

Arrays

In this tutorial we'll take advantage of the power of Arrays to build a news site's story navigator. At its most basic level, an Array can hold multiple individual data values. In this case, we'll be creating 4 Array objects, 1 for each major section of the news site: news, sports, politics, and weather. Of course you could add a bunch more than this once you learn how to implement Arrays into your project.

An array is a general-purpose container. It can contain any number of items and even items of different types. A chest with drawers is analogous to an Array. An individual drawer holds content. The chest doesn't hold the content, it holds the drawers. We build arrays to access the information in the drawers within the chest.

The syntax for Arrays is the following:

```
arrayName = new Array( );  
arrayName[0] = "Some string of text";  
arrayName[1] = "Another string of text";  
arrayName[2] = "Our last string of text";
```

The Array name can be just about anything you want it to be and then we construct the Array object with the new Array() command. The parentheses can contain parameters for the Array. The name of the Array is used again and then assigned a number value starting with zero (Whenever counting in loops or Arrays, Flash usually begins with zero) and then incrementing upwards. Then the contents of the Array item follows, which in this case is a headline for each of our categories. This string of text could also be a set of numbers as well.

There are other ways to construct Arrays in ActionScript and if your interested you can take a look at the Reference Panel in Flash or the ActionScript Library in the Help Menu.

Objective

We'll be using an Array to store our headlines for each of the 4 main categories for the news site. All of the Array items will exist on frame 1 of the Timeline and can be accessed anywhere on that timeline. We'll then use Dynamic Text boxes with variable names attached to them that will access the data from the Arrays and display it on screen. We'll also take advantage of the new ability in Flash to load JPEG files with the loadMovie command. This is a welcome addition to Flash as before this version we would have to import a JPEG into a Flash movie and then export it as a SWF file in order for it to be imported. Now we can just point to the JPEG file itself and use a blank Movie Clip as the target destination for the JPEG file.

The Data

1. Open up the File arrays.fla. The layout is created for you already, you'll be adding the functionality by adding the scripts to the Movie.

2. Click on Frame 1 of the Actions Panel. Be sure that you are in the Expert Mode of the Actions Panel by Clicking on the icon in the upper right-hand corner of the Actions Panel and choosing Expert. Enter the following code:

```
// News Array
news = new Array(); {
}
```

3. Here we use a comment to remind us which Array we are building – the News Array. Next, we create the news Array with the Array constructor: `news = new Array();` Finally, we open up a set of curly braces to contain all of our Array values.
4. Insert the following code inside the curly braces in the Actions Panel:

```
news[0] = "Bush talks tough on North Koreas Nuclear Program.";
news[1] = "UN Inspectors report on weapons program in Iraq.";
news[2] = "Israeli Astronaut steps out of Columbia.";
```

5. Your full code should look like this:

```
// News Array
news = new Array(5); {
news[0] = "Bush talks tough on North Koreas Nuclear Program.";
news[1] = "UN Inspectors report on weapons program in Iraq.";
news[2] = "Israeli Astronaut steps out of Columbia.";
}
```

6. Use the chart below to add the other 3 Arrays:

Sports Array
<pre>// Sports Array sports = new Array(); { sports[0] = "Tampa Bay Super Bowl Champions with Ease."; sports[1] = "Fed Cup to come to Lowell."; sports[2] = "Pierce Weekend: 2-3."; }</pre>
Politics Array
<pre>// Politics Array politics = new Array(); { politics[0] = "Inspections yield no proof Iraq rearming."; politics[1] = "Bush appears on irreversible path to war."; politics[2] = "North, South meets over Korea nuke crisis."; }</pre>
Weather Array
<pre>// Weather Array weather = new Array(); { weather[0] = "Cold temperatures to continue all week."; weather[1] = "Frigid weather in Atlanta causes school</pre>

```
closings." ;  
}
```

I only built a maximum of 3 headlines because there will be 3 Dynamic Text fields per screen to hold the Array values, you could obviously do more.

Loading the Data from the Arrays

1. Back on the Main Timeline, examine the 3 Dynamic Text fields on the stage in Frame 1. Each has a Variable name attached to it: news1, news2, and news3. Take a look in Frame 5, 10, and 15 and notice there are 3 Dynamic Text Fields for each of these areas. Each of these have Variable names attached to them as well. We'll use ActionScripts dot syntax to assign the data from the Arrays to the appropriate Variable name for each Dynamic Text box.
2. Click on Frame 1 of the Actions layer again. In the Actions Panel, enter the following script below the Arrays:

```
// Get the headlines for News  
_root.news1 = news[0];  
_root.news2 = news[1];  
_root.news3 = news[2];  
// Get the headlines for Sports  
_root.sports1 = sports[0];  
_root.sports2 = sports[1];  
_root.sports3 = sports[2];  
// Get the headlines for Politics  
_root.politics1 = politics[0];  
_root.politics2 = politics[1];  
_root.politics3 = politics[2];  
// Get the headlines for Weather  
_root.weather1 = weather[0];  
_root.weather2 = weather[1];  
_root.weather3 = weather[2];
```

By now you recognize the Comments. I'll use the first line of code in the news section as an example. First, we identify the Variable text field the data from the Array will go into, in this case news1, which happens to be on the root level or main Timeline, therefore, we use the _root command to identify the absolute location of that Dynamic Text field. Then we assign the data value from the Array, in this case news[0]. Its almost easier to understand if you read the statement backwards: Take the data from the news Array in position zero (0) and place it into the Dynamic Text field with the Variable name news1, which is on the main Timeline _root. All the other commands are similar.

3. We'll finish the scripting for Frame 1 of the Actions Panel now. Add the following code:

```
// load image for news  
loadMovie("bush.jpg", "placeholder");  
// Prevent the Timeline from Moving
```

```
stop();
```

Here is where we use the loadMovie command to load the image “bush.jpg”, which is in the same folder as the array.fla file. Then we set the target to “placeholder”, which happens to be the Instance name of a blank Movie Clip on the stage which is the small white circle you can see under the Dynamic Text fields. The stop() action prevents the Timeline from moving forward so we can stay on the News section until we click one of the menu buttons to move to another section. We’ll use this script again for the next 3 sections.

4. Highlight the last 2 lines of code as seen above in step 3. From the Edit Menu choose Copy.
5. Click on Frame 5 of the Actions layer and move your cursor into the ActionScript Panel. From the Edit Menu choose paste. Change the code for the Sports section as follows:

```
// load image for sports
loadMovie("pierce.jpg", "placeholder");
// Prevent the Timeline from Moving
stop( );
```

6. Click in Frame 10 of the Timeline. In the ActionScript Panel, paste the script again and change it for the Politics section:

```
// load image for politics
loadMovie("uninspectors.jpg", "placeholder");
// Prevent the Timeline from Moving
stop();
```

7. Click in Frame 15 of the Timeline. In the ActionScript Panel, paste the script again and change it for the Weather section.

```
// load image for weather
loadMovie("weather.jpg", "placeholder");
// Prevent the Timeline from Moving
stop();
```

NOTE: The separation on the Timeline is not necessary for the Movie to function. Its merely a visual treatment to make editing easier.

The Menu

1. Notice the Frame Labels in the Labels layer for each section. We will use those labels to make our navigation bar buttons functional. The advantage of using labels instead of frame numbers when scripting is that if you decide later to move the content from frame 5 to frame 15, for example, the script won’t be affected because it is targeting the Label. If the script targeted a specific Frame, you’d have to change the code.
2. Click on the News button at the left of the navigation bar at the top of the stage. Be sure to select the button and not the text label over the button.



3. In the ActionScript Panel, add the following script:

```
on (release) {
    gotoAndStop("news");
}
```

The on (release) statement is the Event Handler that requires the mouse button to be released before the action takes place. The gotoAndStop action will move the Timeline to the frame in the parentheses, in this case, not a Frame, but a Frame Label “news”. Note that we are using a string literal for the Frame Label and not an expression. We literally want to move the timeline to that Frame.

4. Script the remaining buttons as follows:

Sports Button
<pre>on (release) { gotoAndStop("sports"); }</pre>
Politics Button
<pre>on (release) { gotoAndStop("politics"); }</pre>
Weather Button
<pre>on (release) { gotoAndStop("weather"); }</pre>

Testing the Movie

From the Control Menu choose Test Movie. You should initially see the News section headlines and the news image. With the click of the buttons, the data from the Array in Frame 1 is retrieved and placed into the appropriate Dynamic Text fields.



Other Possibilities

5. Go back to Frame 1 and change the headlines.
6. Test the Movie to see the changes.
7. Download some JPEG files from the internet and size them down to 105 pixels in height. Save them to the same folder as the arrays.fla Movie. Change the loadMovie action on the section frames to load different images into the Movie.

In this example, you've learned how to create Arrays and access the data in the Arrays. This is not much different from database interactivity. You could even save an ActionScript file that

contains just the Arrays, then include the Arrays in the first Frame of the movie with the include action:

```
#include "filename.as"
```

That way you could edit the headlines in the .as file instead of having to edit the script in the Movie with a text editor.

Resources: *ActionScript the Definitive Guide* by Colin Moock. O'Reilly Books. ISBN: 1-56592-852-0.