

The Display List

Flash keeps a list of all the elements in a Movie. This list is called the "display list." At the root of the display list is an instance of the **Stage** class. (Notice that "Stage" is capitalized: that's because here we are referring to Stage as a the name of a class. Class names are always capitalized.) This instance of the Stage class is special. It acts as a box to hold our artwork and animation. It also sets the global characteristics of the movie, such as frame rate, dimension, etc.

Whenever you place an object on the stage, the object becomes (in essence) a **child of the stage**. The new object is added to the display list. If you give the object an instance name, Flash will honor it. If you don't, Flash will assign it one. All objects on the stage have unique names. Flash will also assign it a unique depth. You can think of depth as a way objects layer, each one on top of the other. Conceptually it is similar to the Arrange command in a graphics program, where you use the "Bring to Front" command, or the "Push to Back" command.

The Loader Class

ActionScript provides a handy and consistent way of loading external files of all kinds. It's a bit of work, but once you learn it, it is always done the same way. You can load in all kinds of file types. Some of the more popular ones are SWFs, images (JPGs, GIFs), music (in the form of MP3s), video (in the form of FLVs), data (in the form of TXT files or XML files) ... the list goes on and on.

Today we are loading in JPGs to your Galleries. These are visual file types that are added to the display list. Remember: the stage has a parent-child relationship with anything displayed on it.

To load in a SWF, we need an instance of the Loader class. The Loader class does exactly what it says it does -- it loads external files. It will load whatever you tell it to load, and track how the load progresses as well. We store the new instance of the Loader class in a variable. We create the variable by using the "var" keyword. Here is how we make a new loader:
var theLoader = new Loader();

Now that we have a new loader, we want whatever we put in it to be visible. The only way to make it visible is to add it to the display list. We do that by adding the loader as a child of the stage, like this:

addChild (theLoader);

We put our two lines of code together, and we get this (all blue text are reserved words in Flash that have meaning already):

```
var theLoader = new Loader();  
addChild(theLoader)
```

In our case we want add this loader to a particular place on the Stage – inside of our empty movie clip built for this purpose—**load_area_mc**

```
var theLoader = new Loader();  
load_area_mc.addChild(theLoader)
```

then we want the Loader to load the correct file to the Slideshow by giving it an URLRequest:

```
theLoader.load(new URLRequest(pathToFile));
```

in order to reuse the Loader easily we can create a new function with a custom parameter that we can call any file or image by specifying the path to the image when we use the new function – A path name is a URL address, and so it is a string of characters. Therefore, we tell Flash to expect a string type of data

Also put the load method and URLRequest inside that function:

```
function loadFile(pathToFile: string)  
{  
    theLoader.load(new URLRequest(pathToFile));  
}
```

Combine the steps into one complete Loader and URLRequest:

```
var theLoader = new Loader();  
load_area_mc.addChild(theLoader)  
  
function loadFile(pathToFile: String)  
{  
    theLoader.load(new URLRequest(pathToFile));  
}
```

And now you have a Loader that will load ANY type of file – when you need it call the function name that you gave your new function and tell it the parameter (the path to the file) it needs to work:

```
loadFile("images/painting_1.jpg")
```