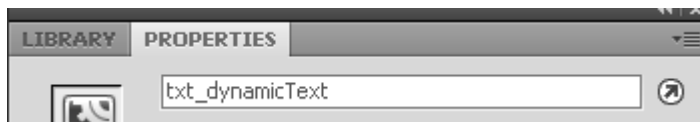
```
1 // DEFINE A LOADER TO HOLD THE IMAGE  
2 var headerLoader:Loader = new Loader();  
3 // CREATE A URL REQUEST TO POINT TO THE IMAGE  
4 headerLoader.load(new URLRequest("header.jpg"));  
5 // ADD THE IMAGE TO THE STAGE  
6 addChild(headerLoader);  
7
```

When you test your movie (Alt + Enter), you should see the image at the top of the stage like:



Step 5 – Place the TextArea on the stage

Click the TextArea on the stage and under your Properties Panel, change <Instance Name> to "txt_dynamicText" (without the quotes).



Step 6 – Load in dynamic text (setting up the variable)

The same as Step 4, we will need to set up a variable to hold the dynamic content:

```
// DEFINE A LOADER TO HOLD THE TEXT FILE
var textLoader:URLLoader = new URLLoader();
// CREATE A URL REQUEST TO THE TEXT FILE
textLoader.load(new URLRequest("dynamicText.txt"));
```

If you'll notice, we changed the name of the variable to textLoader and instead of loading in an image, we're loading in a .txt file. Your code should look like this:

```
8 // DEFINE A LOADER TO HOLD THE TEXT FILE
9 var textLoader:URLLoader = new URLLoader();
10 // CREATE A URL REQUEST TO THE TEXT FILE
11 textLoader.load(new URLRequest("dynamicText.txt"));
```

Step 7 – Add the text to the TextArea

In this step we will use the `addEventListener` we used when creating buttons. This time we want to "listen" for the text file to load. Once the file is completely loaded, we will display it in the `TextArea`. Here's the code:

The first 2 lines are:

```
// RUN THE FUNCTION TO LOAD THE TEXT IN THE TEXT FIELD
textLoader.addEventListener(Event.COMPLETE, textLoadComplete);
```

"textLoader" - The name of the variable holding the `dynamicText.txt` file.

".addEventListener();" - Tells Flash to wait for an event to be completed.

"Event.COMPLETE," - When the `textLoader` has completely loaded, the `COMPLETE` event will trigger and run the assigned function.

"textLoadComplete" - The name of the function to run when the event is triggered.

The next 3 lines are:

```
function textLoadComplete(event:Event):void {
    txt_dynamicText.text = textLoader.data;
}
```

"function" - Defines a function.

"textLoadComplete" - The name of the function, referenced in the `textLoader` event listener.

"(event:Event):void {}" - Completes the function. Any code written inbetween the `{}` will be executed when the function is run.

"txt_dynamicText" - References the `TextArea` instance name we defined on the stage.

".text" - Tells Flash we are looking at the text inside of the `TextArea`.

"= textLoader.data;" - Sets the text inside of `TextArea` equal to the variable that we set up to hold that dynamic text. ".data" grabs the information stored inside of the variable, in this case the "dynamicText.txt" file.

Your final code should look like this:

```
1 // DEFINE A LOADER TO HOLD THE IMAGE
2 var headerLoader:Loader = new Loader();
3 // CREATE A URL REQUEST TO POINT TO THE IMAGE
4 headerLoader.load(new URLRequest("header.jpg"));
5 // ADD THE IMAGE TO THE STAGE
6 addChild(headerLoader);
7
8 // DEFINE A LOADER TO HOLD THE TEXT FILE
9 var textLoader:URLLoader = new URLLoader();
10 // CREATE A URL REQUEST TO THE TEXT FILE
11 textLoader.load(new URLRequest("dynamicText.txt"));
12 // RUN THE FUNCTION TO LOAD THE TEXT IN THE TEXT FIELD
13 textLoader.addEventListener(Event.COMPLETE, textLoadComplete);
14
15 function textLoadComplete(event:Event):void {
16     txt_dynamicText.text = textLoader.data;
17 }
```

Step 8 – Using HTML in your text file

Simply set up the HTML inside of your text file. Inside of your ActionScript you will need to make the following changes:

On line 11:

Reference your new file. If you are using the sample file it is, "dynamicTextHTML.txt".

On line 16:

Change

txt_dynamicText.text

To

txt_dynamicText.htmlText

That's it! Shoot me an email if you have any questions at john@smple.com.