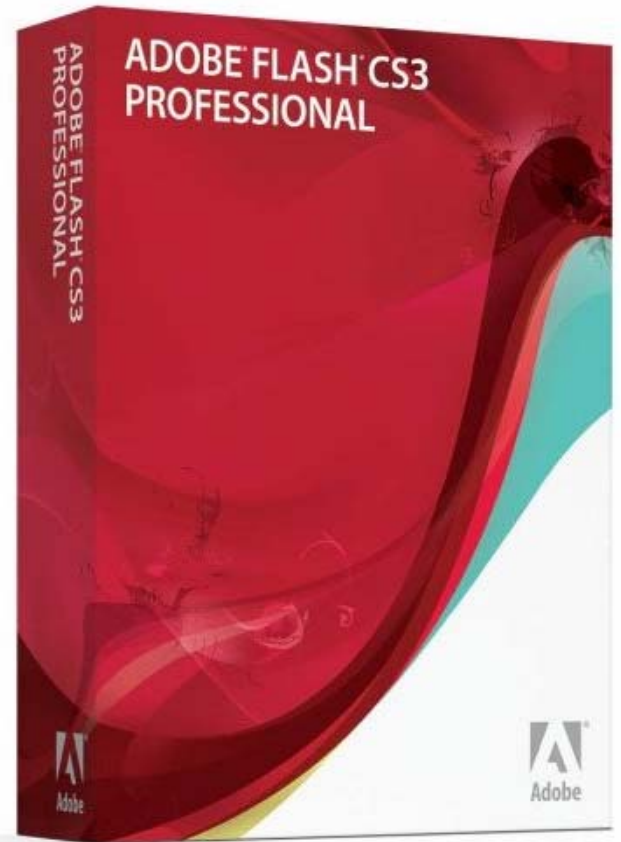


Animation Workflows In Flash CS3

Robert Penner

Flash Authoring Team

Adobe Systems



Main Topics

- Copy and Paste Motion
- Paste Motion Special
- Copy Motion as ActionScript 3.0
- Motion XML
- Copy Motion to Flex
- Advanced Topics (time permitting)

Ode to the Timeline



Love the Timeline Or Else



My History

- My first year in Flash 4 was all Timeline stuff
- Later, scripting took me beyond the Timeline's limitations
- Easing functions
- *mx.transitions.Tween*
- Adobe hired me to work on Flash Authoring features

Usefulness of Copy Motion as AS3

- Need to take designer's animation and make it scripted and dynamic
- The timeline matters!

Copy and Paste Motion

- Demo

Saving Motions as XML

- Store Motions as XML files and apply to various objects
- Check out the Commands Menu in Flash CS3
- Export Motion as XML
- Import Motion as XML
- Similar to saving “Styles” in LiveMotion

We Copy Core Transformations

- Position, Scale, Rotation, Skew
- x and y are the location of the transformation point, not the registration point
- Paste Motion applies these relative to the target object.
- E.g. if the *source* scales from 50% to 100%, a *target* at 40% will grow to 80%
- If you need absolute values, use Paste Motion Special.

We Copy Tweens

- Ease Percent
- Custom Ease (single or multiple curves)

We Copy Tweens

- Transformation Point

We Copy Color Transformations

- Color Mode: Alpha, Brightness, Tint, Advanced
- Blend Mode
- Filters

We Copy Motion Guides

- There are two modes for copying motion guides
- Magical Copy
- Practical Copy

We Copy Frame Properties

- Tween Scale
- Rotate Times
- Rotate Direction: None, Auto, CW, CCW
- Snap
- Runtime Bitmap Caching
- Frame Labels
- **For Graphic Symbols:**
 - Loop
 - First Frame
 - Tween Sync

Paste Motion Special

- Lets you pick and choose the properties you want to apply.
- The settings are saved each time.

We Work With Containers

- Movie Clips
- Graphic Symbols
- Button Symbols
- Text Fields (static, dynamic, and input)
- Groups
- Drawing Objects (which are basically groups)
- Rectangle and Oval Primitives
- You can transfer motion between object types.

Copy Motion Doesn't Support

- Multiple elements on a frame or layer
- Raw Shapes
- Shape Tweens
- If the Transformation Point changes location after the first frame, it is ignored.

We Change Containers, Not Contents

- It applies transformations to the container.
- It leaves vector or bitmap content alone.

Copy Motion as ActionScript 3.0

- Demo

Any DisplayObject Can Be Scripted

- MovieClip, Sprite
- Textfield

MotionXML

- Uses MXML syntax.
- The Flex compiler maps MotionXML to the AS3 classes automatically.
- More on Flex integration later.

The <Keyframe> tag in MotionXML



The `<source>` tag in MotionXML



The <transformationPoint> tag in MotionXML

- Problem: How do you transfer rotation from a wide object to a skinny one?
- When the dimensions change, where should the transformation point go?
- Solution: Use the bounding box and make the transformation point relative to that.
- Top-left corner is (0, 0). Bottom-right corner is (1, 1).

Motion AS3 Files

- Are located in Flash's Configuration folder under Program Files
- E.g., on Windows:
C:\Program Files\Adobe\Adobe Flash CS3
\en\Configuration\ActionScript 3.0\Classes\fl\motion

The *Animator* Class

- Controls all animation properties for a single object.
- Provides methods and events for scripting.

The *Motion* Class

- Is at the root of the data structure.
- The MotionXML is parsed into objects that match the structure

The *fl.motion* Package

Model-View-Controller-esque

- The MotionXML or Motion class is the Model.
- The DisplayObject is the View.
- The Animator class is the Controller.
- The Motion class doesn't have a "current time".
- Animator queries Motion for data at each frame.
- Animator creates a new Matrix, ColorTransform, filters array, etc. and updates the DisplayObject.

Animator has playback methods

- We matched Flex's *Effect* class APIs where possible.
- *play()* begins at first frame
- *stop()* returns to first frame
- *end()* goes to last frame
- *pause()*
- *resume()*
- *rewind()* preserves playing/stopped status

Animator Has Playback Properties

- *time* is getter/setter; seeks to a specific frame.
- *isPlaying* is read-only
- *repeatCount* sets the number of loops, same as Flex.
- *autoRewind* is useful for eliminating “drift” [example]

Knowing When a Motion Ends



Animator Broadcasts These Events

- *MotionEvent.MOTION_START*
- *MotionEvent.MOTION_END* fires after loops are completed
- *MotionEvent.TIME_CHANGE* fires before the screen is updated
- *MotionEvent.MOTION_UPDATE* fires after the screen is updated

MotionXML in Flex

- The *Copy Motion as XML* Command was requested by the Flex Team
- Demos

Transitions are Ported to AS3

- *AS2 mx.transitions => AS3 fl.transitions*
- The *mx.transitions* were originally for Screens in MX 2004
- These pre-date Flex Effects, and so are not compatible
- Screens are not supported in AS3
- But the Transitions are popular with non-Screens, so we did a straight-forward port to AS3

The Easing Classes

- The `fl.motion.easing` package has all of the functions, under their original mathematical names (e.g. `Quad`, `Cubic`)
- The `fl.transitions.easing` package has the subset from AS2 `mx.transitions.easing`, with friendly names (e.g. `Regular`, `Strong`)

The *FunctionEase* Class



The *CustomEase* Class



The *BezierEase* Class



The *fl.motion.Color* Class

- Inherits from `flash.geom.ColorTransform`.
- Will work wherever `ColorTransform` is accepted, e.g. `mc.transform.colorTransform`.
- Adds properties for brightness and tint.
- Has methods for interpolating between two color values, or two `ColorTransforms`.

The *MatrixTransformer* Class

- Is a collection of static methods that manipulate a Matrix instance.
- Gives you more control than the `flash.geom.Matrix` class.
- `MatrixTransformer` lets you rotate a Matrix *to* a specific angle, whereas `Matrix.rotate()` changes the rotation *by* an angle.
- `MatrixTransformer` gives you the current scale, rotation and skew values in a Matrix.
- It lets you change one property without affecting the others.
- `Matrix.scale()` can change a rotation into a skew (undesirable).

AS3 Template for Copy Motion as AS3



Animator.orientToPath

- Auto-rotates the object in the direction of motion.
- The auto-rotation is relative to the object's angle in the first frame.
- *Animator.orientToPath* is separate from *Keyframe.orientToPath*.
- *Keyframe.orientToPath* only works with motion guides, but *Animator's* property will work with any change in position.

Animator.positionMatrix

- This is an advanced feature suggested by the Flex Team.
- The *positionMatrix* can scale and rotate the motion path as a whole.
- It doesn't affect the object's own scale or rotation.
- Useful for pretty rotation effects.
- Useful for mapping a motion path onto different start and end points.
- One approach: use a diagonal motion and scale it into place.
- It's easier if you use a Flash 4 line-drawing hack: 100x100 diagonal.
- Drawback: squishing too much in one direction can lose detail.
- Another approach: use a horizontal motion and rotate and scale into place.

The ITween Interface



Have Fun!

- <http://robertpenner.com/presentations>