

published on 8th November 2005

Using the Tween Class

```
import mx.transitions.Tween;  
import mx.transitions.easing.  
var xScaleT:Tween = new Tween  
    "_rotation", Elastic.easeOut  
xScaleT.onMotionFinished = f  
    this.yoyo();  
};
```

BY JAMES LEE

Chief Instructor

“When you install Flash MX 2004 or Flash 8, you acquire powerful classes that allow you to create animation independent of using Motion or Shape Tweening methods. Some of them are the Tween class, Transition class and a few Easing classes. This article shows you how easy it is to use the Tween and Easing classes, even if you aren't comfortable using ActionScript.”



James Lee
*Adobe / Macromedia
Certified Instructor*

Utilising the Tween Class

The main part of this tutorial is the code explanation, so instead of having you go through creating the text, I will provide the full source code for the animation.

```
import mx.transitions.Tween;
import mx.transitions.easing.*;
var xScaleT:Tween = new Tween(flash8,
    "_rotation", Elastic.easeOut, 0, 360, 5, true);
xScaleT.onMotionFinished = function() {
    this.yoyo();
};
```

(Before I explain the code, you should be aware that flash8 in our code references the instance name of our movie clip on the stage)

Explanation

Let's go through the code:

```
import mx.transitions.Tween;
import mx.transitions.easing.*;
```

- 1) The first line imports the Tween class file into your Flash document. You can find the class file called **Tween.as** at **C:\Program Files\Macromedia\Flash 8\<language>\First Run\Classes\mx\transitions** assuming a default installation.

```
var xScaleT:Tween = new Tween(flash8,
    "_rotation", Elastic.easeOut, 0, 360, 5, true);
```

- 2) This line creates a new tween instance and store as the variable **xScaleT**. The Tween class constructor requires seven parameters, which are in order:
 - **object** - The instance of a movie clip in which the tween is applied to.
 - **property** - The property in which to tween. This parameter requires a string of one of the following properties: **_alpha**, **_xscale**, **_yscale**, **_x**, **_y**, **_rotation**, **_height**, **_width**
 - **ease class and method** - The type of tween and the method of ease applied to it. Flash comes bundled with 6 easing classes:
 - i) **Back** - Extends the animation once beyond the transition range at one or both ends to give the effect of being pulled back from beyond its range.
 - ii) **Bounce** - Adds a bouncing effect within the transition range at one or both ends.
 - iii) **Elastic** - Adds an elastic effect that falls outside the transition range at one or both ends.
 - iv) **Regular** - Adds slower movement at one or both ends. This feature lets you add a speeding up effect, a slowing down effect, or both.

- v) **Strong** - Adds slower movement at one or both ends. This effect is similar to the Regular easing class, but it's more pronounced.
- vi) **None** - Adds an equal movement from start to end without effects, slowing, or speeding up.

These six easing classes each have three methods to control the ease:

- a) **easeIn** - The ease is applied only at the start of the tween.
 - b) **easeOut** - The ease is applied only at the end of the tween.
 - c) **easeInOut** - The ease is applied at both the beginning and end of the tween.
- **begin** - The beginning value for the property in the tween.
 - **end** - The end value for the property in the tween.
 - **duration** - A number indicating the length of time of the tween motion. If omitted, the duration is set to infinity by default.
 - **useSeconds** - A Boolean value indicating to use seconds if true or frames if false in relation to the value specified in the duration parameter.

```
xScaleT.onMotionFinished = function() {  
    this.yoyo();  
};
```

- 3) This is **event handler** for the xScaleT tween instance will be invoked when the current tween finishes (reaches its end value). Using the **yoyo** method, it basically instructs the tween to flip the end and begin values creating an endless back and forth motion. As you can see, the Tween class is actually very easy to use. Please feel free to manipulate and create the desired effect!