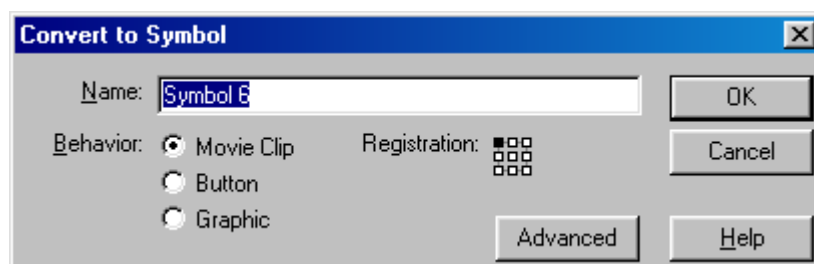


## Flash MX Tutorial on building a Dynamic Loading Slide Show with music – Images load from an external folder and fade In and Out. By R. Berdan update Nov. 29, 2002

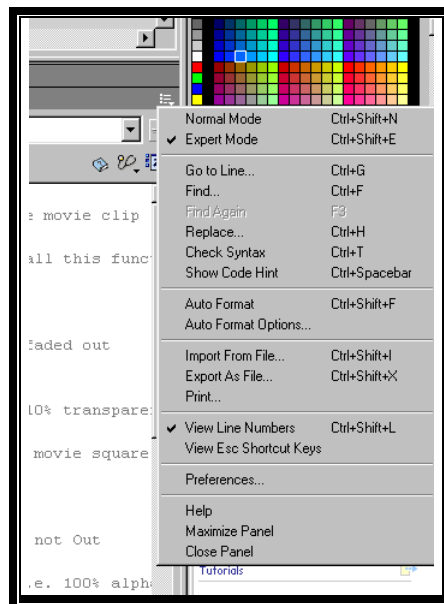
This tutorial involves creating a movie in which you will dynamically load images into a movie clip. The images will be stored in a folder called **images** and each image will be numbered sequentially: **image1.jpg, image2.jpg, image3.jpg etc..** The number of images you can include is unlimited, only one image will be loaded at a time from the server which makes the movie very memory efficient. You will also add background music that loads dynamically. Before you begin you will need a series of images numbered sequentially: image1.jpg, image2.jpg photos all the same size and you will need to download an mp3 music track ( Spanish\_guitar.mp3) feel free to find your own music track.

1. Create a folder and give it a name e.g. slide show.
2. Create another folder inside this folder and call it **images**.
3. Place several in the folder called images each image should be named sequentially image1.jpg, image2.jpg, image3.jpg etc. You can use .gif or .png files in which case change the file extension and action script accordingly. I have supplied 10 images.
4. Open Flash MX and create a new movie.
5. Call the first layer background – double click on the layer and type it the text “background”.
6. Create 4 more layers naming them: movie clip, buttons, text and actions.
7. If you wish to change the background color of your slide show select the background layer, select Modify menu>Document>background color - choose a color (you can do this at anytime).
8. Select the 1<sup>st</sup> Keyframe in the movie clip layer, then select the box drawing tool and draw a box the size of the window you wish to display your images in. The box can be any color or even just have a stroke color. The box should be slightly larger than your images. When you finished drawing the box – click and drag around it, then select Insert>Convert to Symbol>Movie clip. In the pop up box select the **Registration** and choose the square in the upper left corner. If you leave Registration in the center pictures will load with their upper left top corner from the center of the movie. You do not have to name the movie clip here in the pop up box but you will have to give it an instance name in the properties box, for this tutorial call the movie clip **square**.



Note the registration box select the upper left square

9. Select the movie clip with your pointer and in the Properties box at the bottom of the program select the instance text field and type in a name: **square** all lower case. If the properties box is not present select Window>properties. The movie clip will be referred to in action script by its instance name "**square**".
10. Select the buttons layer, 1st keyframe, open the Common libraries Window>Common Libraries>buttons, double click on the folder named Playback and drag two instances of the button named "playback play". You will need to rotate one of the buttons so it points to the left – select one of the buttons, select the scale\transform tool directly under the pencil in the tools palette – and rotate one of the buttons 180 degrees. (Note you can select other buttons if you wish). Drag a third button onto the stage – this button will make the show play automatically – i.e. you should have: back, play continuously, and forward button under the movie clip.
11. Select the buttons and add instance names: select the left pointing button and in the properties box name it **back**, and the right pointing button **next** in the instance properties box and name the play button **ssOn (for slide show on continuously)**..
12. Select the text layer and type in a title above the movie clip e.g. Pictures from Western Canada by include Your Name (this part is not essential to the slide show).
13. Select the Actions layer, right click and select Actions – the actions panel should appear. We are going to use the action panel window in Expert Mode to write action script. In the top right hand corner of the action pane – click on the icon that looks like a bulleted list and select Expert Mode and show line numbers. Put your cursor in the action pane window.



Select Expert Mode from the pop up

14. First we will create and initialize two variables: type in the following code starting from line 1. Comments in the script are preceded with -- and do not affect the code they are only to describe what is happening. Objects and properties are connected using dot syntax, properties in Flash are often preceded with an underscore e.g. `_alpha`.

In the actions layer select the first keyframe, right click select actions and add the code below

```
square._alpha=0; // set movie clip called square to transparent i.e. alpha=0  
mypic=1; // mypic is a variable we will use to count the images in the show
```

```
_root.onEnterFrame = function() // _root main movie timeline, when movie starts call function  
{  
    if (square._alpha <10) // if square movie contains image with less than 10% transparency  
    {  
        loadMovie("images/image"+mypic+".jpg", "square") // load image1.jpg into movie square  
        fadeOut=false; // image is not fading out  
        fadeIn=true; // image fade In is set to true  
    }  
  
    if (square._alpha > 10 && fadeOut)  
    {  
        square._alpha -= 10; // if fading fade movie = movie – 10% alpha until faded out  
    }  
  
    if (square._alpha < 100 && fadeIn && !fadeOut) // if image is fading and not Out  
    {  
        square._alpha += 10; // increment fade in 10% until completely faded in i.e. 100% alpha  
    }  
    else // if image is not fading in or out i.e. has loaded then do not fade in the image  
    {  
        fadeIn=false;  
    }  
}  
}
```

15. Save your flash movie into the folder you named slideshow, the movie should be outside the images folder but inside the slideshow folder on your desktop. Select Control Test movie – if you did everything correctly the first image should fade in and stop. If not select the check box – to check for syntax at the top of the action script window and try to locate any errors.
16. Now we will add the actions to the next and back buttons so they load additional images. The total number of images in the folder provided is 10 – they can be any number if you use a different number of images you will need to change the code to reflect this. Below the code you have typed in so far enter the following code in the action script window. If the window is no longer open close the Control >Test Movie window select the actions layer, 1<sup>st</sup> keyframe and right click select actions. Type in the code below in the actions frame below the `_root.onEnterFrame = function()` – the code is outside of the brackets of the function.

Continued on next page

```

next.onPress = function()
{
    if (mypic < 10 && !fadeIn && !fadeOut)
    {
        fadeOut=true;
        mypic++;
    }
    if (mypic >= 10)
    {
        mypic=1;
    }
}

back.onPress = function()
{
    if (mypic > 1 && !fadeIn && !fadeOut)
    {
        fadeOut = true;
        mypic--;
    }
    if (mypic == 1 && !fadeIn && !fadeOut) // if image is on 1 go to the last image in series
    {
        fadeOut=true;
        mypic=10;
    }
}

```

### Explanation of the code above

next.onPress next is the name of the button, onPress is an event handler that triggers the script below. The script then tests if the image mypic is numbered less than 10 if true, select the next image and increment by one i.e. mypic++ + mypic + 1, since mypic = image#.jpg this goes from image1.jpg to image2.jpg. The back.onPress button does the reverse. We have added an additional bit of code in the next.onPress so that when the button reaches 10, the last picture it will jump back to one so the user does not have to go back one image at a time.

How would you add a button so that when you click on the back button it would go to the last picture in the series?

\*if you leave out the && !fadeIn && !fadeout) part the program will not work properly.

### Creating an Input Box that shows the slide number being displayed.

To create an input text field that dynamically updates to show the slide number when the buttons next or backward are pressed do the following.

1. Select the text or background layer, if you don't have one create a text layer then select the text tool "A"
2. From the properties box drop down menu select> dynamic text –give it instance name **message**

3. On the stage type the number 1 – it should be surrounded by a bounding box, you can put space in front and after the number if you want to expand the box. Make sure you select a text color other than white!
4. Add the additional `message.text=mypic`; inside the first if statement as shown below, this will put the image number in the message box when the slide show advances or retreats.

```

square._alpha=0; // set movie clip called square to transparent i.e. alpha=0
mypic=1; // mypic is a variable that refers to the images loaded into the movie clip

_root.onEnterFrame = function() // _root main movie timeline, when movie starts call function
{
    if (square._alpha <10) // if square movie contains image with less than 10% transparency
    {
        loadMovie("images/image"+mypic+".jpg", "square") // load image1.jpg into movie square
        fadeOut=false; // image is not fading out
        fadeIn=true; // image fade In is set to true
        message.text = mypic; // places mypic into textbox with instance name message
    }

    if (square._alpha > 10 && fadeOut)
    {
        square._alpha -= 10; // if fading fade movie = movie – 10% alpha until faded out
    }

    if (square._alpha < 100 && fadeIn && !fadeOut) // if image is fading and not Out
    {
        square._alpha += 10; // increment fade in 10% until completely faded in i.e. 100% alpha
    }
    else // if image is not fading in or out i.e. has loaded then do not fade in the image
    {
        fadeIn=false;
    }
}

```

At this point you should have a slide show that you can advance and reverse and the slides should work fine when the first and last slide is reached.

To add a button that play the slides automatically – we will have to use time function in Flash.

### **Making the Slides advance automatically - 3 different ways you can do this.**

In order to advance the slides at a set interval say every 5 seconds we need to track the time when the movie is playing. In order to this we will use the built in Flash timer command `getTimer()`. Note both Flash –Actionscript and Javascript record time in milliseconds thus 5 seconds = 5000 milliseconds.

## 1) 5 Second Slide Advance no button required to start put the code – Simple but no on off

```
square._alpha=0;
mypic=1;
// script causes slides to advance automatically every 5 seconds – no start button it just plays

_root.onEnterFrame = function()
{
    if (square._alpha < 10)
    {
        loadMovie("images/image" + mypic + ".jpg", "square")
        fadeOut=false;
        fadeIn=true;
        message.text=mypic;
    }

    if (square._alpha > 10 && fadeOut)
    {
        square._alpha -= 10;
    }

    if (square._alpha < 100 && fadeIn && !fadeOut)
    {
        square._alpha += 10;
    }
    else
    {
        fadeIn=false;
    }

    if (mypic>=10)
    {
        mypic=1
        startTimeSet=0;
    }

    if (getTimer() > 5000 && mypic < 10 && !fadeIn && !fadeOut && mypic==1)
    {
        fadeout=true;
        mypic=2;
    }

    if (getTimer() > 10000 && mypic < 10 && !fadeIn && !fadeOut && mypic==2)
    {
        fadeout=true;
        mypic=3;
    }

    if (getTimer() > 15000 && mypic < 10 && !fadeIn && !fadeOut && mypic==3)
    {
        fadeout=true;
        mypic=4;
    }
    // continue to add if statements for every image in the series mypic==4, 5, 6, 7 etc.
}
```

**2) Second method requires a button with instance name ssOn of course you can change the name of the button just change the code accordingly.**

Select the play button>right click>actions and add the following code. The code allows one button to start and stop the slide show. An alternative way is to use two buttons and make the on button disappear when you click and an off button appear – you will do this with the sound buttons.

```
on (release)
{
    if(_root.ssOn) // if play button is on turn it off, off=0
    {
        _root.ssOn=0;
    }
    else if(!_root.ssOn) // if play button is off turn it on, on=1
    {
        _root.ssOn=1;
        // trace("_root.ssOn" + _root.ssOn) remove comment symbols to see value
    }
}
```

The code above permits you to use the same button to turn the auto advance (play continuously button) on and off – button instance name = **ssOn**. The button is initialized as off or 0 at the start of the program. The script above checks when you press or release the button – if the button is OFF – turn it ON and if it was ON turn it OFF. Now add the following script to your existing program

```
square._alpha=0;
mypic=1;
_root.ssOn=0; // button instance name ssOn is set to 0 or off when the program starts
statTimeSet=0; // staticTime variable set to 0 when the movie starts.
```

```
_root.onEnterFrame = function()
{
    if (square._alpha > 10 && fadeOut)
    {
        square._alpha-=10;
    }
    if (square._alpha < 10)
    {
        loadMovie("images/image" + mypic + ".jpg", "square")
        fadeOut = false;
        fadeIn= true;
        message.text=mypic;
    }
    if (square._alpha < 100 && fadeIn && !fadeOut)
    {
        square._alpha += 10;
    }
    else
    {
        fadeIn = false;
    }
}
```

*continued on next page*

```

curTime = getTimer(); //create a variable curTime and set it to the current time object

if(!statTimeSet) // if stat Time variable is not set then set
{
    statTime=curTime; // set static time = Current time built in
    statTimeSet=1; // set statTimer to 1 or ON
}

if ((curTime >=statTime + 5000) && mypic <= 10 && !fadeIn && !fadeOut && _root.ssOn)
{
    trace("curTime" + curTime)
    trace ("statTime" + statTime)
    fadeOut=true;
    mypic++;
    if (mypic > 10) // do not use >= or it will skip the last slide
    {
        mypic=1;
    }
    statTimeSet=0; // make sure this line is outside the if statement above it
}
}

```

*end of \_root.onEnterFrame=function()*

getTimer() – is a function that returns the number of milliseconds that have elapsed since the movie started playing.

if (!statTimeSet) - initially we set a variable statTimeSet to 0 which means off, this tests if the variable has not been set i.e. it is off the set a new variable statTime=curTime - we need to do this since curTime will always record the current Time elapsed and we need to compare it to another time statTime + 5000 seconds – in otherwords, you are taking a snapshot of the time the script run “statTime” then comparing it with the current time + 5000 – if 5000 milliseconds have elapsed go to the next frame. You can see what is happening with respect to the time by using the trace command.

**If you run a trace command this is what you will see the first interation of the slide show**

```

trace(statTimeSet) = statTimeSet1
trace(curTime) = curTime5104
trace(statTime) = statTime98 (i.e. the time that will be compared to curTime)

```

I admit this script is a bit “unusual” and I have to give the credit to my associate Juan Rivera for coming up with it – but it works!

- 3) Here is another way to advance the slides a specified amount of time - this script was written by Glen Zandboer - and frankly it is much easier to understand and follow.

Create a button and give it an instance name **autoforward**

In the first frame along with the other scripts we have done so far add the following script

```
autoforward.onPress=function()
{
    autoAdvance();
    setInterval(autoAdvance, 5000) // calls function autoAdvance every 5000 milliseconds
}

function autoAdvance()
{
    if (mypic <= 10 && !fadeIn && !fadeOut)
    {
        fadeout=true;
        mypic++;
    }
    if (mypic > 10) // must not be >= or will stop at 10
    {
        mypic=1;
    }
}
```

the key element in this script is the **setInterval** action - this Action; calls a function or a method or an object at periodic intervals while a movie plays. You can use an interval function to also update variables from a database or update a time display.

### **Adding Dynamic Sound with a start stop buttons to your slide show**

1. Create two new layers in the main time line – label the bottom layer stop music and the other layer start music
2. Open up the button library – I recommend the oval buttons, select two different colored buttons of the same shape and size, drag them onto the movie: one on the stop layer and one on the start layer. Give the button on the top layer and instance name – **musicbutton** in the properties box and the other one **stopmusic**. Then drag the music buttons so that one is directly under the other.
3. You will need to acquire a music file, preferably and .mp3 sound file, and place it n the root of your folder with slideshow.fla file. This file will be loaded dynamically into the movie clip when the user presses the music button.
4. Select the first frame in the actions layer and add the following code below the other code on the page:

*Continued on next page*

```

backmusic = new Sound() // create a sound object called backmusic for background music
{
    musicbutton.onPress = function
    {
        backmusic.loadSound("Spanish_Guitar.mp3", true) // load your music file
        music.button._visible=false; // make the music play button disappear
        stopmusic._visible=true; // make the stop music button visible
    }

    stopmusic.onPress = function()
    {
        backmusic.loadSound("Spanish_Guitar.mp3", false) // unload music i.e. stop playing
        music.button._visible=true; // make the music play button visible
        stopmusic._visible=false; // make the music stop button invisible
    }
}

```

In summary – there are many variations one could add to the scripts above, the unique feature of this program is that you can dynamically load any number of slides and music files. This keeps the original .swf file very small. For this slide show to run you will need to keep the image files reasonable small (under 30 Kb) or the show will not run smoothly on slower modems. You also created a new Sound object which you turned on and off. You can extend the music by looping it in the background.

In Flash MX – there is also a template for a dynamic slide show New>From Template>photoslideshow be sure to check this template out for other ideas and scripting tips.

In your text - Chapter 17 has a section on loading external assets dynamically and also shows you how to create a dynamic slide show.

**For a challenge re-write a script so that you only need one button to turn the music on and off**