

# **KoolMoves**

**version 5.6**

**User Manual**



# KoolMoves User Manual

Lucky Monkey Designs, LLC  
South Carolina, USA  
[www.koolmoves.com](http://www.koolmoves.com)

The information in this document is subject to change without notice and should not be construed as a commitment by Lucky Monkey Design. While every effort has been made to insure the accuracy of the information contained herein, Lucky Monkey Design assumes no responsibility for errors or omissions. Lucky Monkey Design assumes no liability for damages resulting from use of the information contained in this document.

Flash™ is a registered trademark of Adobe, Inc. in the United States and other Countries.

The software described in this document is furnished under license and may be used or copied only in accordance with the terms of that license.

This publication may be reproduced, photocopied, stored on a retrieval system, or transferred without the expressed prior written consent of Lucky Monkey Design.

## PRINTING HISTORY

August 2005	Version 0.2
February 2006	Version 0.3
February 2006	Version 0.3i
April 2006	Version 0.4
October 2006	Version 0.5

## Table of Contents

<b>Introduction.....</b>	<b>vii</b>
<b>Part I – KoolMoves Environment.....</b>	<b>9</b>
<b>Chapter 1 – KoolMoves User Interface.....</b>	<b>11</b>
Screen Components.....	11
User Interface Style Settings.....	12
Configuration Wizard.....	13
Screen Component Descriptions.....	13
Tools Toolbar (aka Toolbox).....	20
<b>Chapter 2 - Preferences and Options.....</b>	<b>23</b>
Movie Menu.....	23
File > Preferences.....	25
Options Menu.....	29
File > Export Settings.....	32
<b>Chapter 3 – Help Features &amp; Tutorials.....</b>	<b>35</b>
Help Menu.....	35
Wizard Interface Style (aka Wizard Mode).....	39
Help in Other Languages.....	49
<b>Part II – Menu Toolbar &amp; Menu Commands.....</b>	<b>51</b>
<b>Chapter 4 – File Operations.....</b>	<b>53</b>
File > New (Ctrl-N).....	53
File > Open (Ctrl-O).....	53
File > Save (Ctrl-S).....	53
File > Save As.....	53
File > Export Menu / Export Icon.....	54
File > Insert KoolMoves Movie.....	60
Working with Website Editing Tools.....	61
MIDI Sound Files.....	61
<b>Chapter 5 – Data Views.....</b>	<b>63</b>
1) Properties Data View (Ctrl-F3).....	64
2) Score/Timeline Data View (Shift-F3).....	83
3) Movie Overview (Alt-F3).....	95
4) Actions and Sounds Data View (Ctrl-E).....	95
5) List of Shapes (F9).....	96
6) List of Frames (Ctrl-K).....	97
Other Views.....	97
<b>Part III – Tools Toolbar (Toolbox) &amp; Menu Commands.....</b>	<b>99</b>
<b>Chapter 6 – Drawing/Editing Shapes.....</b>	<b>101</b>
Selecting Shapes and Points.....	101
Shape Menu Commands.....	102
Points Menu Commands.....	104
Edit Menu Commands.....	106
Miscellaneous Commands.....	106
Drawing Tools.....	108
Drawing Lines.....	110
Reshaping.....	110
Bones.....	114
Question to Flashpulse – Comics (flashkit thread 663649).....	118
<b>Chapter 7 – Buttons.....</b>	<b>119</b>
Creating Buttons.....	121
Working with Buttons.....	122

<b>Chapter 8 – Toolbox Libraries.....</b>	<b>129</b>
Web Interfaces.....	129
Text Effects Templates.....	129
Clip Art.....	131
Animated Clip Art.....	132
Button Gallery.....	133
<b>Chapter 9 – Importing Images.....</b>	<b>135</b>
Static Images (BMPs, GIFs, JPGs, PCXs, PNGs, and TIFs).....	135
Vector Graphics.....	135
SWF Flash Movie.....	136
Flash Live Video.....	136
Animated GIFs (Work Around).....	137
<b>Chapter 10 - Text.....</b>	<b>139</b>
Static Text.....	139
Dynamic Text.....	142
<b>Chapter 11 - Movie Clips.....</b>	<b>147</b>
Creating Movie Clips.....	147
Movie Clip Timeline.....	148
Movie Clip Properties Data View.....	148
<b>Chapter 12 – Transforms &amp; Effects.....</b>	<b>155</b>
Toolbox Transforms.....	155
Transforms Menu (other commands).....	156
Toolbox Effects.....	159
Creating Your Own Transition.....	159
<b>Part IV – Advanced/Additional Features.....</b>	<b>161</b>
<b>Chapter 13 - Action Scripting.....</b>	<b>163</b>
Opening the Action Script Editor (ASE)for the first time.....	163
The Action Script Editor (ASE).....	164
Editing ActionScript code.....	167
Unimplemented ActionScript.....	169
<b>Chapter 14 – KoolMoves Components.....</b>	<b>171</b>
Application Programming Interfaces (APIs).....	171
Interface Components in the Toolbox.....	172
Interface Components not in the ToolBox.....	195
Non-Interface Components.....	197
Using the Symbol Library to add Components.....	203
<b>Chapter 15 – Symbol and Link Library.....</b>	<b>205</b>
Symbol Library.....	205
Link Library.....	207
<b>Appendices.....</b>	<b>209</b>
<b>Appendix A – Known Issues and Error Messages.....</b>	<b>211</b>
Exceeded SWF Buffer Size.....	211
SWF Too Big.....	211
Animation Not Displaying or Warning Messages Display .....	212
Internet Explorer Security Issues .....	212
<b>Appendix B – KoolMoves Reference Tables.....</b>	<b>213</b>
Menu Structure.....	213
Shortcut Keys.....	216
Slide Show Transition ID Table.....	217

<b>Appendix C – Programming / Coding Advice et.al.....</b>	<b>221</b>
ActionScripting.....	221
Accessing Databases.....	222
Adding Flash Movies to PowerPoint.....	222
Tips on Adding Flash Movies to Web Pages.....	222
Backend Code – Sources and Samples.....	224
HTML Coding.....	226
<b>Appendix D – Useful 3rd Party Software.....</b>	<b>227</b>
Audio Editors.....	227
Converting SWF to AVI.....	227
Flash Toolset .....	227
Vector Graphics Applications.....	227
Exe / Projector Creation.....	228
<b>Appendix E – Toolbar Library / Effect Listings.....</b>	<b>229</b>
Text Effect Templates.....	229
Web Interfaces.....	229
Clip Art Gallery Categories (Folders).....	230
Animated Clip Art.....	230
Button Gallery Categories (Folders).....	230
Motion Scripts.....	233
3D Effects.....	233
Filter Effects.....	234
Blend Mode Effects.....	234
<b>Appendix F – Frequently Asked Questions (FAQ).....</b>	<b>235</b>
<b>Index.....</b>	<b>239</b>

# KoolMoves User Manual

## Introduction

### About Flash

Flash™ was developed by Macromedia (since bought by Adobe) to make it possible to have high impact web sites, alive with animations, sounds, interactive interfaces, and powerful graphics. Flash utilizes a very compact vector graphic format, so web sites with Flash animation download quickly. Flash also has a huge market penetration, with most web browsers containing the Flash player plug-in. For these reasons, most high-end web sites use Flash animation. However, the Adobe Flash package is expensive and has a steep learning curve.

### Why KoolMoves

Unlike Adobe Flash, quality animation is easy and affordable with KoolMoves. It is ideal for creating animated web pages, banners, navigation systems, and intro pages. It features libraries of text effects templates, web page templates, vector clip art and buttons, ActionScript text effects as well as many transition effects. For advanced users, it even has Flash MX ActionScripting, dynamic text fields, and character animation with bones.

KoolMoves takes an approach to Flash™ movie creation that is in tune with the animation process, because it was designed by a professional animator. It is a mature animation package with ease of use and power as guiding principles. To make animations in KoolMoves all you have to do is draw or import shapes or images and then reposition, reshape, and recolor the animation objects in adjacent key frames to achieve the desired look. KoolMoves automatically morphs shapes in between each of the shapes you create to achieve smooth animated motion. It's like putting together a cartoon strip. Each frame contains text, images, shapes, sounds, and actions. For character animation, one frame would have the characters in one pose and in the next frame in another pose.

KoolMoves also has “Wizards” to help users new to computers and computer animation. Ranging from automated guides that assist you in performing fundamental tasks for the first time, to shortcut menus that execute simple processes like creating buttons or editing shape properties, the Wizard tools allow you to express creativity in the quickest fashion with the least amount of frustration.

Beyond simple animation, KoolMoves provides Flash MX ActionScripting capability. A large fraction of the Adobe® Flash 5, 6, 7 and 8 operators and commands are supported at this time. This functionality is regarded as advanced and requires knowledge of ActionScripting that can be obtained from books or web sites devoted to that subject. This manual is not intended to teach ActionScript and where ActionScripting is addressed it is assumed that you have scripting experience.

## About This Manual

This version of the manual has been composed and typeset in Windows™ Word®, and exported to the Adobe™ Portable Data File (PDF®) format. The inconsistent use of Capitalization with topic names is intentional and is done to control the topic incidence in the manual's index.

The organization of the manual is based loosely on the layout of KoolMoves Graphical User Interface (GUI), as described in Chapter 1. Part I of the manual focuses on the overall appearance of the interface and the Help function, Part II on the Tools Toolbar and related visual element functionality, and Part III on the Menu Toolbar and miscellaneous menu actions not covered elsewhere. Part IV has brief introductions to advanced features while the Appendices contains reference information useful for troubleshooting, using KoolMoves' features and expanding the user's capabilities.

## User Guides

Effective with manual version 0.3b, certain portions of the manual featuring advanced features have been broken out into independent User Guides. The user guides are located with the manual on the KoolMoves support page. It is believed that this will help users print only those materials that they need. Bookmarks and hyperlinks have been added with 0.3m to facilitate navigating these materials and accessing context relevant information.

## Administrivia

All questions concerning installation, purchase, and / or registration of KoolMoves that are not addressed below can be emailed to [support@koolmoves.com](mailto:support@koolmoves.com).

## Installation

Installation of KoolMoves is simple. Double click on kmsetup.exe. The installation program will step you through the process of installing the software. For most screens, just keep pressing Next.

To uninstall, use the Control Panel Add/Remove Programs to remove the program.

## Registering KoolMoves

Upon purchase of a registration code, you will receive a download location to a full version of KoolMoves without restrictions and a full set of library items.

## Upgrades

Those who purchased KoolMoves after March 2005 will receive free upgrades through January 2007.

## **Part I – KoolMoves Environment**

**Chapter 1 – KoolMoves User Interface**

**Chapter 2 – Preferences and Options**

**Chapter 3 - Help Features & Tutorials**

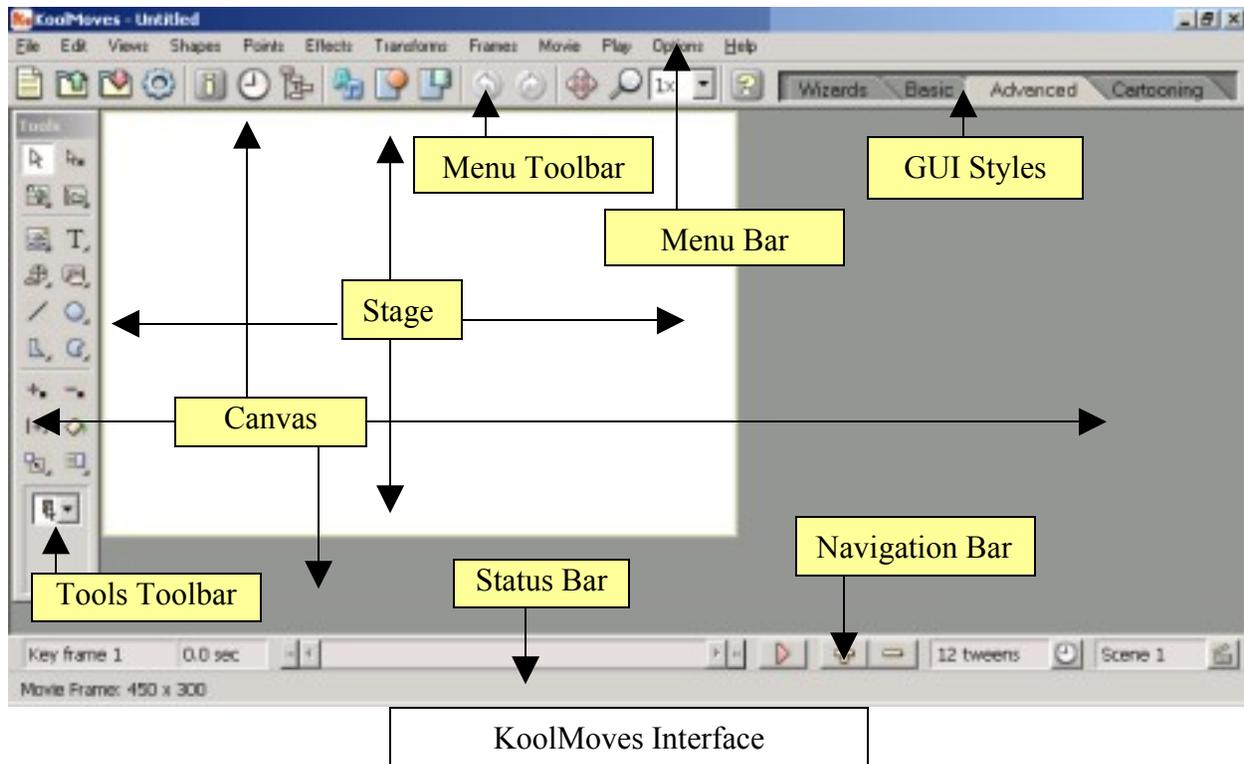
# KoolMoves User Manual

## Chapter 1 – KoolMoves User Interface

This chapter covers the KoolMoves user interface. Following the principles of simplicity, usefulness and flexibility, the screen layout is highly customizable, incorporates many shortcuts, and continuously displays information concerning the current action. Many of the tools briefly addressed in this chapter will be covered in depth later in the manual. But first we will look at a couple tools that will help customize the interface to your unique requirements.

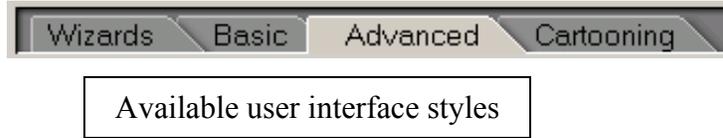
### Screen Components

The appearance of the KoolMoves environment depends on the graphical user interface (GUI) style selected and your preferences. For example, below the Advanced user interface has been selected and none of the optional data views have been opened. Screen components common to all configurations have been labeled.



## User Interface Style Settings

KoolMoves has four user interface styles available. You can select the appropriate interface on the Screen and also in File > Preferences > View. KoolMoves will determine the appropriate Tools Toolbar based on the user interface style selected.

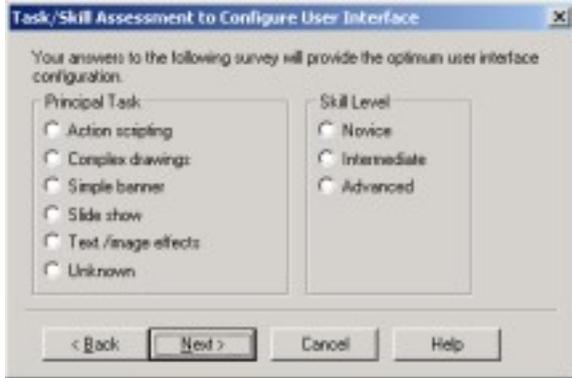


The following table identifies the mapping between functionality and the 4 interface styles.

	Wizards	Basic	Advanced	Cartooning
Banner wizard	x			
Slide show wizard	x			
Effects	x	x	x	x
SWF Flash Movie import		x	x	x
Full set of drawing tools			x	x
Shape point manipulation			x	x
Dynamic text			x	x
Masking			x	x
ActionScripting			x	x
Interface components			x	x
Flash Live Video import			x	x
Non-linear morphing				x
Bones				x

User Interface Features

## Configuration Wizard



File > Configure launches the Configuration Wizard. After answering a couple questions on your major use for KoolMoves and your level of experience with graphics software, click on Next and the wizard will recommend a set of interface settings.

You can change your preferences before clicking on OK, and change the settings later. If you have any questions about this or any other window, check the index at the end of this manual to find entries on the topic.

## Screen Component Descriptions

### Work Area: Canvas and Stage

Canvas - Shapes, text and other visual elements are placed or created here. Also located on the Canvas with the visual elements is the Stage. To navigate over the drawing area, particularly in the case of zooming, use the Pan Movie tool.

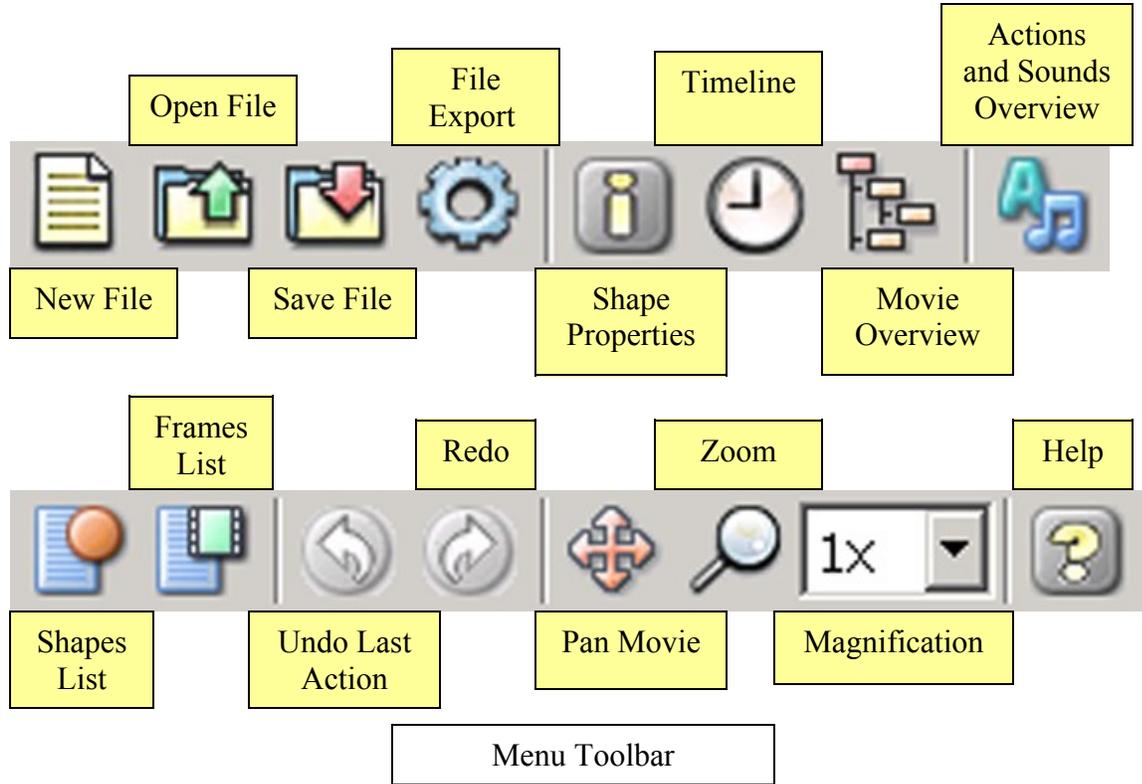
The Toolbox and other KoolMoves interactive elements (dialog boxes, data views, and preview windows) float over the Canvas and can be repositioned as desired with a simple left-click and drag.

Stage - The white section of the Canvas. The Stage represents the runtime display area / the physical dimensions of the movie. Each frame can start with a copy of the stage from a previous Frame, or start with a blank Stage like a fresh canvas.

Bounding Boxes – Used to display the height and width of a visual element, bounding boxes have color coded corners to indicate the type of object being displayed (blue for buttons, green for movie clips, and pink for imported swfs and flvs). Green dashes indicate effects have been applied to the object.

## Menu Toolbar

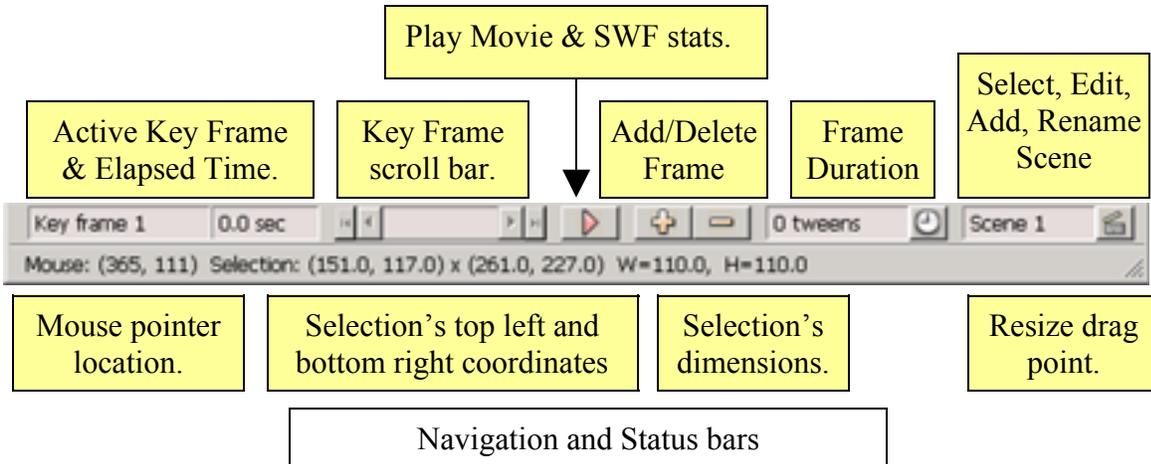
Located below the menu bar is a collection of icons allowing quick access to frequently used menu commands. These features are covered in detail in Part II of the manual.



## Navigation and Status Bars

Navigation Bar – Located near the bottom of the screen, these tools allow you to navigate, create and delete Key Frames and Scenes. You can also use the PageUp and PageDown keys to move to previous and next Frames, respectively. Frames and Scenes are discussed further below.

Status Bar – This context sensitive display has information on the current tool’s position and a selected shape’s physical dimensions and location on the Stage. The location and dimension units are pixels/pts.



Flash uses a Cartesian coordinate system. The origin point (0,0) on the stage is located at the upper left hand corner. Within movie clips and button states, the origin point is located at the center of the editing area. For an excellent description of Cartesian coordinates in Flash, please see the trigonometry tutorial at [actionscript.org](http://actionscript.org).

### Play Movie



KoolMoves has an internal player which plays the movie in your web browser as a Flash™ movie on a blank web page or in a stand-alone Flash™ player if you have one (download the browser plug-in from Adobe). You can also single-step through the movie. All of the following commands can also be found in the Play menu.



#### *Play in Web Browser*

View your project in your default web browser. This command is also available when editing movie clips.

#### *Play Portion of Movie*

Select Key Frames (and Scenes if there is more than one) to view in your default web browser.

### ***Play Stand-Alone Player***

View your project with an internal player and a Flash™ player if you have one. If you purchased Adobe Flash™, you undoubtedly have a Flash™ player in your program files\Adobe folder. It is named SAFlashPlayer.exe.

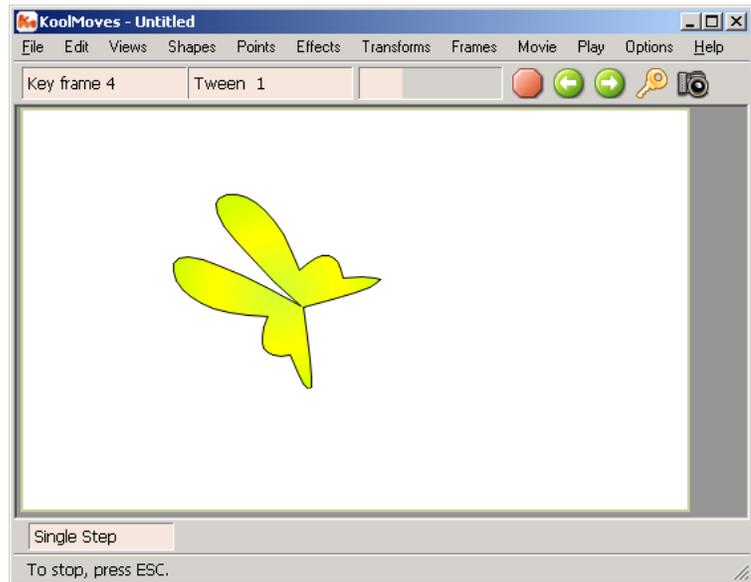
### ***Set Play Directory...***

You can assign the directory from which movies will play. This command is also available when editing movie clips. This information is preserved between sessions.

### ***Play Single Step***

Single step play mode allows you to view morphing changes between each tween and key frame. This analysis can help you to decide if you need more tweens or can do with fewer tweens. To exit single step play, press Esc or the Stop icon.

To move back or advance through the movie by one tween frame, click on the arrow icons. To convert the current tween frame to a key frame, click on the key icon. To open the [Capture Movie Frame](#) dialog, click on the camera icon.



### ***SWF File Statistics***

See [SWF File Statistics](#) in Chapter 2.

## **Frames**

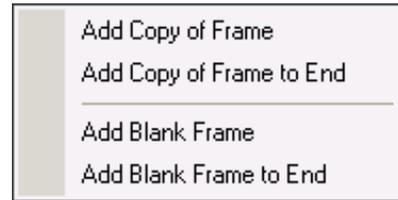
### ***Key Frame vs Tween Frame***

A key frame is a frame that you create yourself. When cartoons are drawn by hand, a master animator typically creates the key frames and an apprentice creates the tweens. In KoolMoves, the software generates the tweens based on the positions and colors of shapes in the two key frames. Tweening occurs when KoolMoves automatically makes one or more frames that go in between two key frames.

**Editing and Playing Frames**



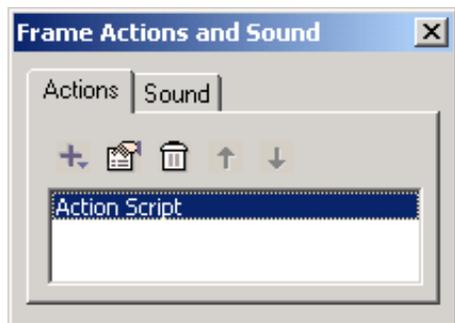
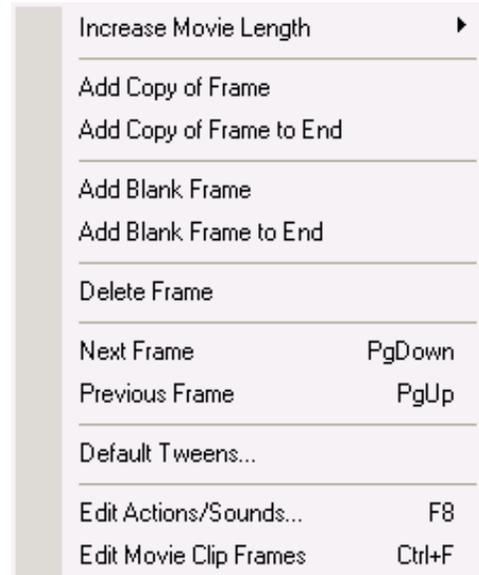
Inserting a copy of the current frame at the end of the movie or after the current frame is the principle method for growing the movie. The copy provides the starting point for the next stage of the animation: changing positions of shapes, moving shape points, changing colors, adding new shapes, etc. When a frame copy is made, the shapes in the copied frame retain a relationship to the shapes in the original frame, which allows morphing to occur.



Additional Commands are available in the Frames menu. These commands consist of adding, deleting, or navigating frames.

The Default Tweens... command changes the default tween value applied to new frames.

The Edit Actions/Sounds... command opens the Frame Actions and Sounds dialog. Use this dialog to add ActionScript coding and sound files to a frame (see Chapter 5, [Score/Timeline Data View](#), for more details).



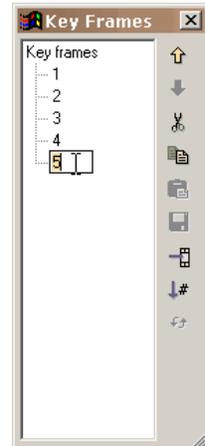
The Edit Movie Clip Frames command opens the selected Movie Clip for editing (see Chapter 11, [Movie Clips](#), for more details).

You can navigate between key frames by moving the frame slider or pressing first frame, previous frame, next frame, or last frame buttons on either side of the slider. PageUp and PageDown will move to the previous and next frames, respectively. The name of the current key frame is shown on the left, along with the length of the movie when that frame is loaded.



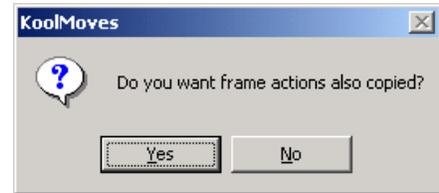


You can display a list of all the key frames by clicking the icon in the menu bar or using Views > List of Key Frames. This list can be used for navigating, renaming key frames, and frame editing (cut, copy, paste, and reversing the order of frames). To change the position of one or more frames in the movie or to cut/copy/paste frames, make your frame selections in the list of frames and use the editing commands on the right.

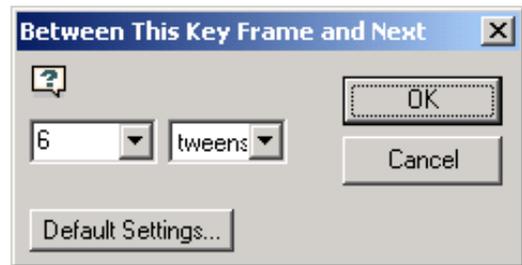


Shift/Ctrl select to select multiple frames. Paste places the cut or copied frames before the currently selected frame. To rename (label) a frame, select the item in the list and click again to get a cursor. Type in the name you want. The window can be resized by dragging the resizing marks in the lower right.

KoolMoves also allows you to copy the Frame Actions associated with a frame when the frame is copied.

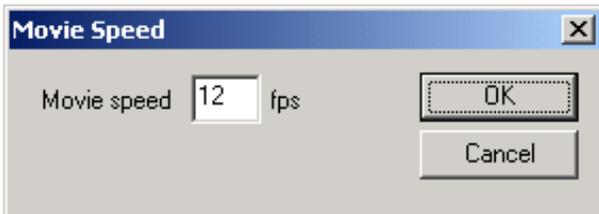


The value next to the clock indicates the Frame Duration – the seconds or tween frames between the current frame and the next. You generally want this number to be greater than zero so KoolMoves will generate smooth transitions (called morphing) between the key frames. Click on the clock to change the frame duration. To speed up action between two key frames, assign fewer tweens (dramatic effects can be achieved with 0 tweens!). To slow down action or to make the animation smoother, assign more tweens.



Key frames can have different frame durations. The default value is applied when new key frames are added. The frame Duration can be changed simultaneously for all key frames or for selected key frames through the List of Frames window.

**Frame Rate**



Video and television typically run at 30 fps (frames per second) and film traditionally runs at 24 fps. Most cartoons and anime are done at 12 or 15 fps to save money whereas most high quality animations like Disney feature films use 24 or 30 fps. Faster frame rates produce smoother animation but require more tween

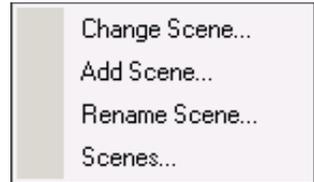
frames to achieve the same length of movie. Frames per second is set in Movie > Movie Speed.

**Scenes**

Frames can be grouped into Scenes, allowing Flash movies to be organized like, for lack of a better word, movies. Scenes are most useful when breaking up large movies into manageable pieces, or when a project has been allocated across different animators. The current scene is displayed in the Navigation bar.



 Click on the Clapboard button to open a popup menu: Click on Scenes to open the List of Scenes window. To view all scenes defined in the current project open the Movie Overview or the List of Scenes window.



 Add new scene



 Move selected scene.

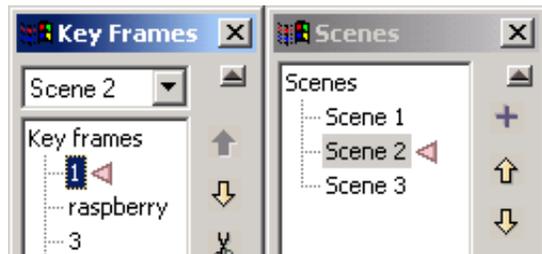
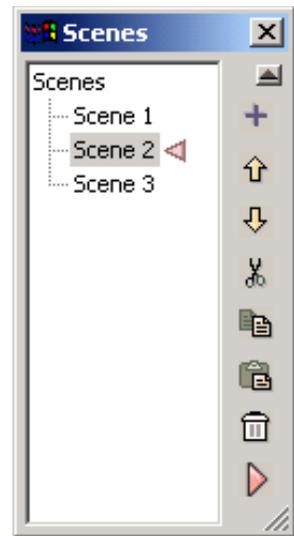
 Cut selected scene.

 Copy selected scene.

 Paste selected scene.

 Delete selected scene.

 Play selected scene in web browser.



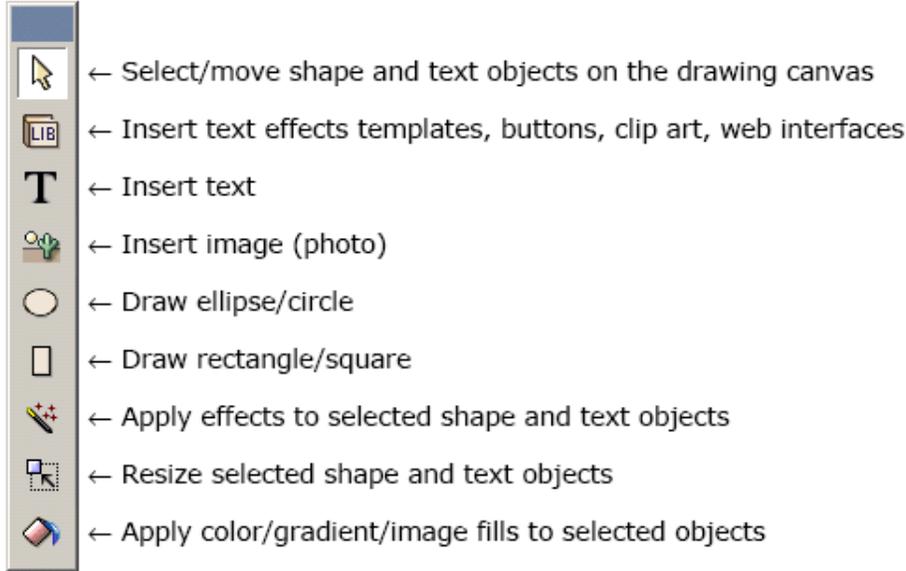
The List of Frames Data View will automatically create a list box of available Scenes when a movie has more than one scene.

Under the Play menu the Play Portion of Feature can also use Scenes to preview a subset of the movie.

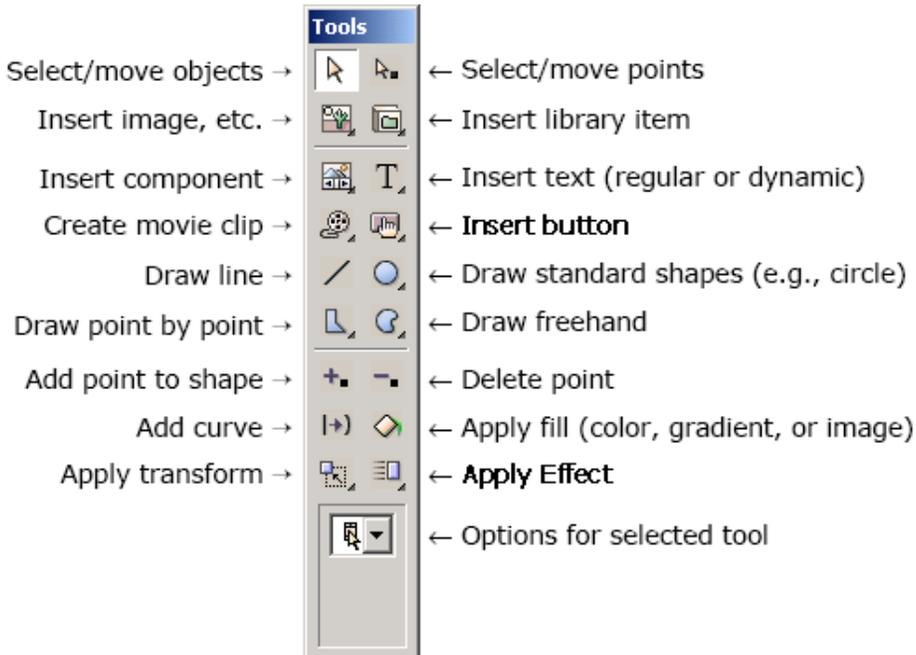


### **Tools Toolbar (aka Toolbox)**

A collection of icons for frequently used object creation and editing tools; toggle toolbox display with Views > Tools. The full set of tools is available in the Advanced and Cartooning interface, while the Basic interface has a reduced set of tools. No tools are available in the Wizard user interface. The tools are covered in Part III of the manual.



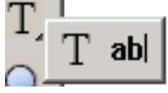
Basic Interface Tools Toolbar



Advanced and Cartooning Interface Toolbar



Clicking on a tool will select it and change the mouse cursor to reflect the current activity. Many of the tools in the Advanced / Cartooning level toolbar actually have a list of choices – you will see a black triangle in the bottom right corner.



Click and hold on the tool, or click on the black triangle, to open the tool drawer and view the choices.



The tool icon will change to reflect the selected / active tool. Even when another tool is being worked with, the last choice for a given tool is still reflected in the toolbar. The toolbar will reset to the default set of tools the next time you launch KoolMoves.

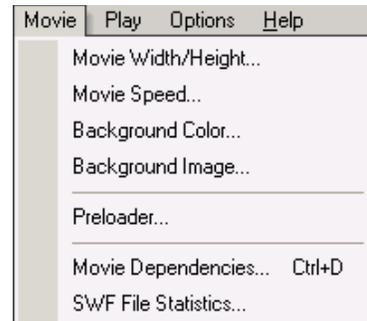


## Chapter 2 - Preferences and Options

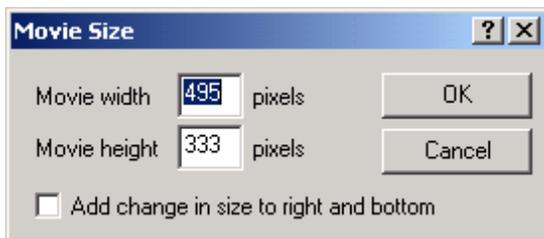
Within KoolMoves you can customize the work environment as well as alter certain settings that affect the appearance and performance of the movie both in the author-time environment as well as in the Flash™ Player. Parameters that exclusively pertain to the swf output file are covered in the [Export Setting](#) section.

### Movie Menu

The following are submenu items under Movie. For Example, Movie > Movie Width/Height.



### Movie Width/Height



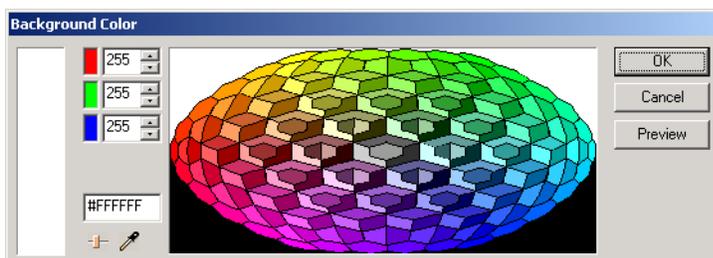
The size of the Stage is set here. In exporting to a HTML file this size should be entered with the .swf movie in the appropriate place. The maximum width and height values are 2880.

### Movie Speed

The speed of the movie is set here. The actual speed of the movie may prove to be less than your desired speed because of performance limitations; for example, movies with a lot of transparency will tend to play slower. A value in the range of 12 to 60 FPS (frames per second) is reasonable. The actual FPS rate depends on the capabilities of the computer playing the movie. For example, FPS in the range of 18-24 on old machines might be 24-30 on new machines and 40-50 on brand new machines. Older versions of the Flash player become unstable around 35 FPS.

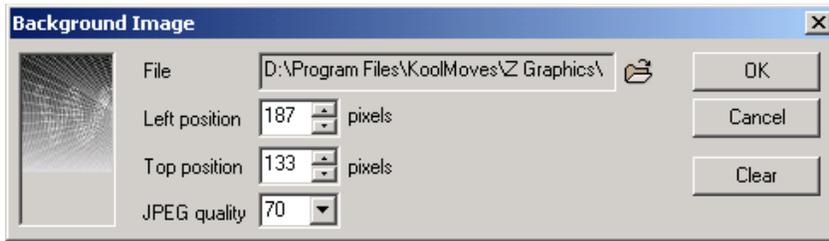


### Background Color



Use the Background Color picker to set the background color. See the [Coloring / Transparency](#) entry in Chapter 5 for details in setting solid colors.

## Background Image



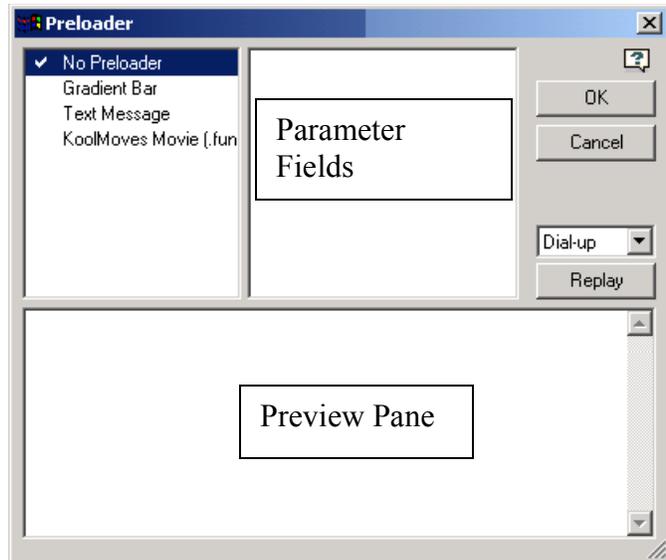
Click on the folder icon to browse to the image file. KoolMoves will center the image by default. Supported file formats are GIF, JPEG, and PNG.

## Preloader

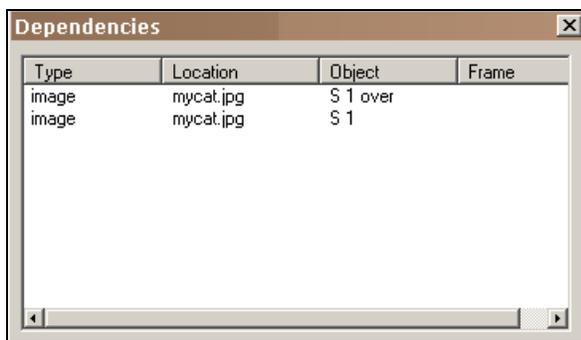
Preloaders give viewers something to watch while they wait for the animation to start. Use the Preload Movie interface and modify the default message or gradient bar settings, or select a KoolMoves fun file to run.

The interface has a preview window; select a connection type and preview the Preloader at different download speeds.

KoolMoves adds an additional 2 frames to the beginning of the swf file on export – name your important key frames to work with action scripting commands.



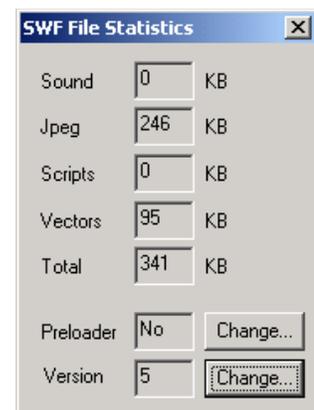
## Movie Dependencies



Dependencies displays a list of all files used or referenced by the movie -- images, sounds, imported swfs, and go to URLs. This is useful for spell checking file paths and identifying files that need to be included with the movie.

## SWF File Statistics.

The File Statistics window summarizes the dynamic factors that affect movie performance. As a diagnostic tool, this screen allows you to quickly identify performance killing movie elements. Additionally, it has shortcut buttons to open the Preloader and Version dialog windows.



## File > Preferences

The following are items in the File > Preferences dialog window.

### View Preferences:

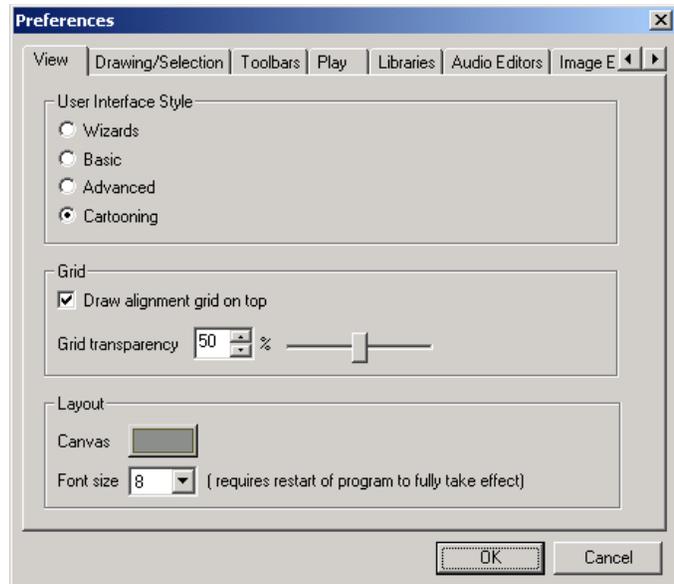
**User Interface Style.** This allows you to select the interface you feel most comfortable with. Advanced skills include masking, shape point manipulation, and ActionScripting. Cartooning skills include bones and non-linear morphing.

**Draw alignment grid on top.** If this option is selected, the alignment grid will display on top on the visual elements. If this option is not selected, the alignment grid will display under the visual elements but over any background image.

**Grid Transparency.** Adjust the grid appearance from dark grey (0) to fully transparent (100).

**Canvas color.** This is the color of the application window outside the movie frame area.

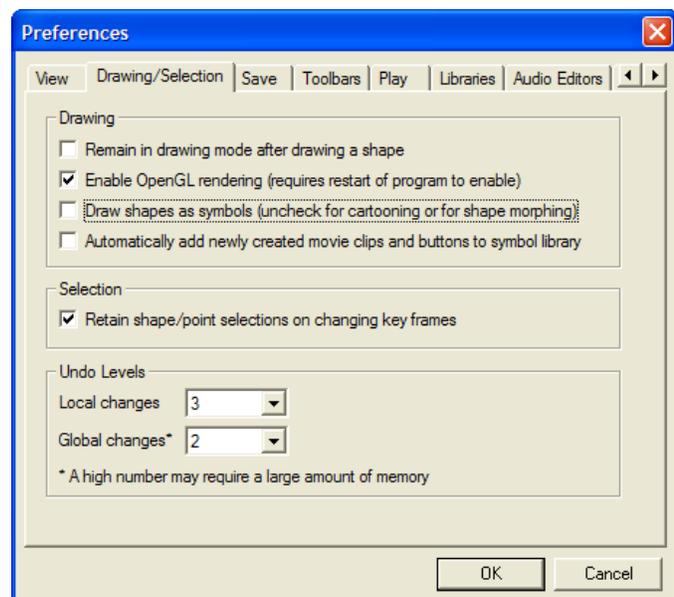
**Font size.** This font size setting affects the Data View windows and some of the dialogs.



### Drawing/Selection Preferences:

*Draw shapes as symbols.* With this option checked, shapes are drawn as symbols. This saves a great deal in the size of the final swf. The downside is that points to the shape cannot be selectively moved in a frame without affecting other frames. If you are doing character animation, turn off this option.

*Remain in drawing mode after drawing a shape.* If this option is selected, you are able to repeatedly draw shapes without having to reselect the shape drawing tool. If this option is not selected, the select/move shape tool becomes the current tool after drawing a shape.



*Automatically add newly created movie clips to symbol library.* Useful when designing components or otherwise want your movie clips to have linking capabilities.

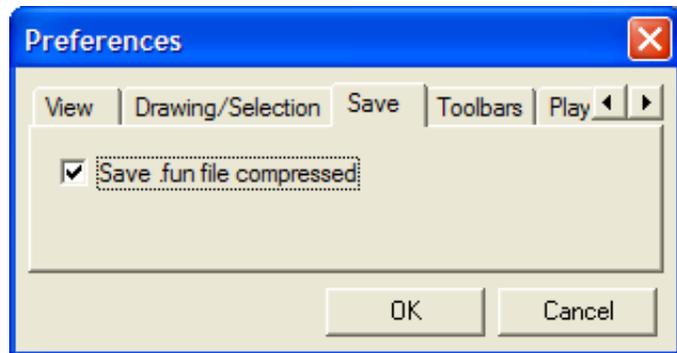
*Enable OpenGL....* OpenGL is an open source application programming interface used by KoolMoves for better anti-aliased rendering of shape borders . Widely supported and efficient, it requires adequate hardware for best results – this option may not operate properly on windows 98 or ME operating systems.

*Retain shape/point selection on changing key frames.* If this option is selected, your current shape or point selection will be preserved when you select a different key frame. If this option is not selected, the current shape or point selection will be lost when you select a different key frame.

*Undo Levels.* Users can set the Undo history level.

### Save Preference

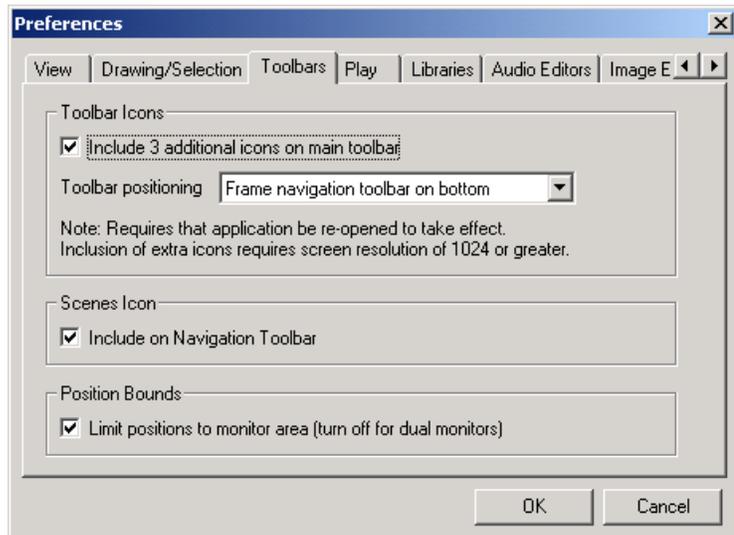
Starting with KoolMoves version 5.6, users now have the option to save .fun project files in a compressed format. Uncompressed fun file will open in newer versions of KoolMoves, but compressed fun files can not be opened with older versions of KoolMoves.



### Toolbars

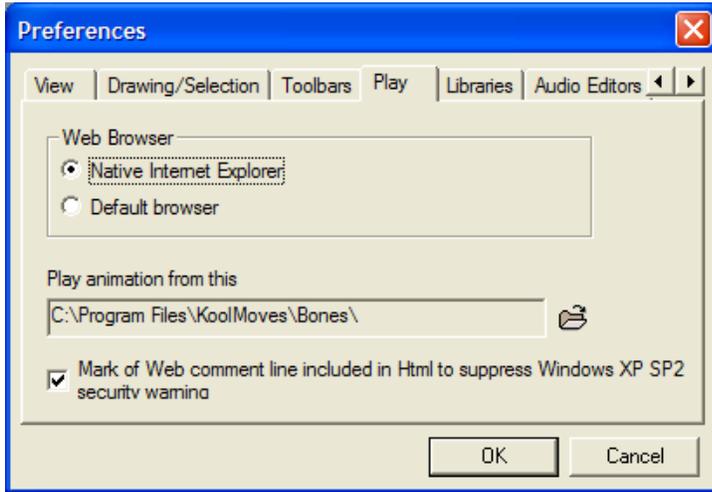
In Toolbar Icons, 1) the first checkbox refers to the [Actions and Sounds Overview](#), the [List of Shapes](#) and the [List of Frames](#) icons in the Menu toolbar, and 2) the navigation toolbar can be placed at the top or bottom of the GUI. (These actions will take effect the next time KoolMoves is re-opened.)

Uncheck the Scenes Icon box to remove the current scene indicator field and the Scenes icon from the [Navigation bar](#).



Uncheck the Position bounds box if you have dual monitors and wish to spread the KoolMoves interface components across the two monitors.

## Play Preferences:



By default, animation preview uses the Windows internal Internet Explorer web browser. You can override this to use your default browser.

You can assign the directory from which movies will play. Every time you open a fun file, the play directory will reset to this directory. This can also be set via the menu with Play > Set Play Directory...

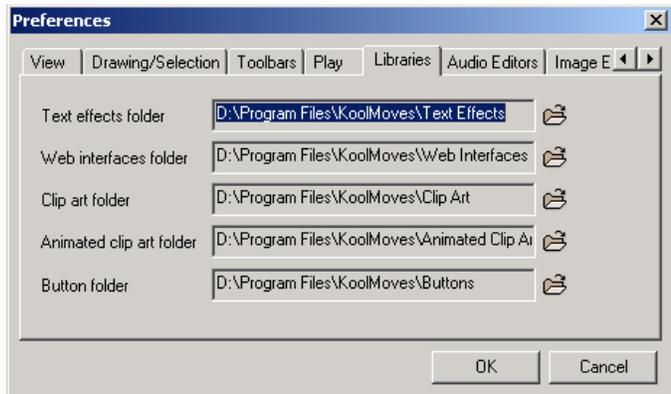
You have the option to add or not add the Mark of Web comment line

<!-- saved from url=(0013)about:internet -->

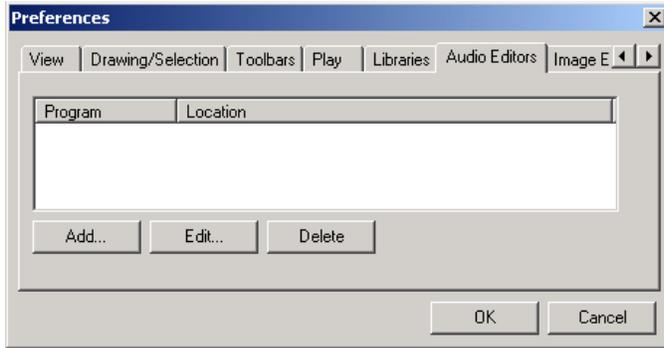
to all generated html files including the temporary files used for preview. With the introduction of Windows XP SP2, there have been changes to Windows which affect Flash animation. A reference for this is [http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_19542](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_19542). A more general reference is [microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.msp](http://microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.msp).

## Library Preferences:

Here you can change the default file directories for each of the designated libraries.



### Audio Editors Preferences:

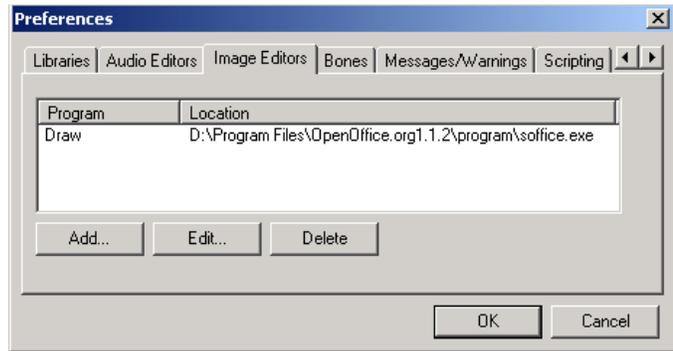


You can assign up to 6 audio editors that can assist you in viewing and editing sound for frames and buttons. You do this by pressing the Add button. A popup will be displayed. Press the browse button identified by 3 dots and select the .exe corresponding to an audio editor.

In the course of adding sounds to frames and buttons, you can press the Edit Sound button and a menu will appear with the sound editors you assigned. Selection of a sound editor from the menu will invoke that editor. See the [Audio Editors](#) entry in Appendix D for a discussion of recommended freeware and shareware audio editors.

### Image Editors Preferences:

You can assign up to 6 image editors that can assist you in viewing and editing images. You do this by pressing the Add button. A popup will be displayed. Press the browse button identified by 3 dots and select the .exe corresponding to an image editor. The image editors are accessible through the paintbrush icon on the Image Fill page of the fill popup.



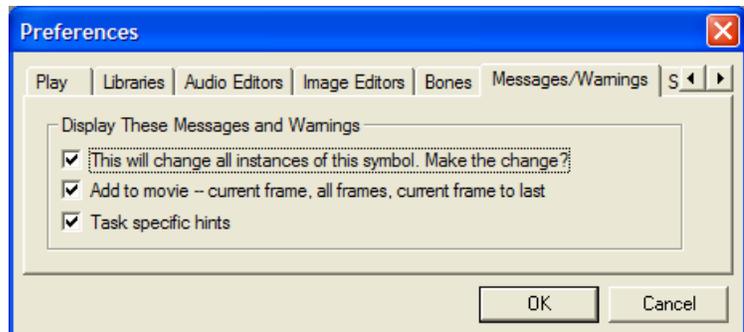
### Bones Preferences:



Set the color for animation bones. Click on the color box to open the Color Option dialog.

### Messages/Warnings

Set Checkboxes to control display of selected warning messages.



## Scripting

Set the Font for the [Action Script Editor](#).

## Options Menu

The Options Menu allows you to customize the appearance and actions of the stage and canvas.

### Reposition Movie Frame



Click and drag cursor on canvas to permanently reposition it. As opposed to the Pan command, screen refreshes will not change the location of the stage after a Reposition.

### Pan Canvas



To preserve valuable drawing area, KoolMoves does not have scrollbars. To navigate over the drawing area, particularly in the case of zooming, click on the Pan button (mouse pointer will change to a hand). Depress and drag the left mouse button to move the entire canvas.

### Zoom Commands



On the Menubar you can also click the Zoom icon to open the Zoom menu, or click on the [Magnification](#) list box (Chapter 6) to change magnification.

**Zoom Dragged Rectangle** – centers screen on selected area and zooms in on that area.

**Zoom In on Click** – centers screen on click point and increases magnification by one.

**Zoom Out on Click** – centers screen on click point and decreases magnification by one.

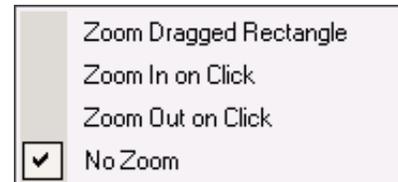
**No Zoom** – equivalent to setting magnification to 1. Stage position is reset to last position set by Reposition Movie Frame command.

### Grid

For alignment assistance, you can turn on a grid and set its size with Options > Grid. Display of the grid on over or under the animation is set in File > Preferences. To further assist in aligning visual elements, turn on Options > Snap to Grid.

### Show All Masking (Ctrl-M)

Toggles display of masking in author time environment.



## View Associated Joints

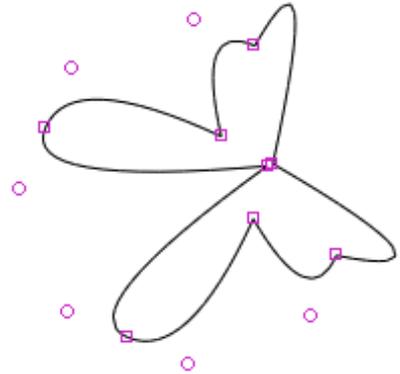
See the [Bones](#) entry in Chapter 6 for a discussion of shape joints.

## Antialiased Rendering

High quality rendering means anti-aliasing. Anti-aliasing is an image filtering technique which alters the boundary of an image to eliminate what are called the "jaggies".

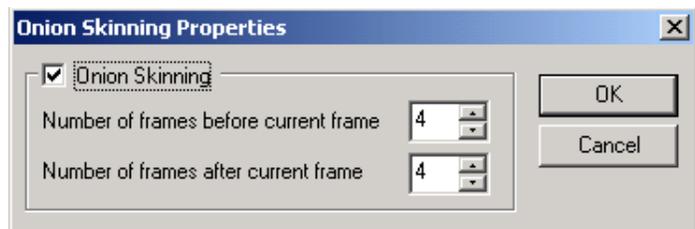
## Wire Frame Rendering

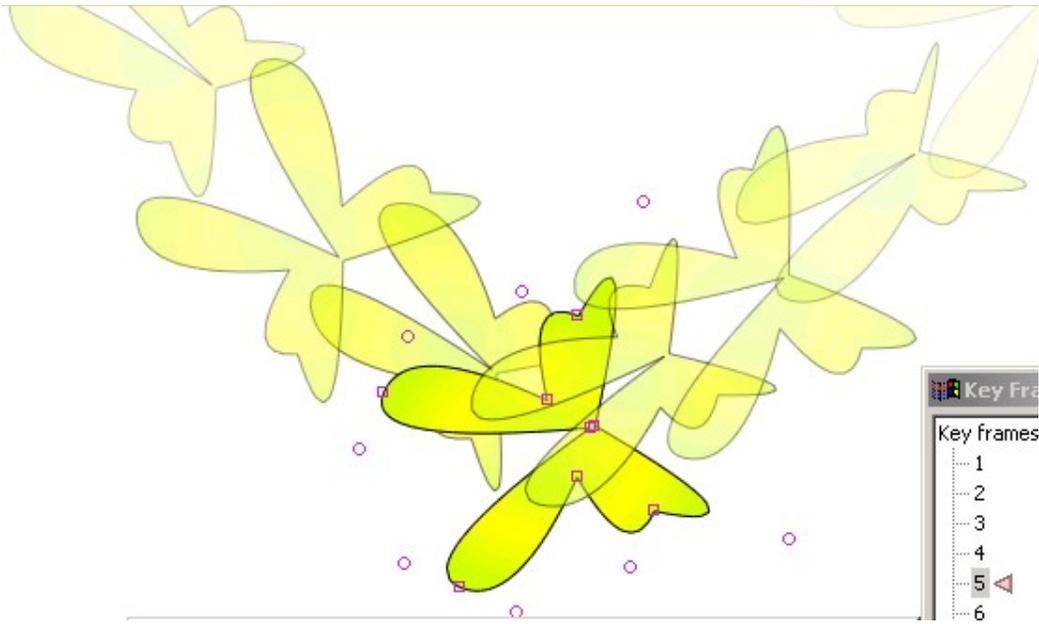
Use this to turn off all appearance settings and just see the outline of your shapes.



## Onion Skinning

Onion skinning allows you to view, in addition to the current frame, frames before and after the current frame. As the frames are further away from the current frame, the dimmer they appear. Onion skinning can be used to observe the movement or changes in the animation and ensure that the animation is fluid. To turn on onion skinning, select Options > Onion Skinning.





## ***File > Export Settings***

### **Flash Movie (SWF File)**

Export Settings affect how the final swf performs in the Flash™ Player.

#### **JPEG Quality**

A number can be selected from a low of 20 to a high of 95. A low value results in poor image quality but relatively small file size. This value is set in File > Export Settings and in Movie > Background Image. Despite the name, this quality setting applies equally to GIF, PNG, and other types of imported images as well as JPGs.

#### **Sound Compression**

The Flash file format provides different levels of sound compression for WAV files expressed as the number of bits (2 - 5) used to compress 16 bit sound where 2 bits provide the lowest quality and 5 the highest.

#### **Flash Version**

KoolMoves will automatically select the minimum Flash Player version that accommodates the functionality in the animation and action scripting, ranging from Flash 6 to 8. Click on the Reason button to see what element of the movie influenced the decision.

#### **Resource Loading**

By default, KoolMoves exports all images and sounds to the beginning of the main movie or the movie clip which references those images and sounds. You may want to change this so they load as needed rather than at the beginning. The first method causes a delay in starting the movie; the second method can cause jerkiness as the movie plays.

#### **Compression**

Flash 6/7/8 movies can be Zlib compressed in order to make them much smaller (about a factor of 2 on average). The disadvantage is that the movie will appear blank if the Flash player is version 5 or lower. Default value is on starting with KoolMoves 5.0.

#### **Dynamic Text**

You can ignore the backward compatibility option unless your movie is exported as Flash 6/7/8. Flash 6/7/8 handles dynamic text differently than Flash 4/5 because of the introduction of action script text fields. In Flash 4/5, the variable name was the object name but in Flash 6/7/8 an instance name was also added. So, txt1 = "hello" works for a swf exported as Flash 4/5 but not for a swf exported as Flash 6/7/8 unless this check box is checked.

#### **Interface Components**

There are currently a total of 7 components: scroll pane, list box, check box, push button, scroll bar, database query object, and tool tip; they can be added through the Tools Toolbar or through ActionScript (see Components section in Examples folder for ActionScript examples). There are detailed examples and readme files in the Components section in the Examples folder.

Interface components require Flash 6 player or later. If you add components via ActionScript and not through the KoolMoves GUI, be sure to check "include interface components" in Export Settings so that KoolMoves will add to your exported swf all the component creation, draw, and interaction routines.

### **Local Playback Security**

Selecting "Access Local Files Only" lets the published SWF file interact with files and resources on the local system only. Selecting "Access Network Only" lets the published SWF file interact with files and resources on the entire network.

### **HTML File**

These options affect how the movie will act and appear in a browser.

### **Movie Loops**

To set the looping of the Flash movie to on or off in the HTML file, set the looping property in File > Export Settings > HTML Settings. Click on the menu item to toggle its value. This parameter sets the value of the loop parameter in the JavaScript in two places in the HTML code. Alternatively, you can put a stop action at the last frame using Views > Actions and Sounds.

### **Transparent Background**

For this feature to allow a web page's background to show through the movie's background, you will need to use the WMODE parameter in the html object code on your target webpage. The quick and dirty solution is to have KoolMoves create an HTML page when you export your project to a swf file, and copy the applicable code from that html file to your target web page.



## Chapter 3 – Help Features & Tutorials

There is extensive context sensitive help available for most dialog screens. Look for the help icon below the window name.



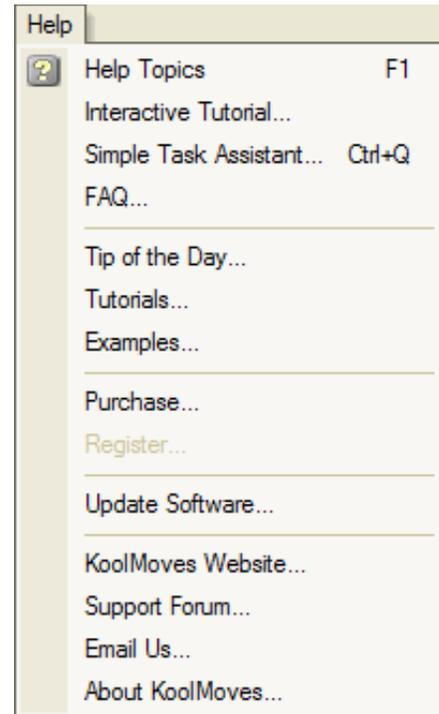
In addition to the Configuration Wizard covered in Chapter 1, KoolMoves has additional user assist dialogs (Interactive Tutorial, Simple Task Assistant) available under the Help Menu, and a simplified interface style (Wizard) for creating Banners, Text Effects, Image Effects and Slide Shows.

### Help Menu

The Help menu provides many levels and types of support, from hyperlinks to the online tutorials and examples located at the KoolMoves website, to tutorials and step by step guides incorporated within KoolMoves. In addition to these, you can view samples created with KoolMoves by going to the Examples folder (\Program Files\Koolmoves\Examples).

The tutorials and wizards address basic animation tasks and internet related operations and exist to assist the new KoolMoves user in quickly creating quality animations and exporting them to the web.

For tough questions, you can tap into the expertise of many people at once on our forum at [www.flashkit.com/board/forumdisplay.php?forumid=24](http://www.flashkit.com/board/forumdisplay.php?forumid=24). The forum is also a great place to share ideas, get feedback, and find out the latest news about KoolMoves, Flash and related topics. KoolMoves users are encouraged to register with Flashkit and peruse the available Flash resources.



### Help Topics



Click on this to open the HTML Help files included with the KoolMoves installation. There are Table of Contents, Index and Search tabs.



## Interactive Tutorial



The Interactive tutorial steps the new user through the author time environment in creating text and image objects, placing them on the stage, and then applying effects.

## Simple Task Assistant

The Simple Task Assistant (STA) provides menu driven assistance in completing moderately difficult tasks. Select a task and click on OK to continue.

Unlike the Interactive Tutorial (which illustrates how a task is done without assistance), the STA compresses the actions required into a minimum number of interactive screens; after completing each screen, click on Next until the task is completed. This allows the new user that needs to create a movie, and doesn't have time to read the manual or work with the tutorials / examples, to achieve the list of available tasks in the least amount of time.



## FAQ (Frequently Asked Questions)

Click on this to check out the KoolMoves FAQ file, or just check out [Appendix F](#) in the manual.

## KoolMoves Website...

Click on this to open the KoolMoves website with your default browser.

## Support Forum...

Click on this to open the KoolMoves help forum on the Flashkit website with your default browser.

## Email Us...

Click on this to email [support@KoolMoves.com](mailto:support@KoolMoves.com) with your default email application.

## Tip of the Day



Useful stuff to know that you can learn a little bit at a time. Uncheck the “Show tips on startup” box to turn this feature off.

## Online Tutorials and Examples

- [www.koolmoves.com/support.html](http://www.koolmoves.com/support.html) - a hyperlink located in the Help menu. KoolMoves users can contribute tutorials and examples to this site.
- [www.necromanthus.com/main.html](http://www.necromanthus.com/main.html) - has many ActionScript examples.
- [www.bridel.org](http://www.bridel.org) - has many ActionScript examples.
- [www.waterlijn.info/km](http://www.waterlijn.info/km) - has many ActionScript examples.
- [www.flashkit.com/tutorials/index.shtml](http://www.flashkit.com/tutorials/index.shtml) - has a wealth of tutorials on Flash techniques, will require some adjustments to work with KoolMoves.

## Purchase

Provides a link to [www.koolmoves.com/purchase.html](http://www.koolmoves.com/purchase.html), where the user with a demo copy can purchase a registration key.

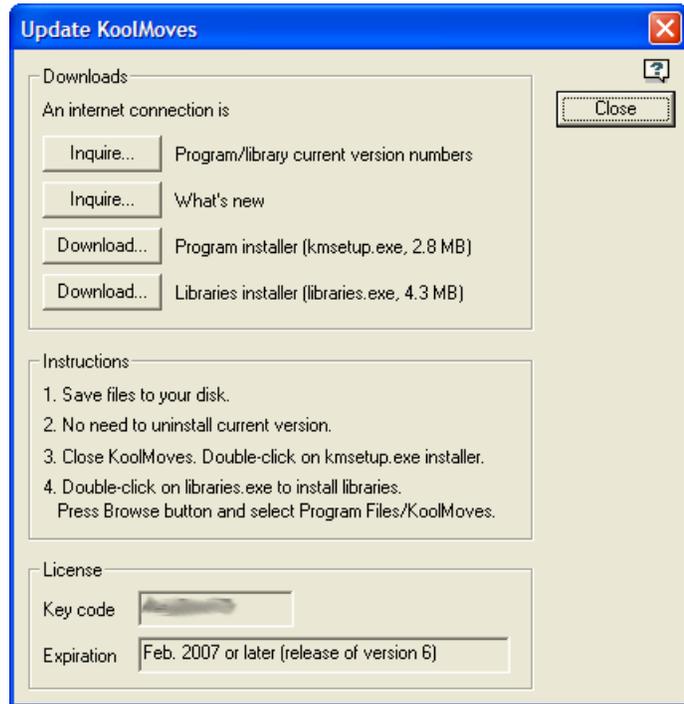
## Register

Use your purchased registration key to make your demo copy of KoolMoves fully functional.

## Update Software

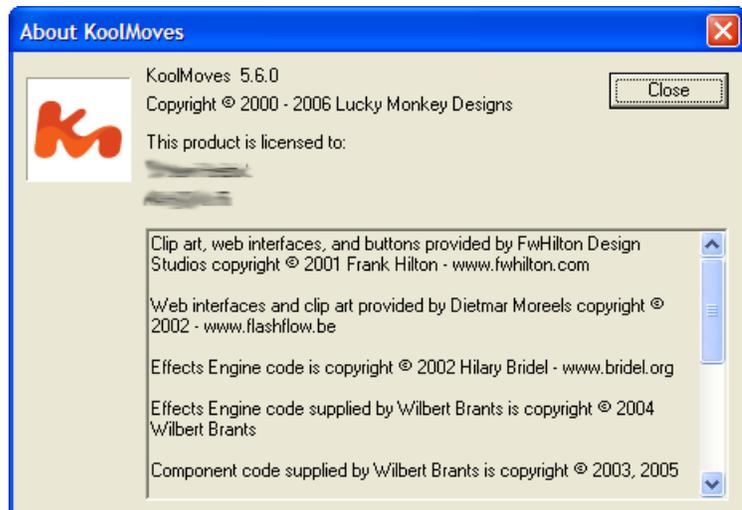
Use the Update Wizard to check for the latest versions of the KoolMoves program files and libraries. You can also find when your upgrade policy expires .

Contact information is located in the context help window (question mark in the upper right corner).



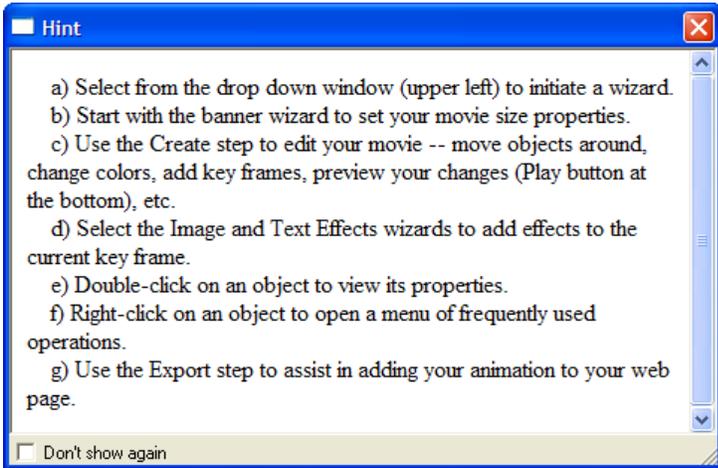
## About KoolMoves

In addition to information about the KoolMoves application, this window displays the registered user's name and registration number.



## Wizard Interface Style (aka Wizard Mode)

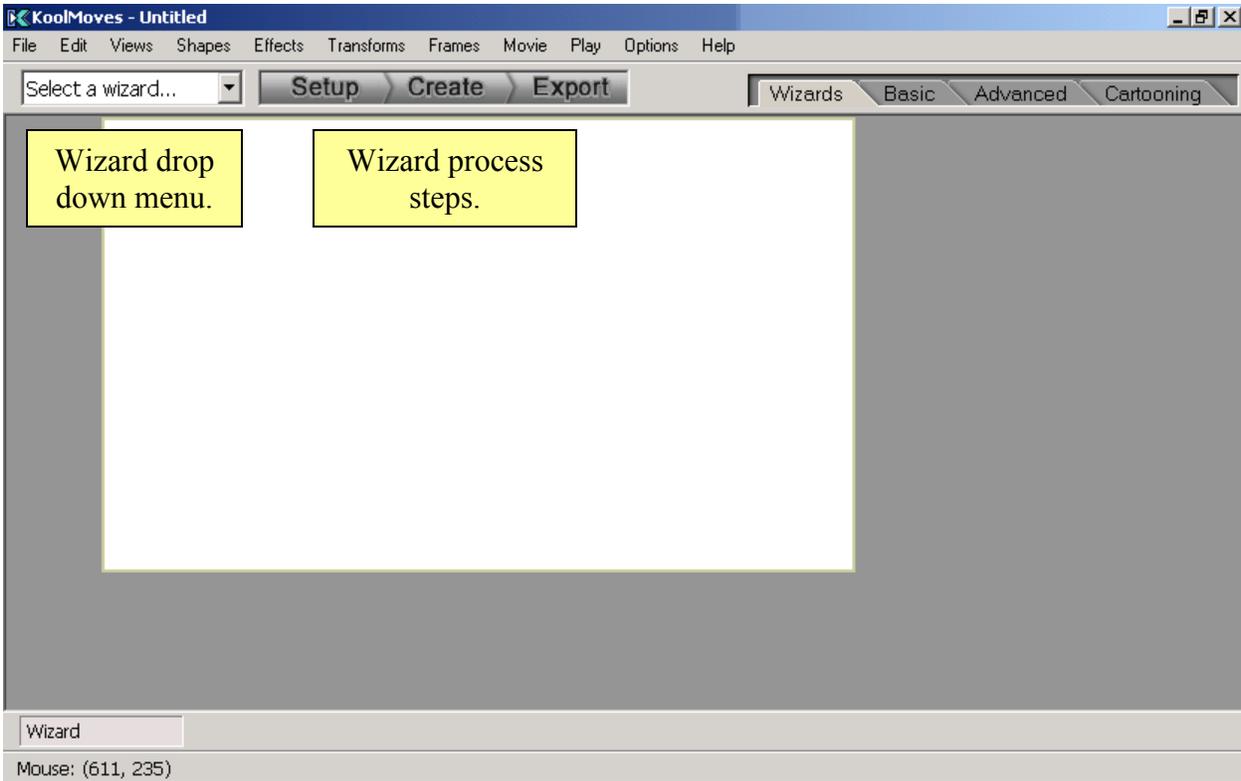
Click on the Wizards tab to change the Interface Style. A Hint window will open, displaying useful tips on how to use the Wizards Interface Style.



The hints describe the general process of using the Wizard Mode to create a movie.

Check the “Don’t show again” box if you do not want to see this hint box the next time you enter Wizard Mode.

The Wizards Interface Style is much simpler than the Advanced Mode displayed in Chapter 1: the Toolbox, Menu Toolbar and Navigation Bar are completely missing (however, the Basic Toolbox will be opened in the Create step to edit effects when applicable – more on that later). Meanwhile, a few new features (mentioned in the Hints window) have been added.



### Three Steps – Setup, Create and Export

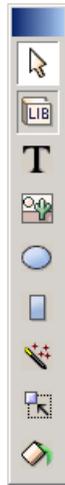
Each wizard is designed to help the novice KoolMoves user achieve a specific goal. Selecting a wizard from the drop down menu will launch that wizard’s Setup step. To manually launch a wizard, click on the Setup button. The Setup step is unique for each wizard.



Wizard mode has three buttons prominently displayed at the top center of the GUI – Setup, Create and Export. If you have canceled out of the Setup or Export steps, you can restart the guides by clicking on the Setup and Export buttons.

The Create step exists for the sole purpose of allowing the user to do some free form editing. Clicking on the Create button opens the Basic Interface Style Toolbox. The Basic Toolbox is a stripped down version of the Advanced and Cartoon Interface Style Toolbox. These tools and interfaces are covered in depth in other parts of the manual. In descending order there are:

- Select/Drag tool,
- Library interface,
- Static Text Box tool,
- Import Image interface,
- Ellipse Shape tool,
- Rectangle Shape tool,
- Shape Effects tool,
- Resize effect and
- Fill effect.



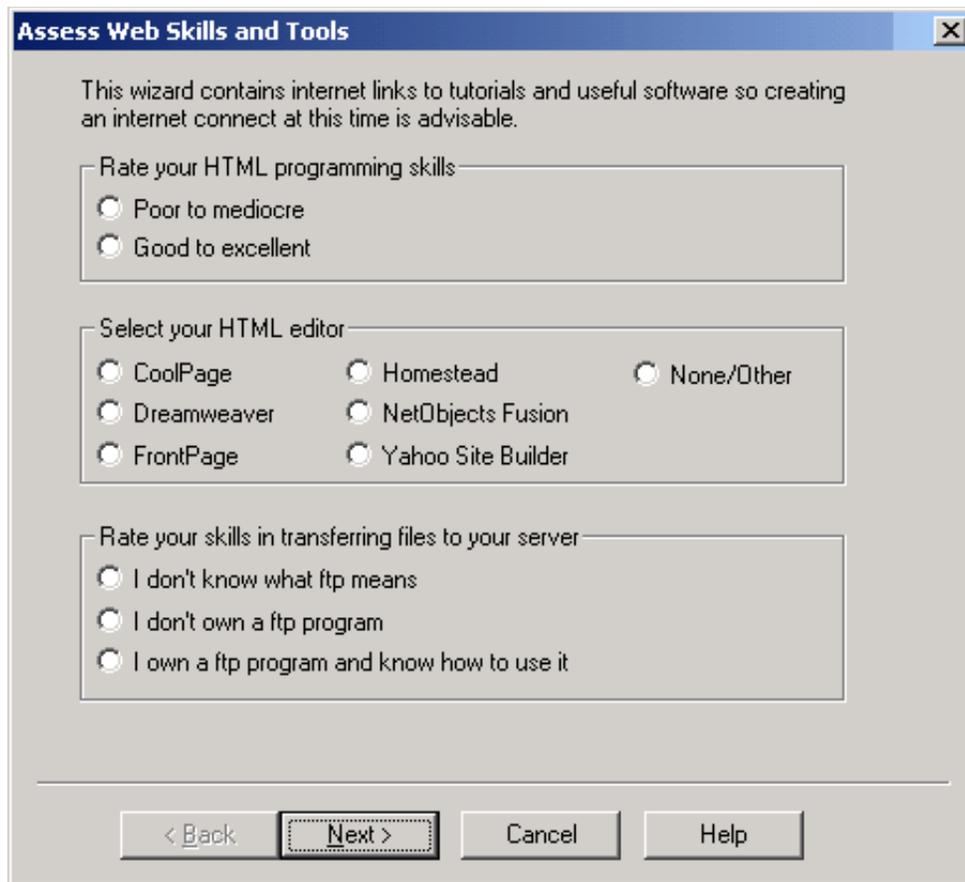
Note: Do not change Interface Styles in order to have more toolbox options unless you are willing to lose your wizard session settings.



Exporting a movie to a web page involves first saving your project to a Flash file (SWF), placing the swf file in the same directory as the target web page, and adding JavaScript so the movie displays properly on the target web page. Click on the Export button for a quick step-by-step guide to exporting your project to a web page.

The export process starts by opening an assessment window to identify which hint files you will need and what assistance to provide. If you do not have an html editor, KoolMoves will attempt to add the appropriate JavaScript to your target web page. If you have one of the html editors listed in the assessment screen (with the exception of FrontPage), KoolMoves will display recommended JavaScript code that you can copy and paste to the target web page. FrontPage users will be directed to a hint file that shows how to import the movie through FrontPage.

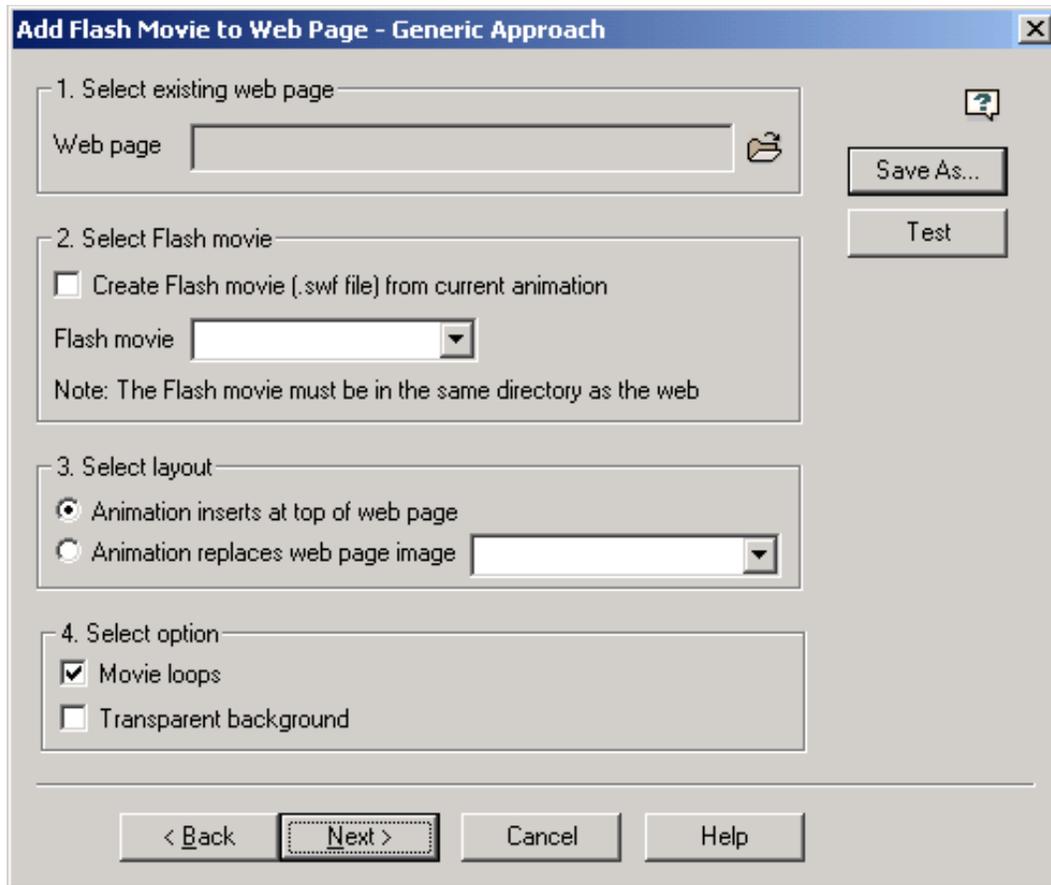
Your answers to the Web Skills and Tools questionnaire are saved for the duration of the session; you will need to complete this window each time you move out of and back into the Wizard Interface Style. Complete the form to the best of your knowledge and click on Next.



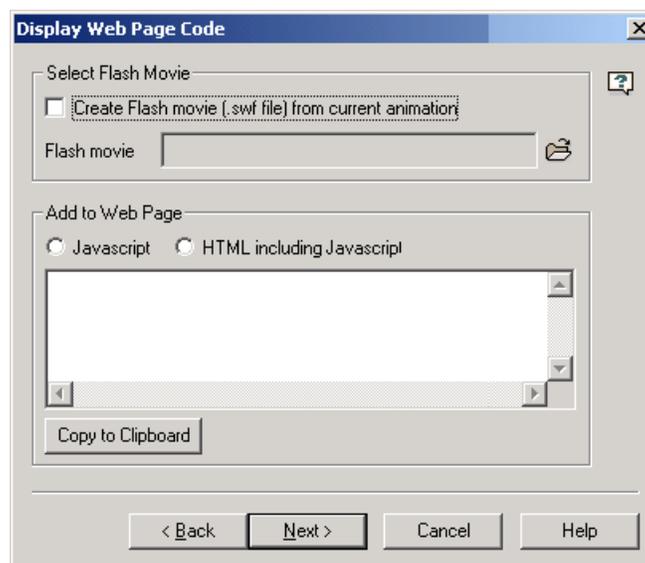
The next few screens will depend on your preceding answers concerning your skill level, the HTML editor that you have selected and how familiar you are with File Transfer Protocols (FTP).

If you have Poor/Mediocre HTML programming skills and did not specify an html editor, KoolMoves will load the Add Flash Movie to Web Page - Generic Approach window as shown below. This window consolidates information entry and output that, if done manually, would require several separate steps and some web page coding.

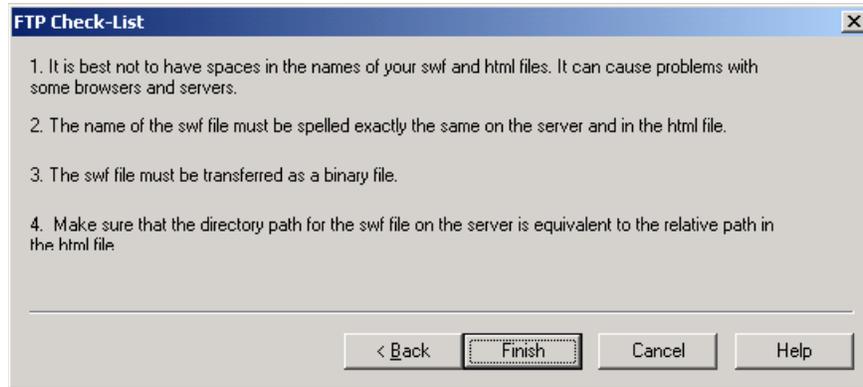
1. Browse to the web page that will contain your movie. Remember that your swf file needs to be saved to the same folder as the target web page.
2. If you haven't already, take this opportunity to export your project to a swf file. You can also choose to export a pre-existing movie.
3. Select a location where your swf will be placed in the target web page. This can always be changed later with your web page editor.
4. Set the loop and background transparency options for your swf file. For information on the html coding required to ensure this feature works, please see the [Transparent Background](#) entry in Chapter 2 – Preferences and Options.



If you specified an HTML editor other than FrontPage, then instead of the Generic Approach window shown above, you will get the Display Web Page Code window. Use this screen to capture sample JavaScript and HTML code that has been generated for the specified movie, and place this code in the target web page via your web page editor.



As mentioned earlier, if your html editor is FrontPage, it is recommend that you follow the provided tutorial and have FrontPage insert all necessary JavaScript coding. The final window has some general html/FTP hints. Click on Finish to close the Export step. If the Generic Approach worked, the target web page should have JavaScript linking to the specified movie.

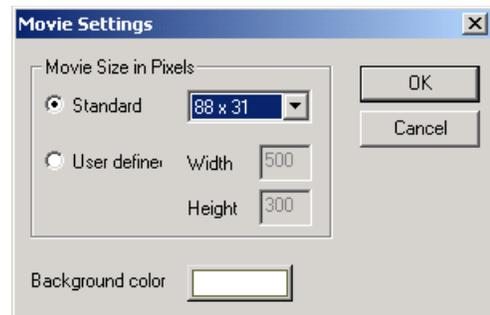
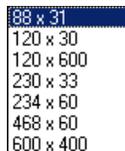


Once a wizard sequence of windows has been started you are not required to follow the steps to the end, but can cancel at any time. On the other hand, if your movie consists entirely of text and or image effects, click on the Export button when you are ready to save your project and export a movie.

If you've edited the movie using the Basic Toolbox, there is a chance that the interface style may have changed from Wizard to Basic. If you want to use the Export button to save your project and export a movie, click on the Wizard tab and then select the Banner wizard to make the Export button responsive. You will be re-entering the Wizard Interface Style, so you will need to re answer the Assess Web Skills and Tools questionnaire.

## Banner Wizard

The Banner Wizard is the simplest wizard – it just guides the user in defining the movie's stage size and background color. It has a drop down list of useful banner / movie sizes.



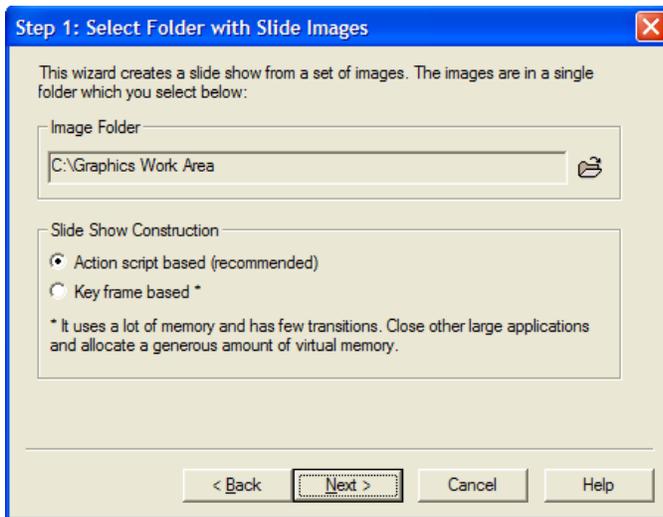
Once you click on OK, the Banner Wizard will advance to the Create step and the Basic Toolbox will be available.

The Text and Image Effects wizards guide the user in adding effects to the movie. The Slide Show wizard is the most complicated, guiding the user in defining images to be use and the transition timing.

## Slide Show Wizard

This wizard creates a series of key frames to hold slide show images. The source images can be GIFs, JPGs or PNGs and must be present in a single folder. The wizard queries for the folder name, allows you to put the images in order and then select the duration each image will display. A simple fade in/fade out transition can also be applied.

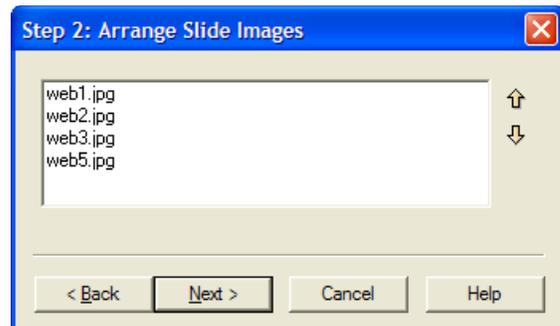
After the step-by-step process ends, you might want to add play buttons and add stop actions to the movie to give the end user control. The slide duration can be edited in the List of Frames using the Frame Duration icon.



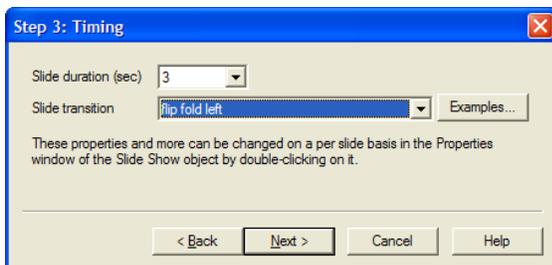
In Step 1, select the folder that contains your slide show images.

Unless you are planning on exporting tween captures to create an animated gif, or adding features to the slideshow that require key frames or tweening effects, it is best to use the ActionScript version. This will insert a component that includes navigation controls, and only take up one Key Frame.

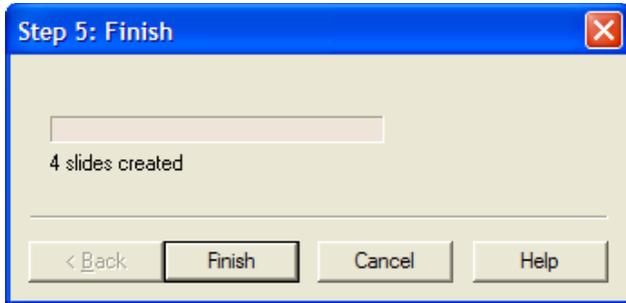
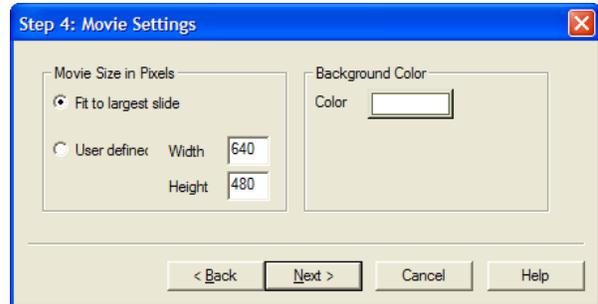
In Step 2, arrange the images in the desired sequence. Select a file name and click on the up or down arrows to change its relative position.



In Step 3, set the slide duration and whether or not there will be a transition effect applied to this slide.



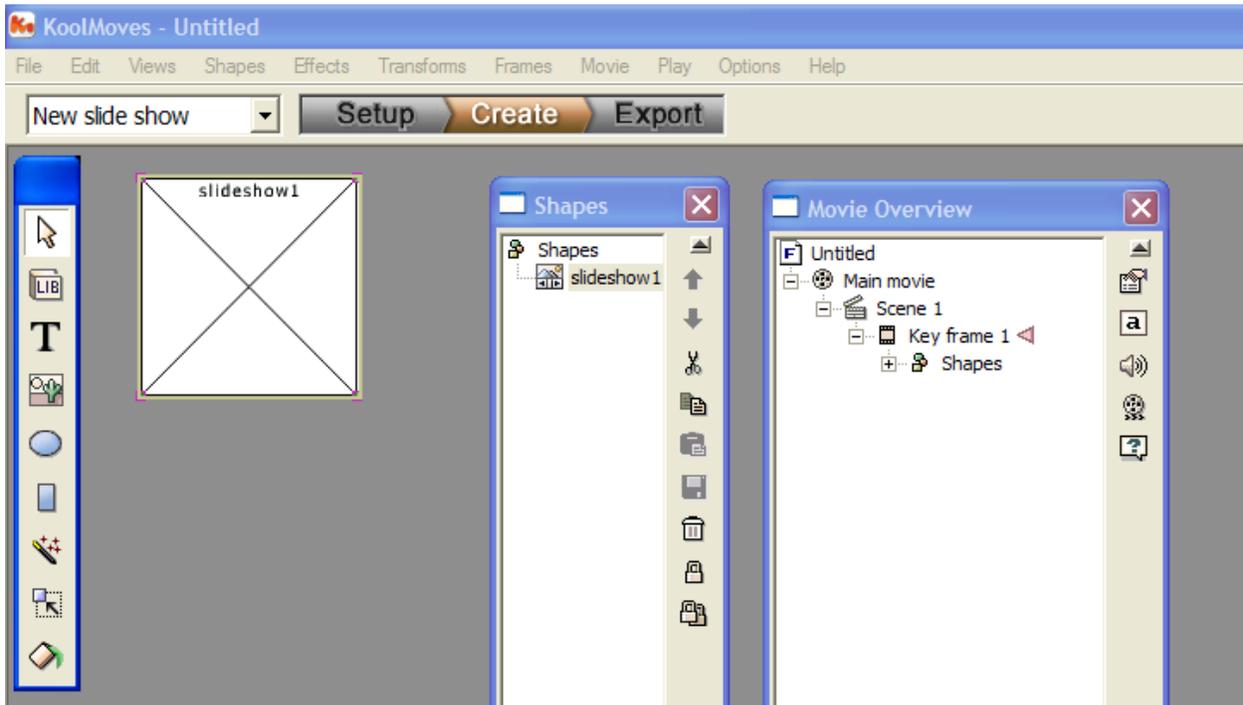
In Step 4, set the movie dimensions and background color.



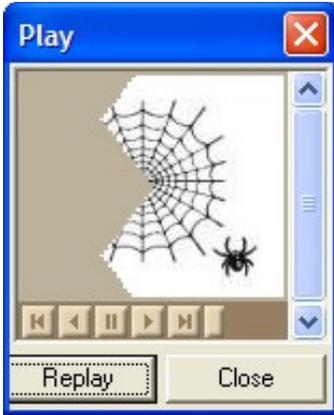
Click on Finish to enter the Create Step.

In this particular project I was evaluating different images for use on a web page, so I organized them into a slideshow for easy comparison. Use the Create step to edit and preview the slide show.

You've now created a single frame movie, though the ActionScript transition effects will make it look a lot larger.

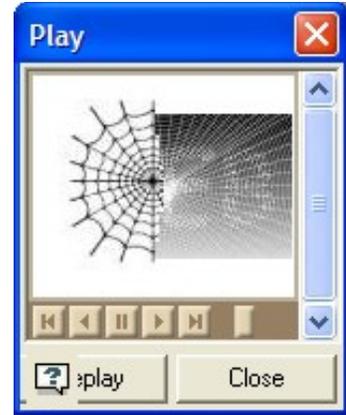


Double click on slideshow1 to open the Properties data view and set the slideshow component options.



web1.jpg appearing

The SlideShow component is covered in detail in Chapter 14. For now, it is just important to understand that if the Key Frame based option had been selected in Step 1, all navigation controls would have to be added by the animator.

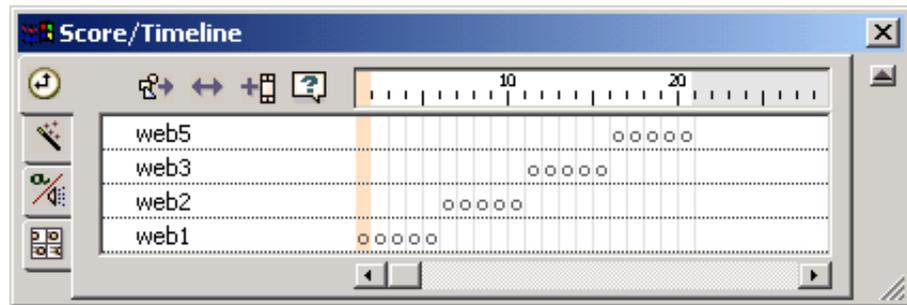


web2.jpg appearing

To illustrate a few things, let's see what we get when we create the same slideshow, but this time choose the Key Frame based option and the Fade In effect. For one thing, there would be a key frame for each image, and a tween frame consistent with the settings at the time the slideshow was created. Open the Views menu, the Score/Timeline, List of Shapes and List of Frames Views.

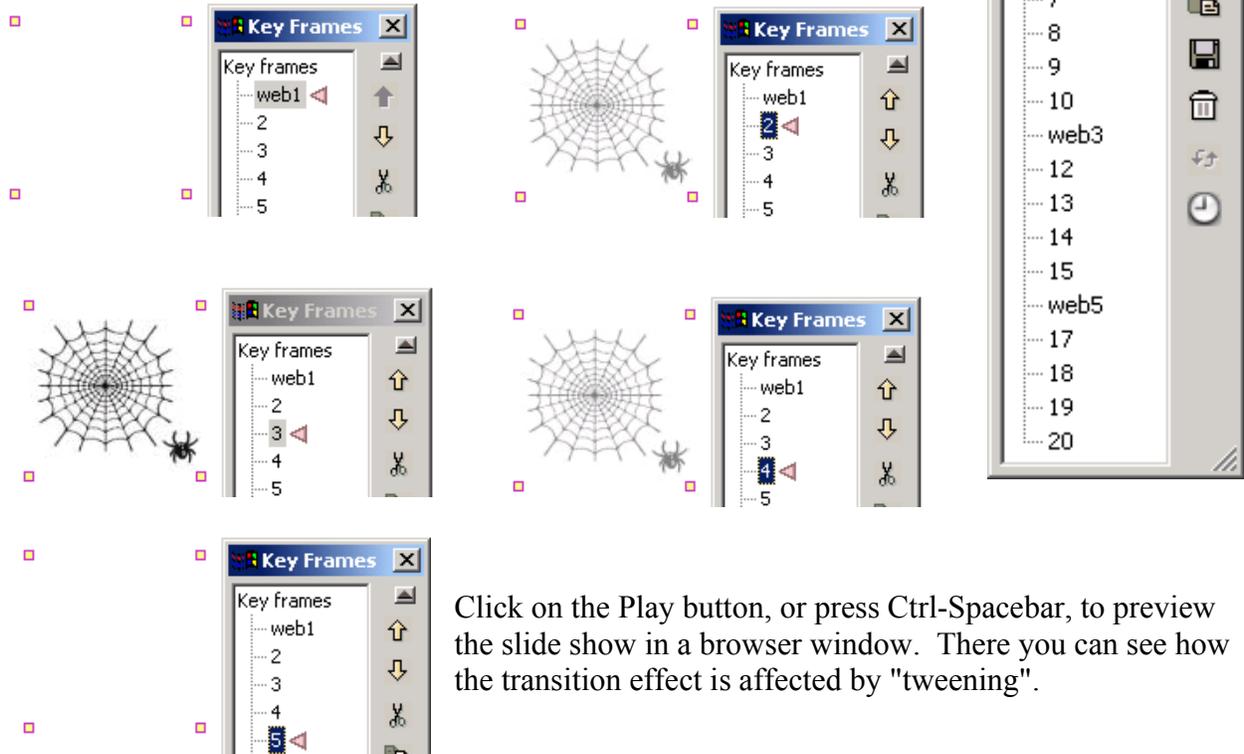


The List of Shapes displays what is present on the current key frame, while the Timeline view displays all the objects in the movie, and on which frames they are present. You can see in the timeline that each image is used five times. The images are initially transparent because of the Fade In effect.



In the List of Frames, you can see that the key frames where the images are first used have been named for the image. This is a built in navigation and programming aid, to help you add features like buttons to control the flow of the slideshow.

In the sequence of images below, the shape web 1 has been selected. As we step through the frames, you can see web1's transparency range from 100% to 0 back to 100%.



Click on the Play button, or press Ctrl-Spacebar, to preview the slide show in a browser window. There you can see how the transition effect is affected by "tweening".

When you are through editing your slide show, the next step is to save your project to a .fun file and export the movie to a .swf file. To take advantage of all the features available in KoolMoves, it is recommended that you read the File > Export Settings in Chapter 2 and File > Export Menu in Chapter 4. Once the project has been exported to a movie, use your web page editor to position the slideshow on the desired web page.

However, for a detailed step-by-step guide to exporting your slideshow to a .swf file, click on the Export wizard button as discussed earlier in the Three Steps – Setup, Create and Export entry.

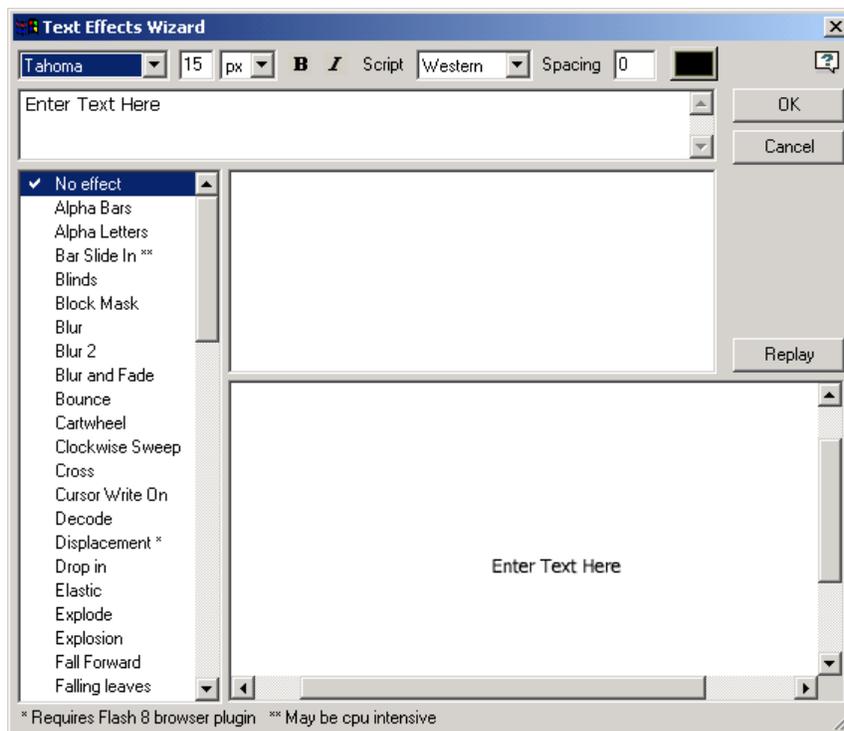


## Text and Image Effects

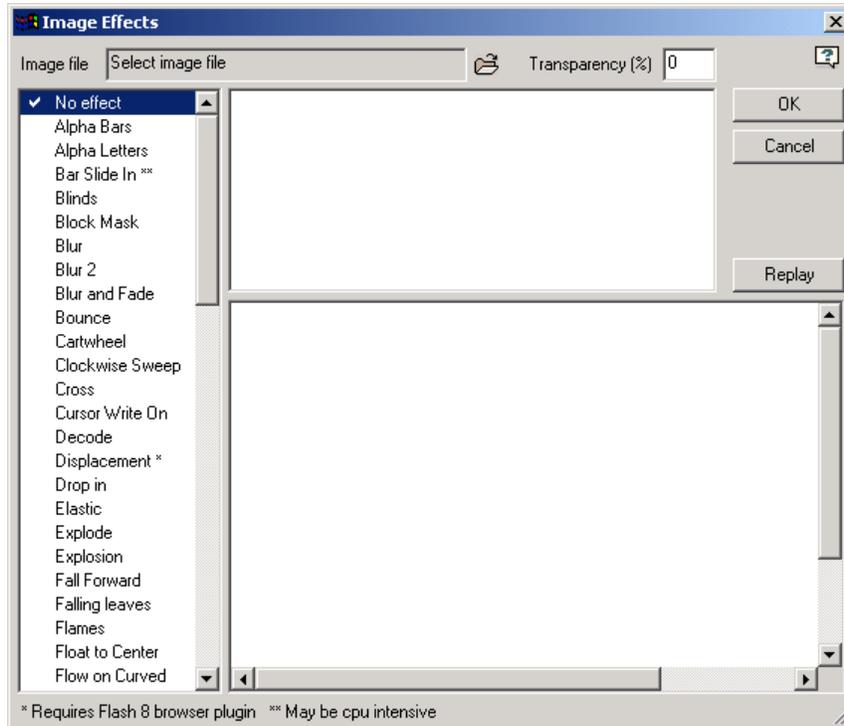
The Text and Image Effects wizards guide the user in adding text and image objects to the movie and applying motion script effects to the objects. These effects cannot be combined with regular fade, scale, spin, or rotate effects and a maximum of one effect can be selected. Applying motion script to static text will disable a drop shadow effect. To remove an effect, select "No effect" in the list of effects.

A motion script effect relies on action scripting to create the effect in the Flash movie. The effect is not viewable in the KoolMoves editing screen or its internal player, but can be viewed in the preview window associated with the motion script or in a web browser.

In the Text Effects interface, 1) select font details, 2) enter your text, 3) select the name of the effect from the list of effects, and 4) click on OK.



In the Image Effects interface, 1) select an image file, 2) set the transparency of the image, 3) select the name of the effect from the list of effects, and 4) click on OK.



These wizards stop with the Create step and open the Basic Interface Style Toolbox. This allows you to reposition or otherwise edit the object you've just added, and also allows you to add additional Text and Image effects to your movie. To launch the same wizard a second time, click on the Setup button. To switch from the text to the image effect wizard, or vice versa, use the drop down menu. Once you are ready to save you project and export a movie, click on the Export button.

## ***Help in Other Languages***

There are a number of machine translated versions of help in different languages on the KoolMoves support page. There is an old version of help for KoolMoves in Spanish at [www.geocities.com/powerxrv/Koolmoves/index.htm](http://www.geocities.com/powerxrv/Koolmoves/index.htm).



## **Part II – Menu Toolbar & Menu Commands**

**Chapter 4 – File Operations**

**Chapter 5 – Data Views**



## Chapter 4 – File Operations

In addition to file actions covered by the Menu Toolbar , this chapter includes file actions not addressed by the KoolMoves GUI outside of the menu.

### ***File > New (Ctrl-N)***



Clears most KoolMoves buffers and resets the KoolMoves GUI.

### ***File > Open (Ctrl-O)***



Opens a Windows Explorer type dialog to browse and open a .fun file.

### ***File > Save (Ctrl-S)***



On first save, this command opens a Windows Explorer type dialog to save current project to a .fun file. On subsequent saves, a backup file is created before the existing .fun file is overwritten. The name of the backup file is ~backup.fun.

### ***File > Save As***

No Icon. This command opens a Windows Explorer type dialog to save current project to a .fun file.

### **File > Export Menu / Export Icon**

The File > Export menu and the Export icon menu are very similar. The individual actions are covered below.



### **Eolas Patent Issue**

If you see a "Click to activate control" message when mousing over a Flash movie on a web page, this is because Internet Explorer was altered in April 2006 in compliance with a patent dispute ruling (Eolas patent) handed down in 2003. Adobe Flash, QuickTime, RealOne Player, Acrobat Reader, Sun's Java Virtual Machine, and Windows Media Player along with applications that embed into Web pages, running on Windows XP SP2 systems, are affected.

A fix developed by Adobe is included in the code provided by KoolMoves. This fix requires that a separate javascript file, called AC\_RunActiveContent.js, be included in the same folder as the html file. This file can be found in the KoolMoves\IE ActiveX Fix\ folder.

Ftp the .swf, .htm/.html, and .js files to your web server. The .swf file is binary and the name should be spelled exactly as you have it on your local disk. Avoid spaces in the name because spaces are a problem for Netscape. Ftp is shorthand for file transfer protocol which is the method used to transfer files from your computer to internet file servers. There are many programs which are easy to use some of which are free which will perform this transfer. If you don't currently have a program to ftp files, there are many free ones to choose from at [www.snapfiles.com/Freeware/network/fwftp.html](http://www.snapfiles.com/Freeware/network/fwftp.html).

## Add Flash Movie to Web Page

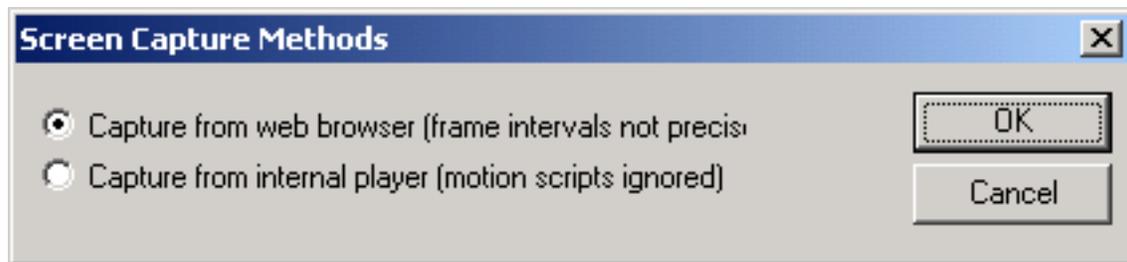
To run a Flash movie from your web page involves: (1) creating a swf file with KoolMoves and (2) adding JavaScript code to your web page which references the swf file. KoolMoves will create the JavaScript code for you. Both the swf file and your html file with the JavaScript are uploaded to your web server. This is analogous to creating an image in an image editing tool and adding code to your web page to display this image. Look under File > Export Movie for a number of different approaches to creating swf files and adding the necessary code to your web page. We recommend using the [Wizard Mode Export step](#) (see Chapter 3) if you have any problems.

## Export as Frame Captures / Capture Movie Frame

This function performs a screen capture of the current key frame, all key frames, or all frames including tweens, and saves the images to sequentially named files. The screen captures are saved in .bmp, .jpg, or .png format.

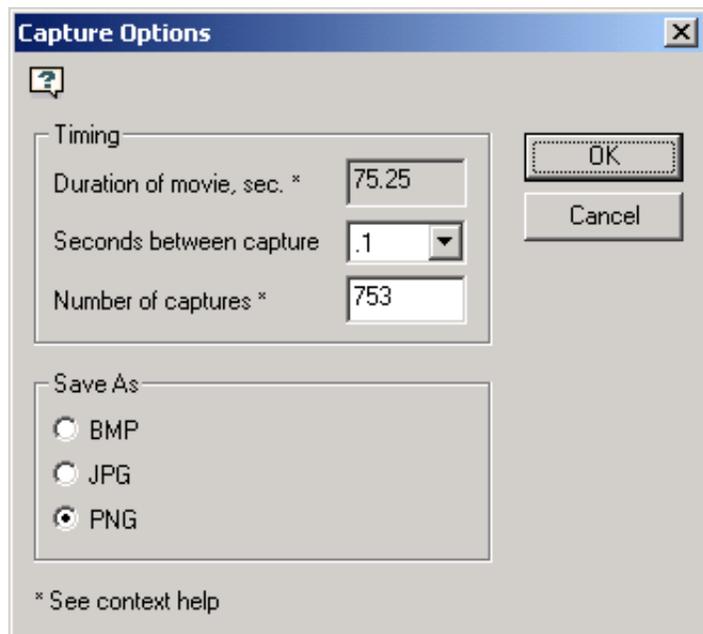
### Capture from web browser

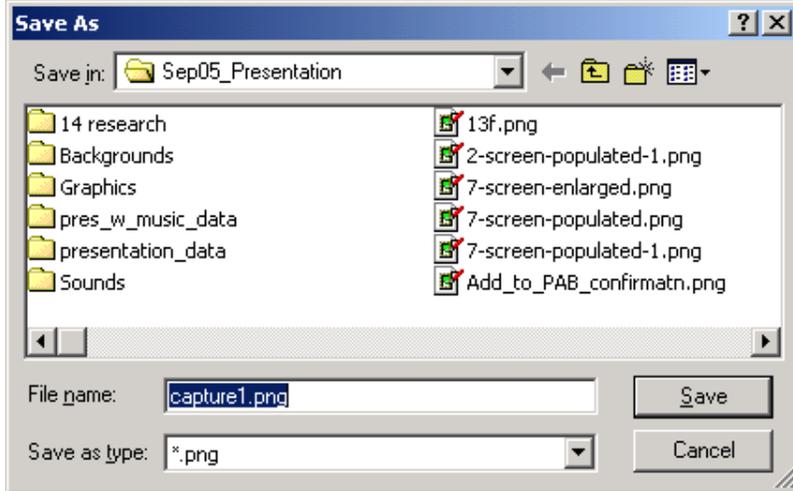
This option allows you to capture what will actually be displayed by your browser.



Select the frequency of screen capture and the number of screen captures. The duration of the movie is indicated but may be quite wrong. The duration is calculated from the number of main movie frames and the frame rate but this does not take into account the time it takes for embedded swfs and movie clips to play. Therefore, it may be necessary to manually increase the number of screen captures in order to capture the entire movie.

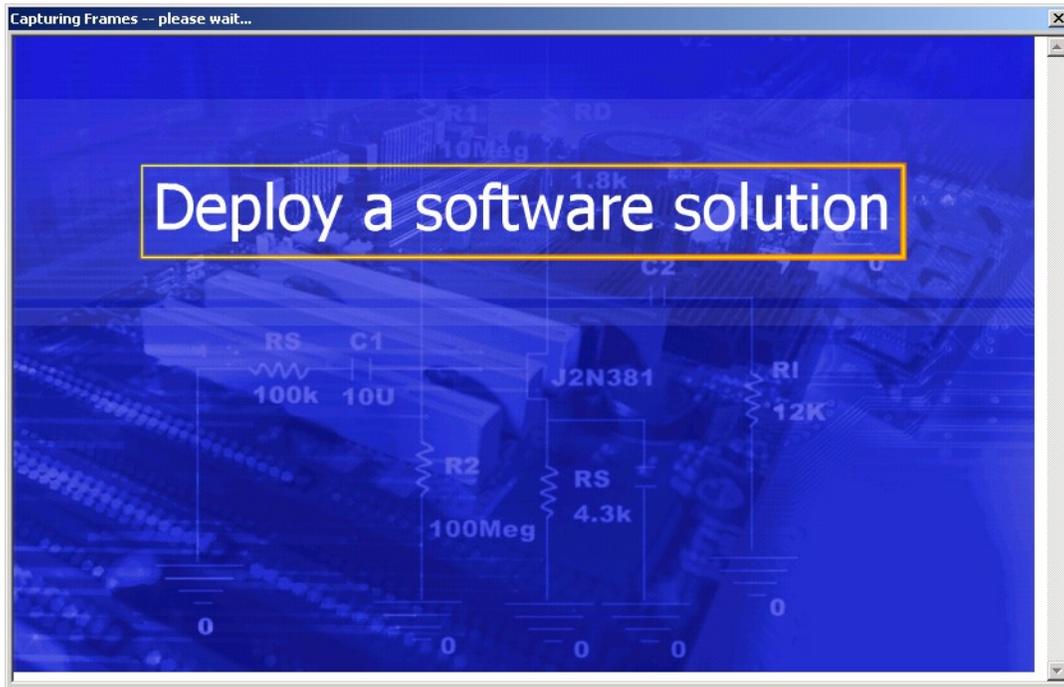
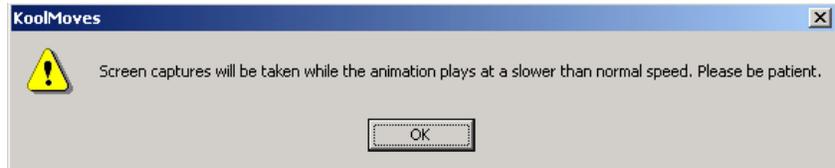
**Note:** Bitmap captures tend to be large.

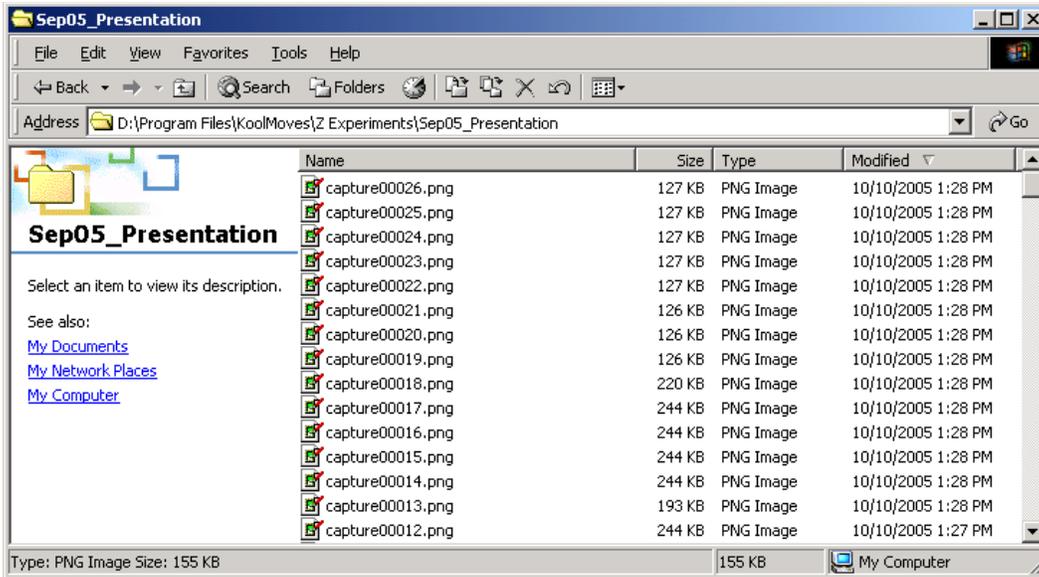




Select a target folder and file name for the first capture. Subsequent captures will be named sequentially based on this file name.

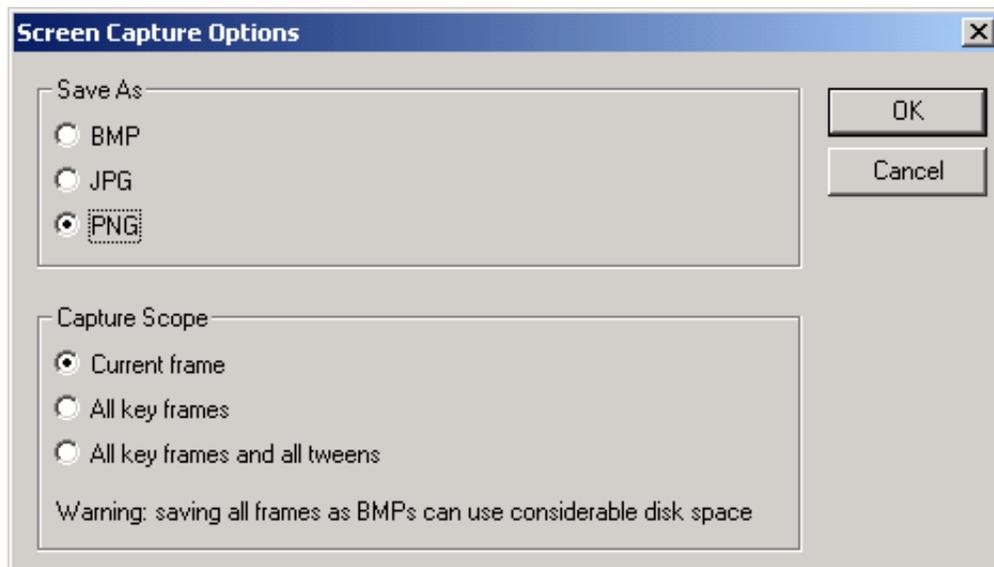
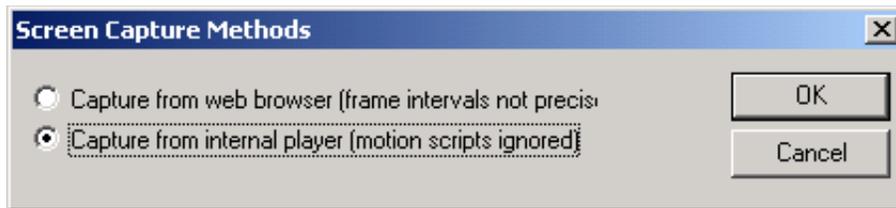
Saving screen captures while calculating tweens creates a load on your processor – the movie will run very slow.





These are the captures, at 5 minutes and about 3% completion. Capturing from the internal player works better if all you need are the key frames, or just the current key frame.

### Capture from internal player



You can create an animated gif by assembling a series of frame captures. Capture a series of frames and use software for creating animated gifs to transform the series of bitmaps into an optimized animated gif. There are many such specialized products available.

### **Export as SWF**

A KoolMoves project can be exported as a Flash™ movie (.swf format) with File > Export Movie > Save as Flash Movie. You can create a stand-alone .swf file, a .swf file and a .html template file which references the .swf file, or add a Flash movie to an existing web page. Both swf and html files are needed for the movie to play on the internet. Movie properties (width and height, speed, background color or image) can be set with menu items associated with Movie menu bar prior to exporting the files.

### **Export as SWF and New HTML Page**

KoolMoves can provide you with the necessary JavaScript code to call the Flash movie as one of the choices in File > Export Movie > Add to Existing Web Page. To replace a gif or jpeg image in an existing HTML file with a Flash movie, manually replace the <IMG SRC="....gif" WIDTH="..." HEIGHT="..." BORDER="0" ALT="..."> HTML code in your existing HTML file with all JavaScript code that KoolMoves provides starting with <OBJECT and ending with </OBJECT> by first copying the <OBJECT... code and then pasting over the <IMG code... You can use copy/paste editing capability in your web page editor or in Notepad. A word of caution: do not paste JavaScript code onto your web page using FrontPage. The code will be pasted into a block of text and the JavaScript will not run correctly.

The name of the .swf file in the HTML file must be the same as the name of the .swf file on the server. If your server does not accept a .html extension, you can rename the extension to .htm. It is best not to include spaces in the name as this causes problems with Netscape browser.

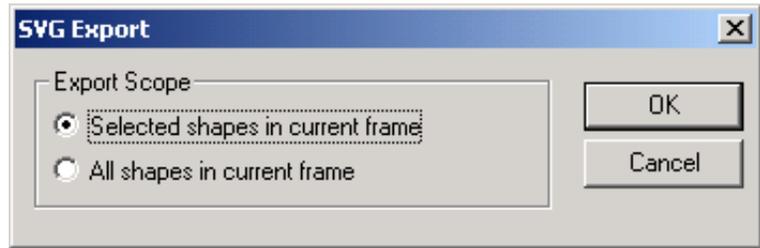
Both the .html and .swf files are next sent to your web site by ftp. When you ftp the .swf file to your web site, make sure you send it as a binary file. The .html or .htm file must be sent as ASCII. The better ftp programs like FtpVoyager do this automatically. <http://www.pageresource.com/putweb/> provides assistance in how to ftp your files to your server.

If you are having trouble getting the movie to play on the internet, these steps are recommended:

1. If you double-click on the .html file generated by KoolMoves on your local disk, does the movie play in your default web browser. If it does, the .html and .swf files are OK. If it doesn't, your browser may not have Adobe's Flash player. Go to the web site associated with your browser or to <http://www.adobe.com/> and download the player.
2. Upload the KoolMoves generated .html (as ASCII) and .swf (as binary) files to your web site. Does the movie play on your web site if you go to the KoolMoves generated web page? If not, there is a problem in how you sent the files to your web site. File > Export Movie > Diagnose Server Problems is a small expert system for diagnosing these types of problems.

## Export as SVG

This function exports all shapes or the selected shapes of the current key frame to an SVG file.



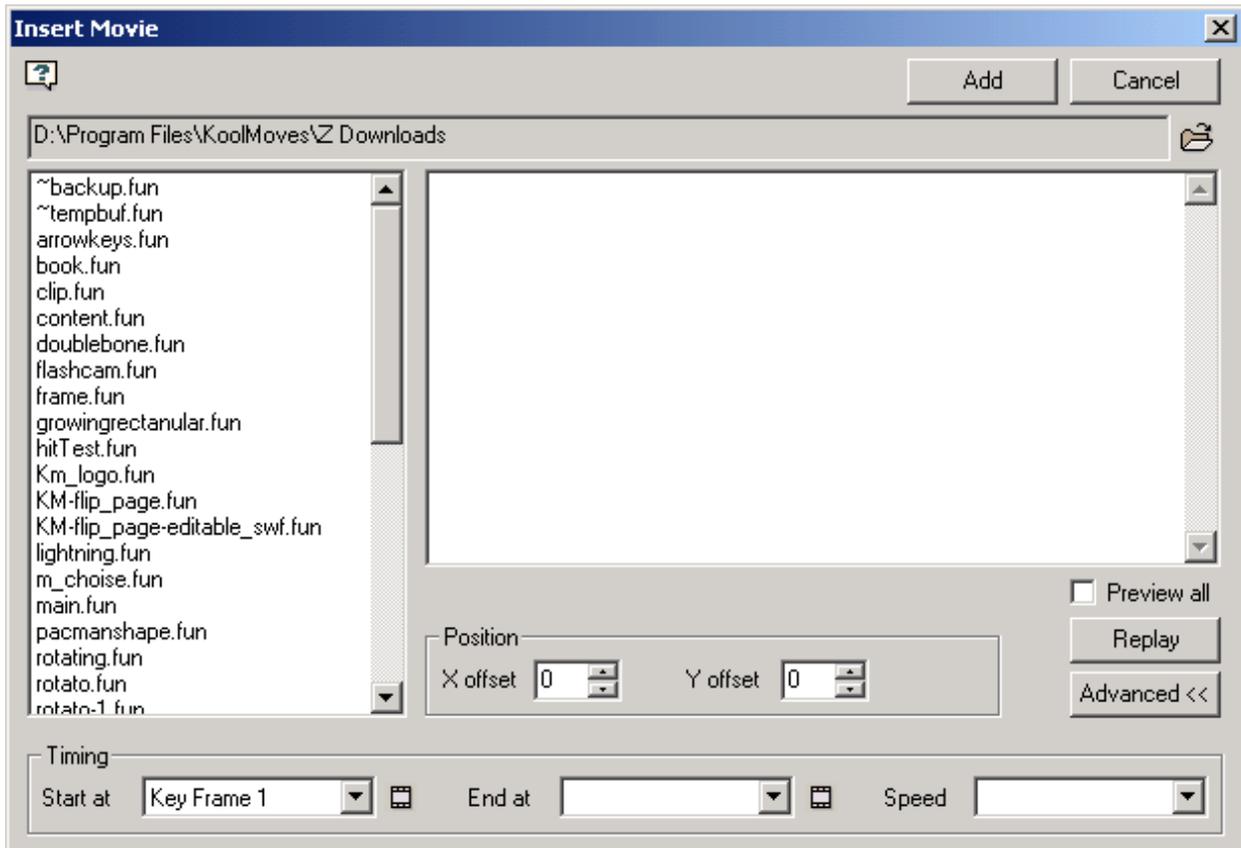
## Diagnose Server Problems

File > Export Movie > Diagnose Server Problems is a small expert system for diagnosing problems associated with exporting KoolMoves to SWF and HTML pages.

## **File > Insert KoolMoves Movie**

Independent animation projects can be developed and then imported into another project. On import, the destination project's speed setting applies so adjustments may be necessary; frame duration/tweens will be based on the destination projects default setting. If the inserted KoolMoves movie requires more frames than are available at its insertion point, KoolMoves will add additional frames to the host project. These animations can be inserted into movie clips.

Select from the names of .fun files in the left list box and press Add to add to your animation. To preview the entire movie, check Preview All. Otherwise, only the file to be inserted will be previewed.



X & Y Offset – Adjust the placement of the clip art relative to the center of the frame.

Timing, Start at – Select an insertion point for the animation – defaults to active frame.

Timing, End at – The process will add frames to the current project if required.

Timing, Speed – adjusts the number of tweens used by the animation, based on the default tween setting of the destination project.

## ***Working with Website Editing Tools***

The Crate Diseño Web has tutorials for the most common web editing tools:

FrontPage, [www.koolmoves.com/tutorials/kmwithfrontpage.pdf](http://www.koolmoves.com/tutorials/kmwithfrontpage.pdf)

CoolPage, [www.koolmoves.com/tutorials/kmwithcoolpage.pdf](http://www.koolmoves.com/tutorials/kmwithcoolpage.pdf)

NetObjects Fusion, [www.koolmoves.com/tutorials/kmwithfusion.pdf](http://www.koolmoves.com/tutorials/kmwithfusion.pdf)

## ***MIDI Sound Files***

The Flash player does not directly handle MIDI sound. You have to convert the MIDI file to WAV or MP3 or synch it using other Flash player methods and FS Commands.

There are several freeware MIDI conversion apps, for example,

[www.polyhedric.com/software/mn/](http://www.polyhedric.com/software/mn/).

[www.tips-tricks.com/sound.asp](http://www.tips-tricks.com/sound.asp) gives a tutorial on how to add MIDI sound to your HTML.

This is html code that will background midi sound to your web page:

```
<EMBED SRC="hittheroad.mid" AUTOSTART="True" HIDDEN=True" LOOP= "True">  
<BGSOUND SRC="hittheroad.mid" LOOP="infinite">
```



## Chapter 5 – Data Views

Data Views are a major component of the KoolMoves interface; they allow you to track, display and edit elements (visual elements, effects, sounds and ActionScript code) placed in the movie at authoring time as well as organize the overall structure of the movie. There are six Data Views, but they fall into three broad categories based on what they do and how they work: Shape Properties, Overviews, and Lists. Data Views are accessible through the Views menu as well as the Menu Toolbar.



Like the Tools Toolbar, Data Views can be dragged to anywhere on your screen. Data Views can also be “collapsed” to their title bar by clicking on the Collapse button located below the title bar, and resized by dragging the resizing marks in the lower right corner of the window.



Collapse Button.



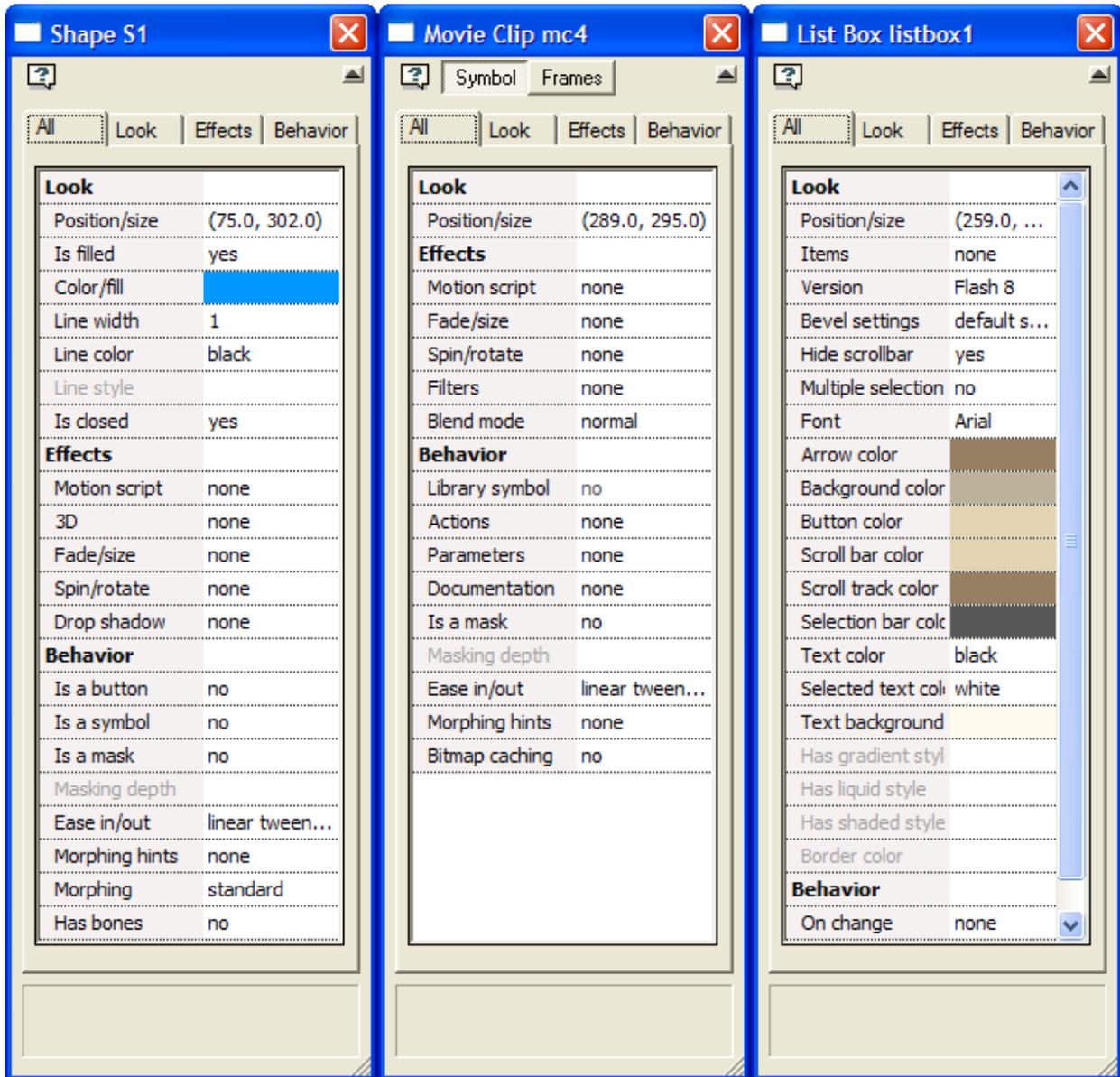
Resize Marks.

Working with visual objects and effects within a movie can require having more than one Data View open at a time. As these elements are addressed in their own chapters, Data View functionalities will be illustrated in greater detail. This chapter will provide a quick overview of the available Data Views, how they are accessed and used.

### 1) Properties Data View (Ctrl-F3)



Almost all properties for a visual element located on the Canvas are accessible from the Properties Data View. It is the most important of the data views. The content/format of the Properties Window is context sensitive; not only do different object types have different properties, but changing a property can change the content/format of the view.



Property views for 3 different types of objects.

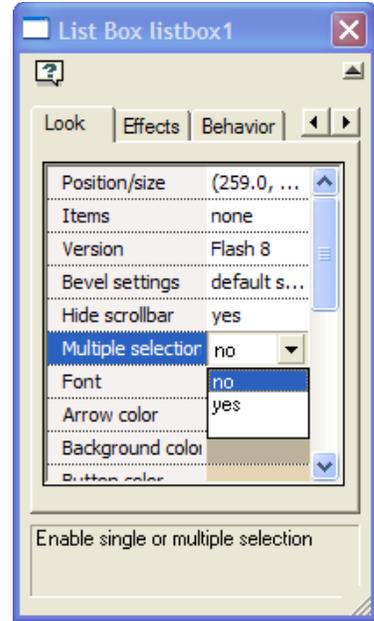
**NOTE:** Double clicking on an object will open the Properties window.

Click on a field within the Properties data view to highlight that field. A description of that field will appear at the bottom of the Properties data view. If the field is not grayed out, a drop down menu arrow or an ellipsis (“...”) should also appear next to the property’s current value. Clicking on an ellipsis will open a dialog window.

All visual elements on the Stage have a Position/size field. Clicking on the ellipsis will open the [Numeric Transform](#) dialog (covered in Chapter 12 – Transforms & Effects).

The Properties Data View is resizable - a scrollbar is displayed if all the fields are not visible, and left and right navigation buttons are displayed all the property type tabs are not visible,.

Click on a tab other than All to isolate related properties.



The Properties Data View fields for a typical shape are covered below; fields specific to Button, Text, Movie Clip and Component objects are covered in their respective chapters.

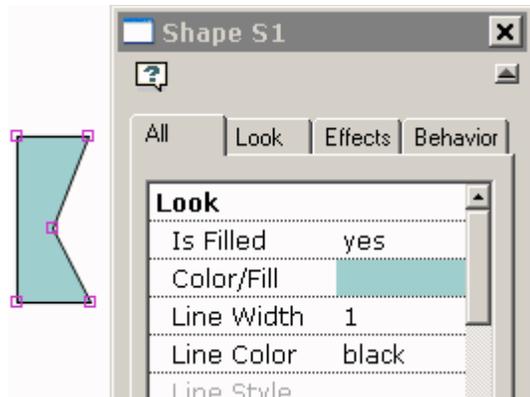
### Look (Appearances)

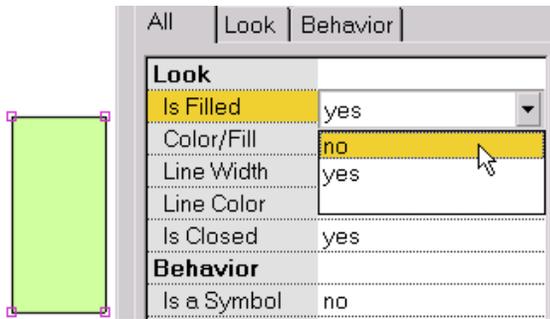
You can control the appearance of a shape with points through its:

- 1) Fill
- 2) Line Width
- 3) Line color
- 4) Last border segment



All conveniently located in the Shape Properties Window.

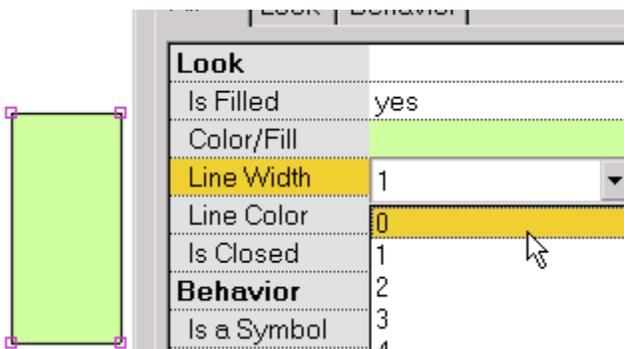
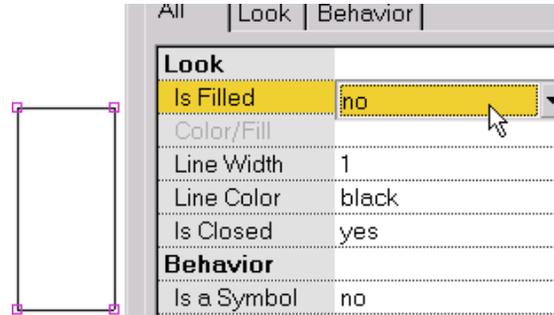




A shape can be filled with a solid color or a bitmap (Image Fill). To make the interior of the shape transparent, select Is Filled and choose No.

For more about Color/Fill options, see [Coloring/Transparency](#) in the next section.

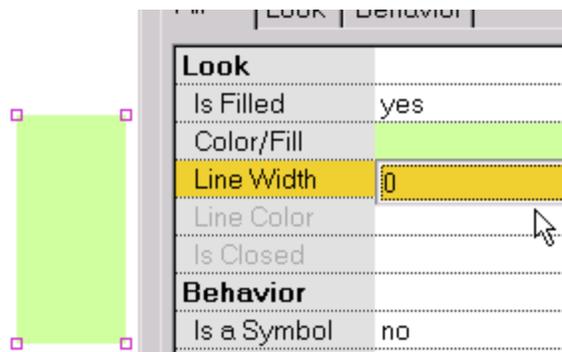
This shows a shape that is not filled.

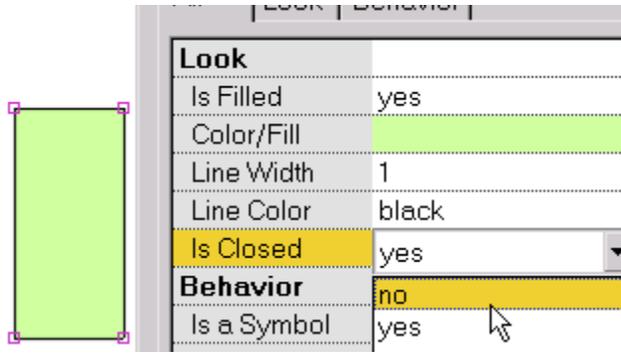


To change the Border Thickness around the shape, select Line Width. The line width is an integer number of pixels.

Borders with values of .25 and .5 or that have some degree of transparency are not rendered correctly in the KoolMoves internal player at this time, but are exported correctly to a Flash™ movie.

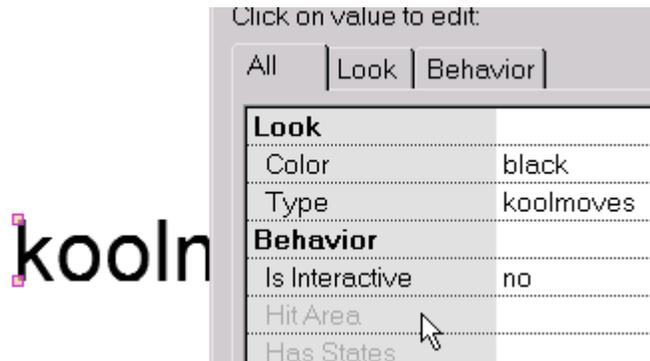
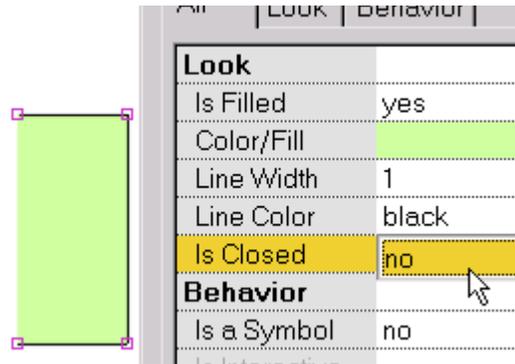
This shows a shape with no border (i.e., line width equal to 0).





To set whether the last border segment of the shape is drawn or not, select Closed.

This shows a shape with the last segment not drawn. When the shape is also not filled, we have a line (i.e., the Closed property is off and the Filled property is off). The Closed and Filled property states can be toggled by selecting them.



The property dialog for text looks different because text has different properties than shapes with points. The color of text can be changed by clicking on the Color property. A dialog that lets you set color will appear. The text, font, font size can be edited by clicking on the Type property. A dialog will appear in which you can edit the text.

Assigning button properties and making a non-text shape into a symbol are explained in the next section.

### Coloring / Transparency

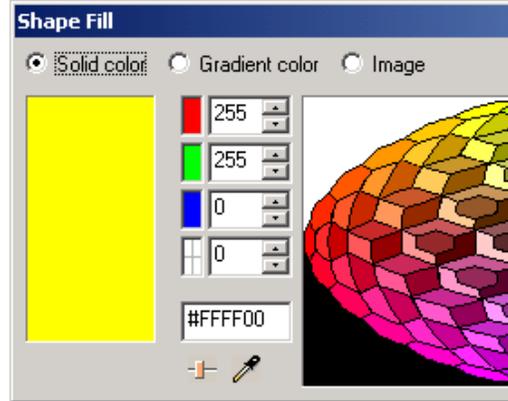


Shapes and text can be filled with a solid color (with or without transparency), a color gradient, or a tiled image (Image Fill). Access to the Shape Fill dialog is through the Properties Data View and the Shape Fill tool in the Tools Toolbar.

Solid color and gradient fills can have degrees of transparency. On the shape fill dialog box, there is an edit box for entering a transparency value. The range for transparency is 0 (opaque) to 100 (totally transparent) %.

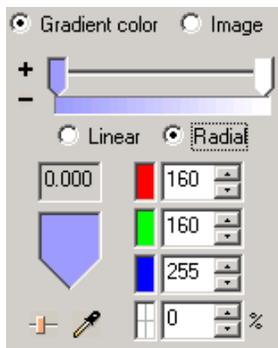
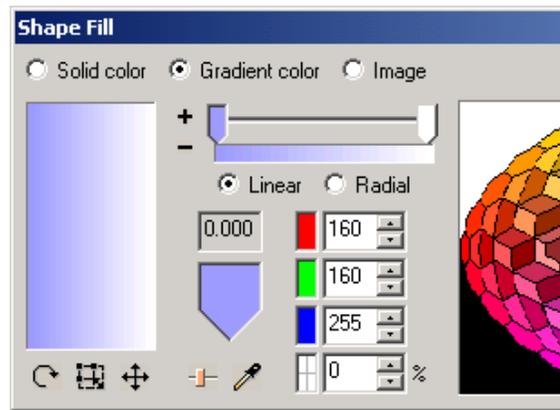
**To fill with a solid color:**

Select Solid Color at the top of the dialog. You can (1) enter red, green, and blue (all 0 to 255 in range) and a transparency value (0 to 100 in range), (2) select with your mouse from the color wheel (partially shown), (3) use red, green, blue, hue, saturation, and lightness sliders (press the button on the left under the transparency value), or (4) use an eyedropper (press the button on the right under the transparency value) to select a color from another shape on the Stage. To view the shape fill before committing, press Preview. On OK, the shape fill is applied to all selected shapes.



**To fill with a color gradient:**

Select Color Gradient at the top of the dialog box. A gradient fill is comprised of up to eight color pointers. The color associated with the selected color pointer is shown underneath the gradient definition bar. The color of the selected color pointer (it is the color pointer which is depressed) can be changed by (1) entering red, green, and blue values (all 0 to 255 in range) and a transparency value (0 to 100 in range), (2) selecting with your mouse from the color wheel (partially shown), (3) adjusting the red, green, blue, hue, saturation, and lightness sliders.

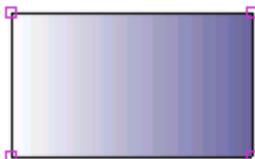


The color gradient can be either linear or radial. Color pointers can be dragged to the left or right along the gradient definition bar to change the gradient effect.

A color pointer can be added by pressing the small button with the plus sign. The mouse cursor turns into a color pointer with a + in the middle. Click at a position on the gradient definition bar. Then change the color of this color pointer.

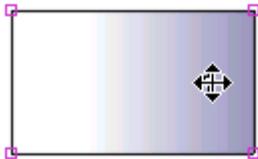
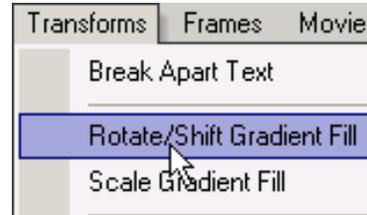


A color pointer can be deleted by pressing the small button with the minus sign and the selected color pointer is deleted.



An example of a selected rectangular shape filled with linear gradient is shown here. Only a shape with points can be filled with gradient color. To fill text with gradient color break it apart into a shape with points (Transforms/Break Apart Text).

The rotational orientation, position and size of the gradient fill can be adjusted with Rotate/Shift Gradient Fill and Scale Gradient Fill under the Transforms menu or with the applicable buttons in the Shape Fill Dialog.

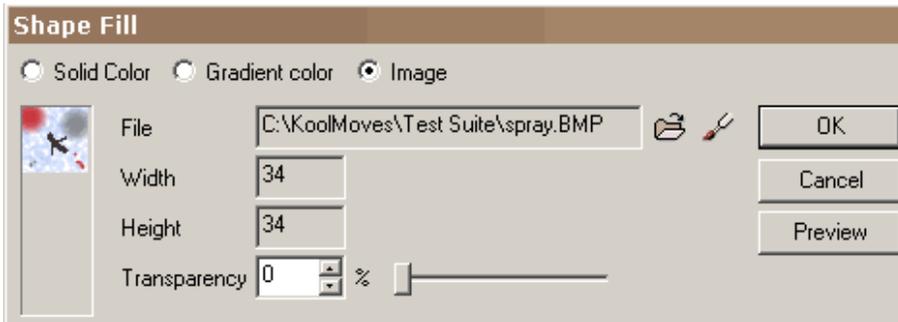


Moving the center of the gradient fill to the right shifts the gradient to the right.

Moving the cursor around the center rotates the gradient.



**To fill with an image:**



Select Image at the top of the dialog box. Use the Browse "..." button to locate your image file. Bmp, jpeg, tiff, pex, and png file formats can be imported. Transparency in png files is ignored at this time. If you use the same image with the same filename multiple times throughout the movie, KoolMoves is smart enough to store the image only once in the exported Flash™ movie. If the image is larger than the shape, only the upper left portion of image will fill the shape. If the image is smaller than the shape in any dimension, the image will tile to fill the shape. Only a shape with points can be filled with a image. The transparency of the image can be adjusted with a slider.

test

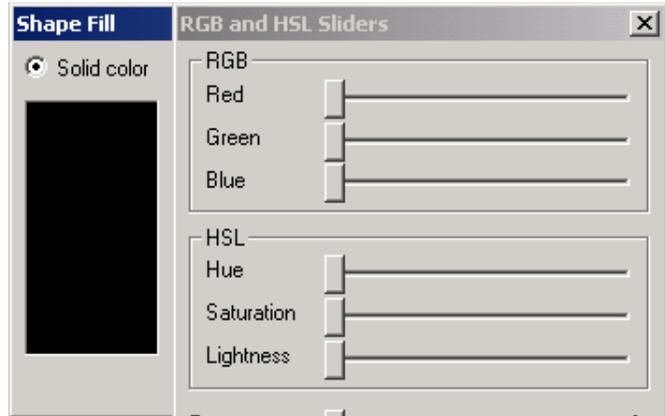
You can preview the fill by pressing Preview. On pressing OK, the fill, either color, gradient, or image, will be applied to all currently selected shapes and, if the fill is color, to all selected text.

**Color Models: RGB vs HSB**

There are many color models in use today; all of them based either on how color is perceived or how colors will be generated for a specific medium or purpose. Three of the most popular color models in use on computer applications are RGB (Red-Green-Blue), HSB (Hue-Saturation-Brightness) and CMYK (Cyan-Magenta-Yellow-Black). The CMYK model is used specifically for (four color) printing applications and thus does not apply to Flash applications.

Color monitors combine red, green and blue (the primary colors) pixels to generate the visible spectrum from black to white (this is why RGB is called an *additive* model). Each primary color can have a value ranging from 0 to 255 (00 to FF in hexadecimal notation – each color is two digits in the final hex color value 0xRRGGBB). 0x000000 is black while 0xFFFFFFFF is white; pure hues of red, green and blue are 0xFF0000, 0x00FF00 and 0x0000FF, respectively. There are 16,777,216 colors possible.

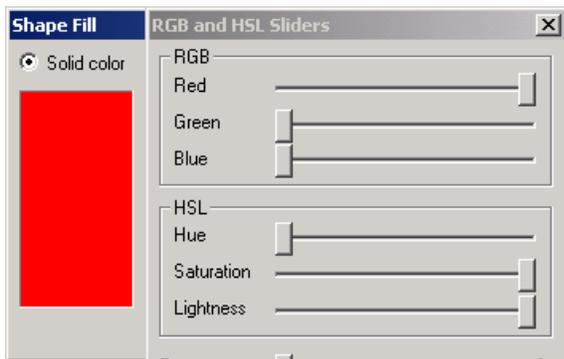
When color models other than RGB are used on personal computers, ultimately they have to be translated into RGB at some point to display on the monitor. Applications using CMYK often have a color calibration process to avoid color shift problems. Color shift is not an issue with HSB – any color that can be represented in HSB can be represented in RGB, and vice versa.



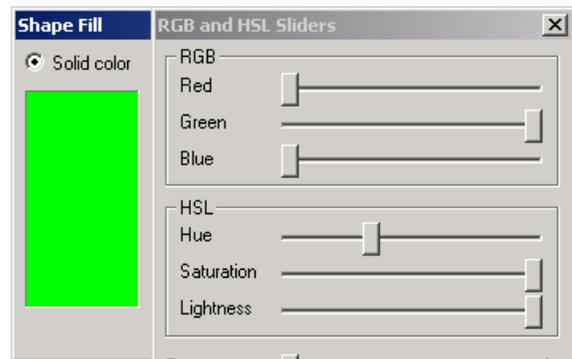
Slider Button

Black Slider Settings

The HSB color model is more in tune with how humans conceptualize colors, and using the HSB slider bars is conceptually analogous to how artists have traditionally mixed pigments to create colors. Hue refers to a color's light frequency, saturation is the amount or strength of the hue (its purity), and brightness is the amount of black in the color (its lightness or darkness). To oversimplify the HSB model, a color is a hue that is modified by saturation and brightness.

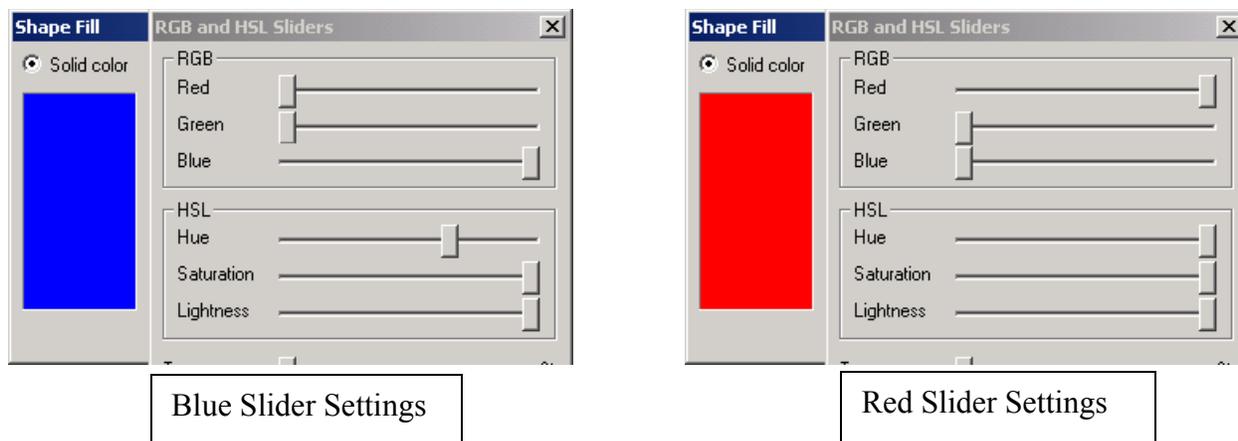


Red Slider Settings



Green Slider Settings

A hue is a "pure" color, i.e. one with no black or white in it. A shade is a "dark" color, one produced by mixing a hue with black, a tint is a "light" color, produced by mixing a hue with white, and a tone is a color produced by mixing a hue with a shade of grey. Hue values are graphically represented as a circle, typically range from 0 to 360, with 0 and 360 being identical (ie, hue cycles from red back to red).

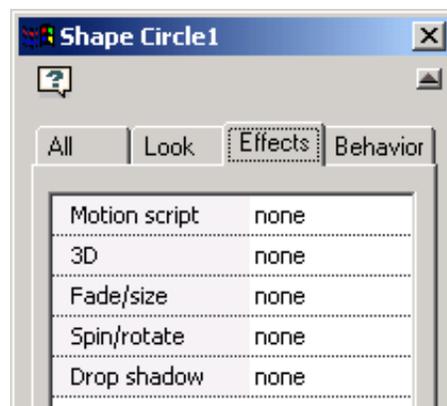


Saturation refers to the intensity of a specific hue. It is based on the color's purity; a highly saturated hue has a vivid, intense color, while a less saturated hue appears more muted and grey or "washed-out." With no saturation at all, the hue becomes a shade of grey. Saturation ranges from 0 to 100 percent.

Brightness, also known as "brilliance," "luminosity" or "lightness", is a relative expression of the intensity of the energy output of a visible light source. Brightness ranges from 0% to 100%. In the RGB color model, the amplitudes of red, green, and blue for a particular color can each range from 0 to 100 percent of full brilliance.

## Effects

There are about half a dozen powerful effects that can be applied to shapes, group of shapes, text, imported SWF movie, or movie clip (Fade In/Out, Expand, Shrink, Zoom In/Out). Not all of the effects work on each object type. Effects are found under the Effects menu, on the Effects tab of the Properties Data View, the Effects tab of the Score Data View, and the Effects tool in the Tools Toolbar. First select a shape and then select the effect. Most of the effects occur over several key frames; you can specify how many key frames when you create the effect or at a later time by using Effects > Modify or by editing the effect in the Properties dialog. You can also combine effects in many cases.



A special kind of effect called motion script uses sophisticated Flash 5 and higher action script code to create the effect. Each motion script is highly customizable, but usually can not be combined with other effects. See the [Motion Scripts](#) entry in Appendix E for a complete listing of available motion scripts.

The advantage of the regular built in effects like fade in and zoom in is that they can be applied to any object and they preserve any frame-by-frame user imposed color, shape, and position changes. The advantage of the ActionScript-based effects is that they are more unique and are highly parameterized.



Objects with Motion Script, Filter, or Blend Mode effects are displayed on the stage with a dashed blue bounding box. Other effects are displayed with a green dashed box.

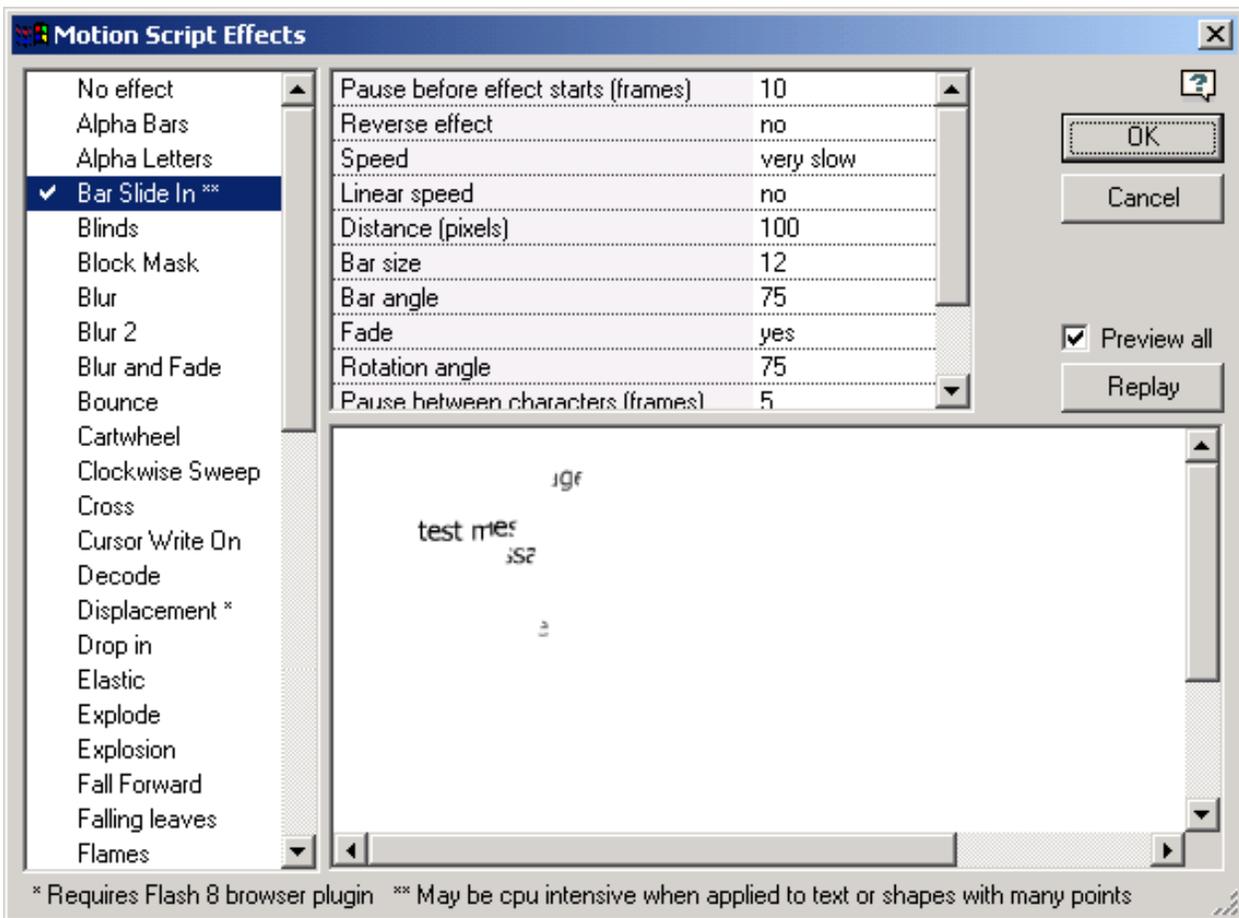


Additional ways to modify objects are available under the [Transforms menu](#) and [palette](#) covered in Chapter 12.

### Motion Scripts



Motion scripts are Flash 5 or higher ActionScript that can be applied to almost any object -- text, an image, a movie clip or a general shape. The motion script animated effects are not viewable within the KoolMoves main editing environment- all you see is a blue dashed box surrounding the shape. The motion script action is visible in the motion script preview window, in a web browser, or in a stand-alone Flash player.



\* Requires Flash 8 browser plugin \*\* May be cpu intensive when applied to text or shapes with many points

Because of the nature of these ActionScript-based effects, frame-by-frame changes in color, shape, and position are ignored by the effect. The position, color, and transparency from the first frame the shape appears in is used by the effect. The length of the effect is determined by the effect itself not by the number of frames the shape appears in. Also, motion scripts disable drop shadow effect applied to static text. If the text is a single letter or if the effect is applied to a non-text object, effect parameters such as left to right or right to left do not apply.

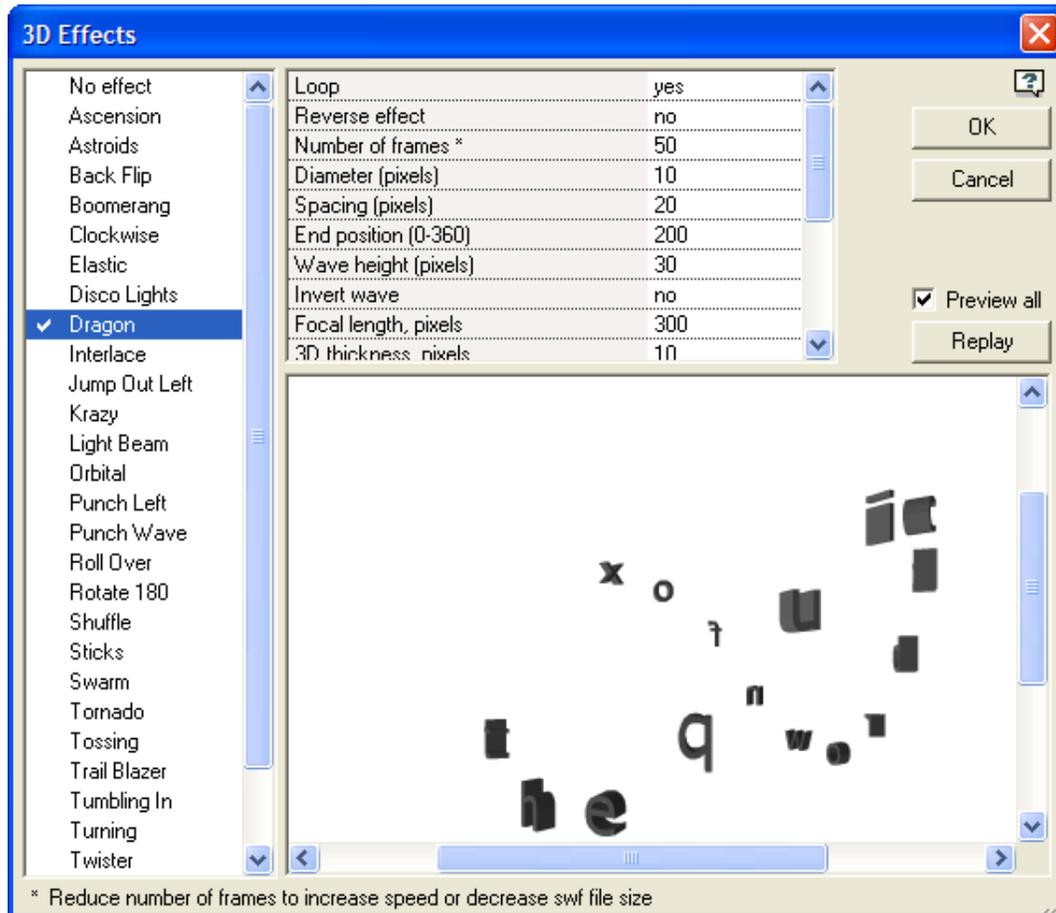
You can achieve some level of frame-by-frame changes by embedding the object with the motion script inside a movie clip and then manipulating the movie clip in position, size or rotational angle.

For those advanced users who want to control the effect from the main movie through ActionScript commands, the name of the movie clip that encompasses the effect has the same name as the shape in the KoolMoves shape list; this is a dynamically created movie clip (i.e., at movie execution) so you will not see it listed in the shapes list. If there is an embedded space, the space is removed. In the example above, code to find the x axis location of the movie clip containing the text would look like “`x_loc = sdf._x;`”.

You are able to create your own motion scripts and use them in your movies. See the `readme.txt` in the My Scripts folder inside the Motion Scripts folder for more information. See the [Motion Scripts](#) entry in Appendix E for a complete listing of available motion scripts.

### 3D

KoolMoves's 3D effects combine motion scripts with a 3D rendering engine. 3D effects can be applied to static text and shapes with solid (i.e., no gradient or image fill) colors. To remove an effect, select "No effect" in the list of effects. Checking Preview All displays the entire movie rather than just the selected object.



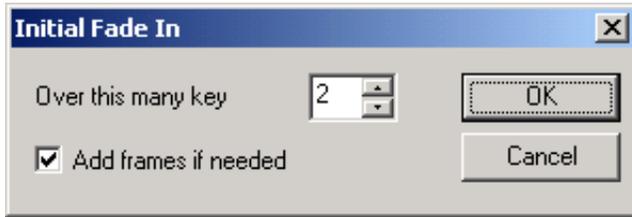
The 3D effect cannot be combined with other effects on the object, and only one 3D effect can be applied at a time. However, if the shape or static text is placed within a movieclip, additional effects can be applied to the movie clip. For example, while a 3D effect can not be combined directly with a drop shadow effect, a drop shadow filter can be applied to a movie clip containing the object with a 3D effect.

3D effects result in larger than usual SWF files. The effect, indicated by a blue dashed box, is not viewable in the KoolMoves editing screen or its internal player but can be viewed in the preview window associated with the 3D effect or in a web browser.

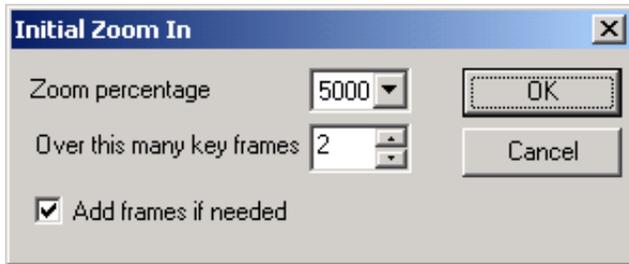
the quick brown fox

**Fade/Size**

 The Fade, Grow, and Zoom Transition Effects have their effect over an object’s beginning and ending key frames. Click on a check box to set parameters for the effect and turn the effect on.



The Fade and Grow settings are identical and work in a similar fashion: the Alpha value or size starts 0% and changes to 100% (“Initial”), or starts at 100% and changes to 0% (“Final”). The number of key frames, combined w/ the frame duration and movie speed, determining how fast the effect will occur.



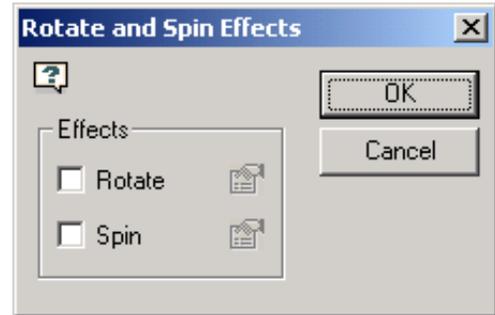
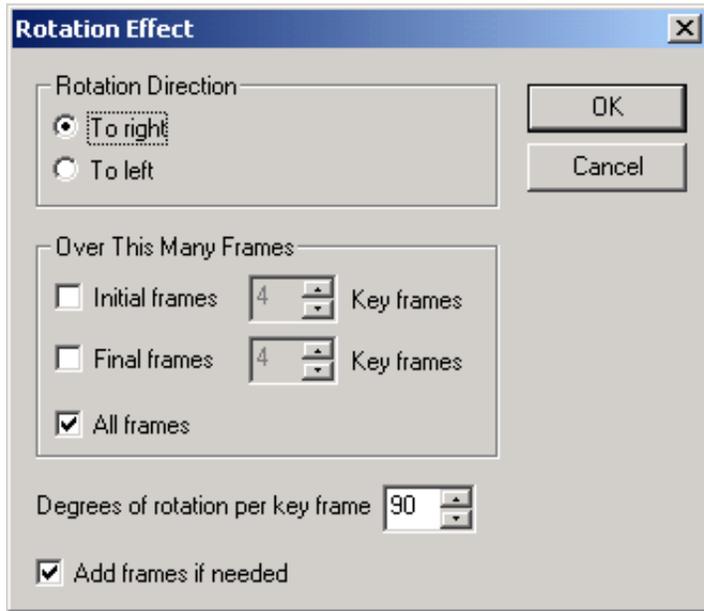
The Zoom effect does not automatically range from 0 to 100%, so an additional parameter exists in these dialogs.

Once the parameters are set, turn on the Effect by clicking on OK. Click on the edit button  in the Fade and Size Transitions window to change the parameters for the respective effect.

Zoom effects can not be used in conjunction with Fade or Grow effects.

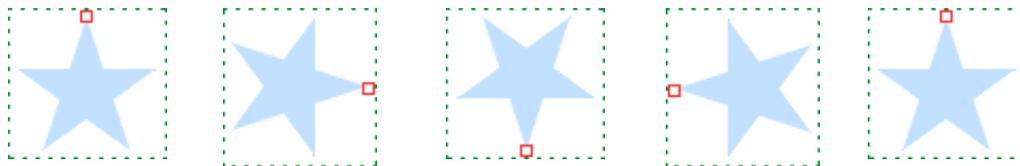
### Spin/Rotate

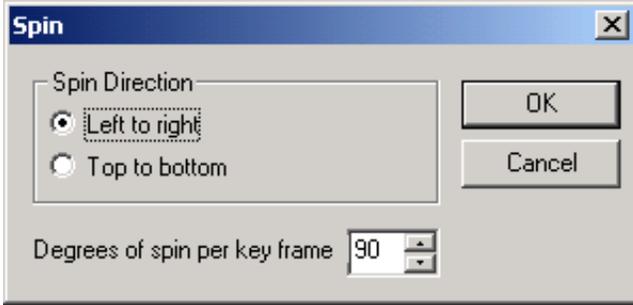
 The Rotate and Spin Effects dialog window is pretty standard – click on box to open parameters dialog, edit buttons disabled until effects are turned on, click on OK button to save effects to selected object.



Like the Fade/Size effects, the Rotation Effect allows you to apply the effect to either the initial or final key frames, or both. The Fade/Size default key frame effect duration value is 2 while it is 4 for the Rotation effect. The explanation for this difference is covered in the [Morphing](#) entry under the Behavior tab – suffice it to say that the Rotation Effect needs 4 key frames to work properly.

Below, with the help of some image editing, you can see the effect of a 90 degree Rotation Effect on a star shape over the first five key frames.





For the Spin/Rotate effects, the only valid values for the degrees of spin or rotation are between 1 and 180, inclusive.

Spinning differs from rotating in that at 90 degrees the shape is edge on to the display plane – at which point a two dimensional shape disappears. The shape below is spinning at 30 degrees per key frame.



### Filters

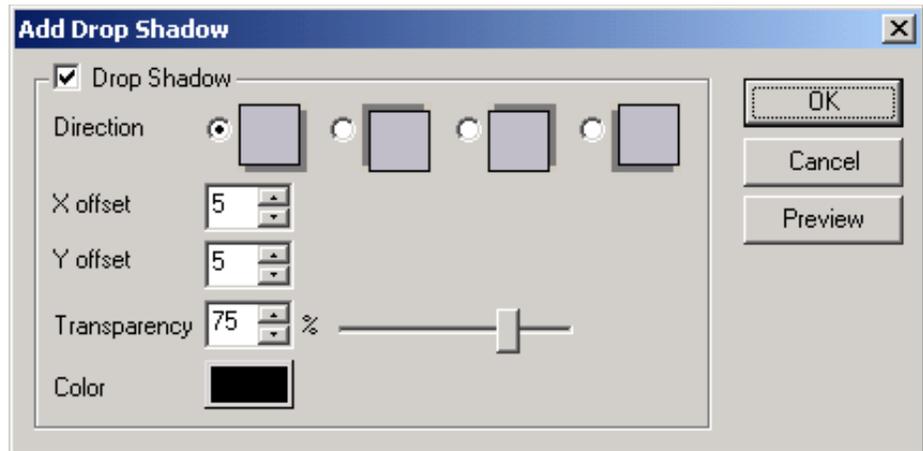
Filters can only be applied to movie clips, buttons and dynamic text. For information on how to use Filters, please see the [Movie Clip Effects](#) entry in Chapter 11.

### Blend Modes

Blend Modes can only be applied to movie clips and buttons. For information on how to use Blend Modes, please see the [Movie Clip Effects](#) entry in Chapter 11.

### Drop Shadow

 Drop shadows do not work well with Fade transitions and are disabled by motion scripts when applied to static text.



## Behavior - Shapes

The fields available under the Behavior tab will vary, depending on the type of object that is selected. Addressed below are the fields for a simple shape – most apply to buttons, movieclips and other object types. The bones property is discussed in Chapter 6 – Drawing/Editing Shapes.

### Is a Button

Set to yes to convert a shape to a Button.

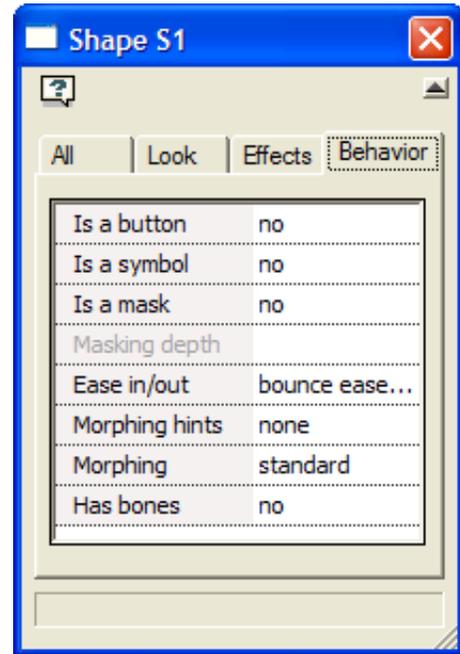
### Is a Symbol (as opposed to Shape)

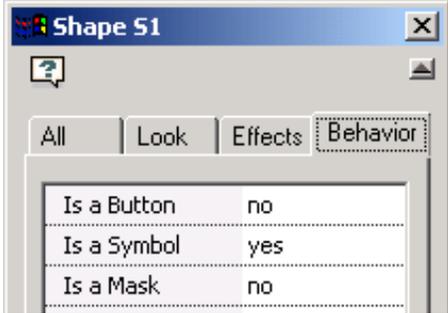
A symbol is a shape which is drawn once and reused more than once throughout the movie. One of the options in File > Preferences is to have objects drawn with the Tools Toolbar tools default to symbols instead of shapes. Drawing shapes as symbols results in the smallest swf file size but puts limitations on how shapes can be changed frame-to-frame. These limitations are acceptable for most application or game designers but would normally be unacceptable for cartoonists.

One of the principal methods for optimizing Flash™ movie performance is to make as many shapes as possible into symbols. Technically, a symbol is a shape that is shared by other shapes (called instances of the symbol). If you move a point in one instance it will be moved in all instances throughout the movie (KoolMoves first asks you if you want to change all instances). If you change the gradient in one instance, all instances will be changed. And, so on. The exceptions to this are solid color fills, shape position, scale and rotate transformations. These four only change the instance acted upon, not all instances. This is dictated by the Flash file format.

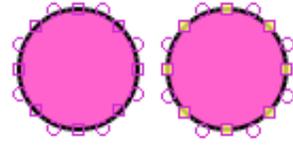
For simplicity, KoolMoves treats all text shapes as symbols.

If you have shapes that appear more than once in a single frame or in multiple frames, then it is advisable to make these shapes into symbols if you do not intend to move individual points from one frame to the next. In many cartooning situations, you would want to move points from one frame to the next to make a mouth move to simulate talking or to make a hand move to simulate waving, etc. so such shapes would not be good candidates for symbols. If the shape is stationary and undergoes no change throughout the movie, it is not necessary to make the shape into a symbol (KoolMoves will automatically handle this situation). In the future, we will be providing a method to optimally convert shapes into symbols after the animation is complete so it is not necessary to be thinking about this optimization while you are developing a movie.





To make a shape into a symbol, select Yes for the Is a Symbol property. This should be done immediately after creating the shape before a copy is made within a frame or a frame is copied. Notice that the points to the shape have a different appearance to indicate that these points are potentially shared with other shapes. The symbol instance is edited like any other shape. If the change affects other instances you will be



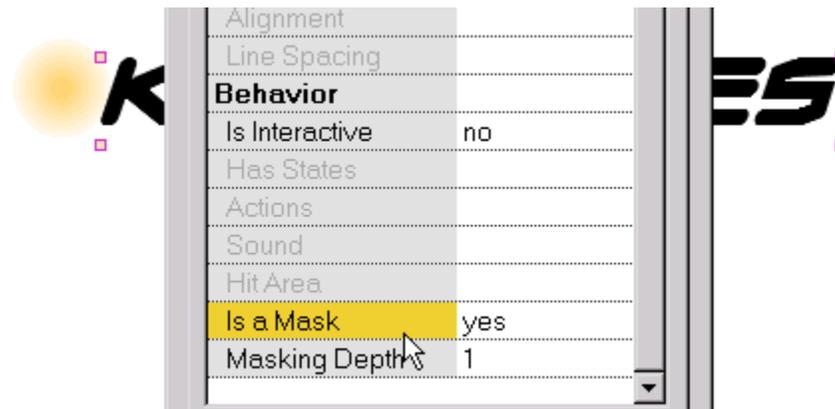
warned.

### Masking and Masking Depth

Masking is a very powerful technique. The following illustrates a simple animation example of letters being revealed from left to right using a gradient and a text mask.



We start with a circular shape, no border, that is filled with a gradient. On top of that we place a text shape. The text is the mask. Think of it as a cookie cutter. The mask cuts out what is underneath to the extent of its masking depth which in this case is 1. The fill or color of the mask is immaterial. The mask itself is invisible when the movie plays. In the editing environment, the mask is by default visible.



A shape is made into a mask in the property dialog. The depth of the masking effect is also assigned in the property dialog. Anything except buttons can be made into a mask.



In the first frame, the gradient filled shape is positioned to the left of the text. The frame is copied and the gradient filled shape is positioned to the right of the text. During animation, the gradient will sweep across the text revealing the letters.

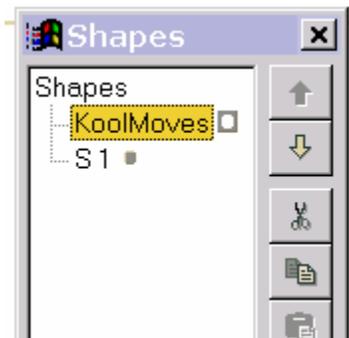
The logo for KoolMoves is displayed in a bold, italicized, black sans-serif font. A yellow-to-orange gradient mask is applied to the text, starting from the left and moving towards the right, partially obscuring the letters.

This is how it looks when the gradient filled shape is underneath the center of the text (in this case the movie spans three key frames).



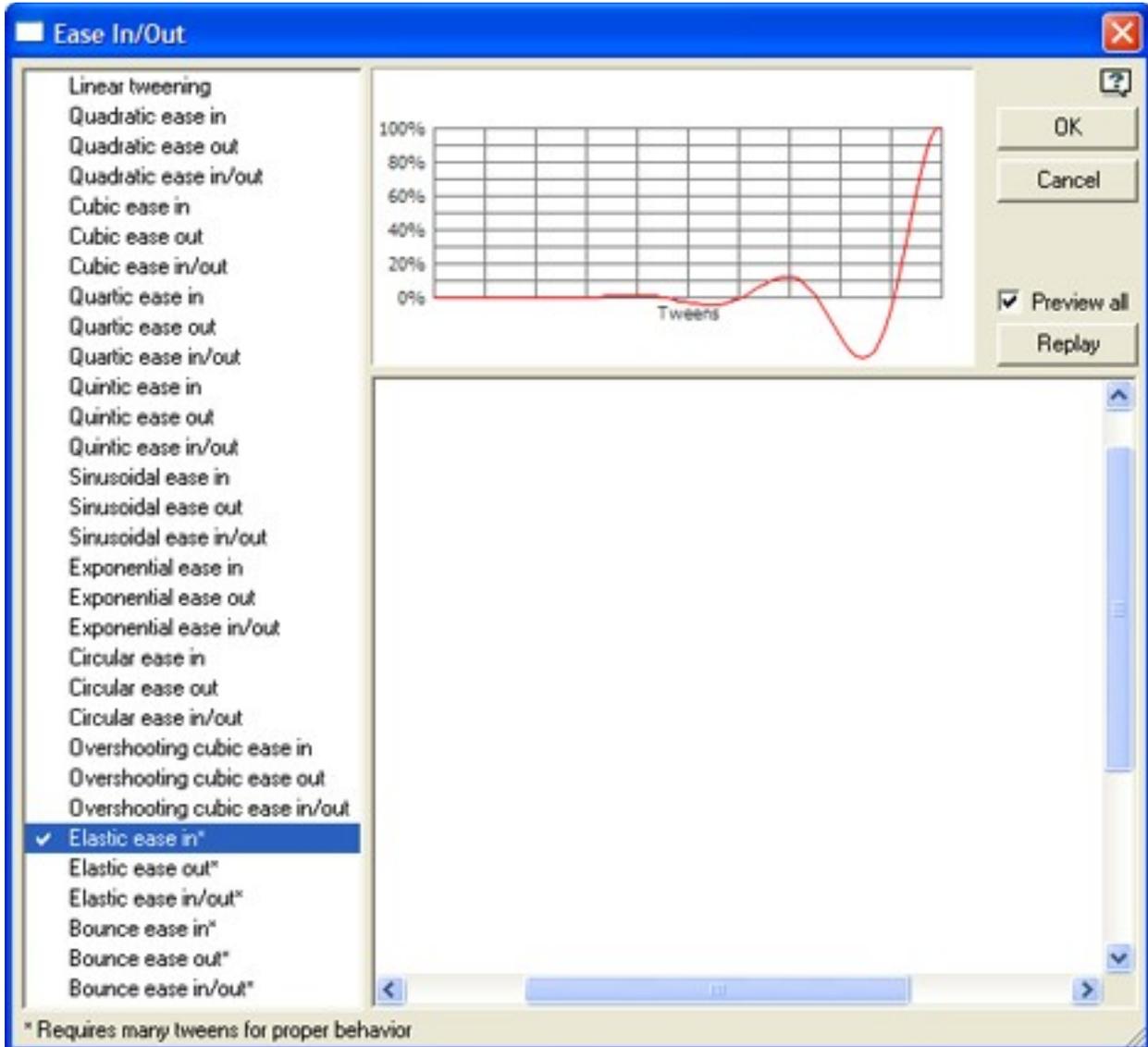
To see the effect of the masking, press Ctrl+M. This is how the movie will look when this key frame is played. Ctrl+M removes the masking effect and the mask is made visible again.

The shapes dialog identifies masking shapes and shapes which are clipped by masks with two different icons. The mask is always on top of the shapes which are clipped by the mask.



### Ease In/Out

The Ease In, Ease Out property is used to apply variable acceleration to an object’s tweening effect. Introduced with KoolMoves 5.6, there are thirty different scenarios in addition to the default linear tweening. Easing cannot be combined with a motion script effect.



To change the easing over many frames, use Selection Scope before changing the easing. Checking Preview All displays the entire movie rather than just the selected object for the current frame. The Elastic ease in acceleration chart above indicates that negative acceleration (reverse motion) occurs at a couple points during the transition.

## Morphing

Morphing is the technique by which shapes in one key frame are smoothly blended into corresponding shapes in the next key frame. Morphing is handled by KoolMoves without you having to think about it. There is morphing between two key frames when the number of tween frames between two key frames is greater than zero. To prevent morphing, set the tween frames in that part of movie to zero. To perform morphing between two key frames, KoolMoves must recognize the relationship between shapes in the two key frames. If a key frame at the end of the movie has shapes A and B and that key frame is copied to the end of the movie, shapes A' and B' in the copied key frame are related to A and B through the copy process so correct morphing will occur. Morphing occurs between a shape and its single counterpart in the next key frame. There cannot be morphing between a single shape and multiple shapes or vice versa. The Morphing Hint displayed in a Properties window is used to tweak performance when problems occur for morphing over multiple frames. .

### *Morphing hint*

Turn this feature on to help in the rendering of the morphing effect when there is a dissimilar number of points between the two shapes.

### *Shapes > Advanced > Morph Source*

The morph source function gives you the capability of morphing two shapes that are not currently morphing. This is particularly useful if you insert a movie into your current movie and you want to cause morphing between shapes in the current movie and those in the inserted movie.

A shape can have only one morph source. Make sure 'draw shapes as symbols' is turned off in File > Preferences > Drawing, or change the symbol to a shape in the Properties window. The morph source dialog asks you to select a shape from the previous key frame to assign as the morph source to a selected shape in the current key frame.

The key to good morphing is to have the same number and type of points in the source and target shapes. For example, when morphing between a circle and a square, add additional points to the square with the Add Point and Line to Curve tools (Chapter 6 – Reshaping). Every other point should be a spline (ie, curve point) when morphing to a circle.



### *Non-Linear Morphing*

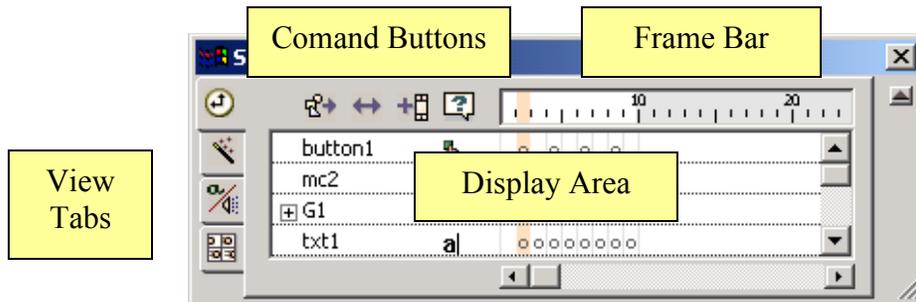
Standard morphing entails linear interpolation between two adjacent key frames. This is sufficient for most situations but fails to produce good results when there is rotational motion. Non-linear morphing involves a cubic fit between 4 key frames; this produces superior movement but does result in a larger swf file size. If the non-linear option is selected and there are less than 4 key frames of motion, standard morphing will automatically occur. Non-linear morphing is only available in the Cartooning interface style.

## 2) Score/Timeline Data View (Shift-F3)



A powerful view of the movie is obtained through the Score Data View. The Score window has four tabbed views -- Score, Effects, Actions and Sounds, and Storyboard.

Each view provides a different look at what is happening in the movie timeline on a key frame by key frame basis, but there are certain common elements.



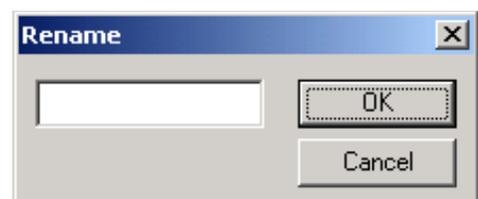
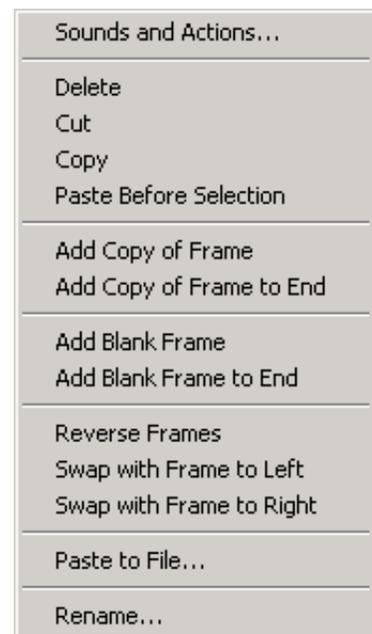
**Note:** The Flash Player usually skips the first frame, so your movie should start with an empty key frame. No object or ActionScript code should occur before the second key frame.

Click on a tab to move between views. Click within the Frame Bar to move to a different key frame. The pink column that extends from the Frame Bar through the Display Area indicates the active key frame. Right clicking on that pink column within the Frame Bar invokes a popup menu with a number of frame editing options.

Sounds and Actions opens the [Actions and Sounds Overview](#) covered later in this chapter. The Add ... Frame commands are discussed in [Editing and Playing Frames](#) in Chapter 1.

Delete, Cut, Copy and Paste Before Selection can work with multiple frames selected. Reverse Frames can only work when multiple frames are selected. To select more than one frame to delete, copy or reverse, click within the Frame Bar while holding down the shift key to add individual frames to the selection. The Swap commands deal with the frame immediately adjacent to the active frame.

Paste to File... will open a Save to... dialog window and paste key frames copied by the Copy command to a new .fun file. Rename... opens a dialog box to rename the active key frame.



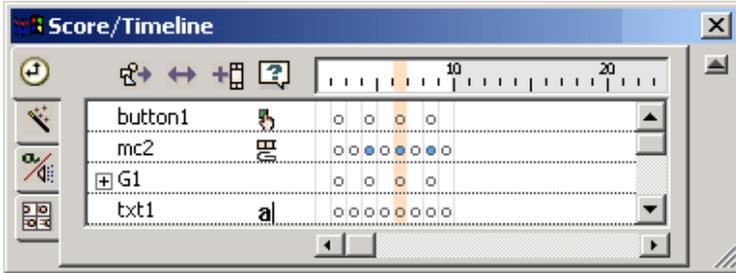
There are certain nuances to the Frame Bar, so let's take a closer look.



- 1) In the Key Frame bar there are white and grey regions. The white region reflects key frames that exist. Clicking in the grey region has no effect.
- 2) At the hash mark labeled 10 one would expect to be at the 10<sup>th</sup> hash mark, but it is actually the ninth. The numbering through the Key Frame bar is off by one. This is a legacy issue that doesn't affect performance, and since the numbering is accurate in the List of Files Data View and Navigation bar, no one has ever complained. Just an oddity of KoolMoves.
- 3) From the pink in the Key Frame bar we can tell that three key frames have been selected. However, which key frame is the active frame? The answer is “the last key frame selected”, also known as the one whose pink bar extends into the display area. In the image above, the last key frame selected was the fourth.

## Score tab

The Score view allows you to see at a glance where an object has been throughout the entire movie. The Score view is equivalent to the timeline view in other Flash tools.



In each key frame, objects that are present are displayed as dots. Clicking on a dot selects the object and moves the project to that frame. Selected objects are indicated by blue filled dots. Shift-click to make multiple selections

The order of the objects in the display area reflects their relative position (“depth”) in the content stack. Objects higher up are displayed over lower objects in the movie.

After an object is selected, right clicking in the display area invokes a context sensitive popup menu. With these commands the selected object(s) can be renamed, copied or moved to an adjacent key frame, or moved in the Content Stack. With KoolMoves 5.0.2, movie clips can be added to the Symbol Library (additional objects can be added starting with the next KoolMoves release).



The Score view also has the following command buttons (popup menus included where applicable):



Copy selected shape to the next frame.



Shift selected shape right or left across the timeline.

Shift Right  
Shift Left



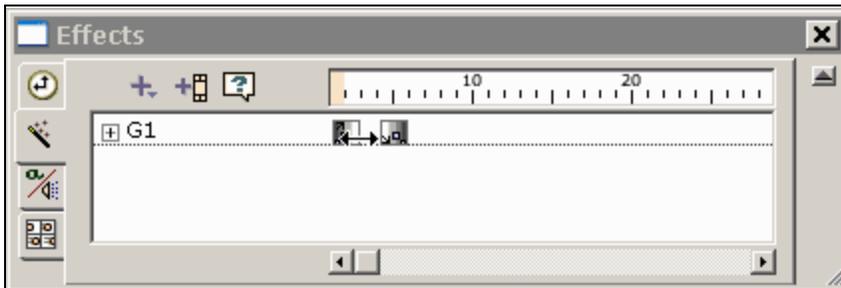
Lengthen the movie.

Add Blank Frames...  
Duplicate Last Frame...

## Effects tab

The Effects view is an efficient method for tracking effects used throughout the movie. This view also allows you to add and manipulate effects. An effect changes the size, position, transparency, and/or rotation of a shape, usually over a series of frames. Generally multiple effects can be applied. For instance a shape can fade in while increasing in size from nothing.

**Note:** Motion Scripts are the major exception referenced by the “usually” and “Generally” above. As a class they work in a single key frame, and can not be used in conjunction with effects that require multiple key frames or tweening.

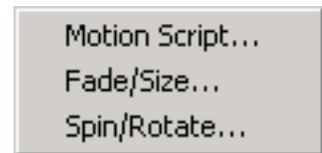


The span of the effect can be changed in the display area by dragging the side of the box identifying the effect.

The Effects view also has the following command buttons (popup menus included where applicable):



**Add Effect.** Opens a menu of effect types that can be applied to the selected shape. Available Motion Scripts are listed in Appendix E, the effects interfaces are discussed in the Effects Chapter.

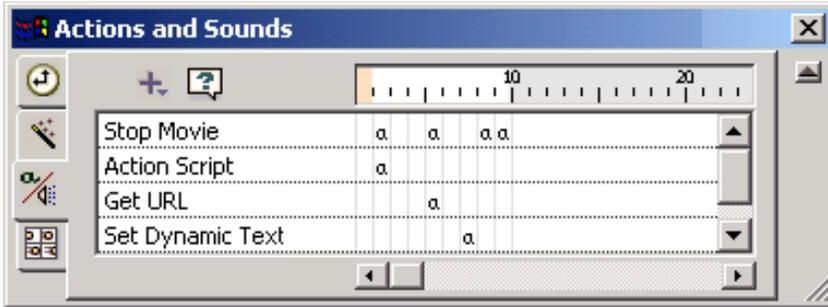


**Lengthen the Movie.** This is useful when you create unique transition effects.



## Actions and Sounds tab

With the Actions and Sounds view you can place either customized pre-configured commands or actual ActionScript code on a key frame. The preconfigured commands provide graphical user interfaces for the most commonly used programming functions. This This view, outside of the Toolbar and the Menubar, is the most often used screen element by the average KoolMoves user.



While pre-configured commands not as efficient as pure ActionScript, they allow users to create sophisticated movies without knowing ActionScript.

**Note:** the Flash Player will skip any actions on the first frame.

Right-clicking on a command in the timeline area of the display opens a popup menu. The Edit function is grayed out except for actions that required user inputs (i.e., configurable commands and ActionScript code).



**+** Add Action or Sound. Opens a menu containing 1) commands that can be applied directly to the selected frame, 2) interfaces to customize configurable commands, and 3) the Action Script Editor.



### Miscellaneous Notes:

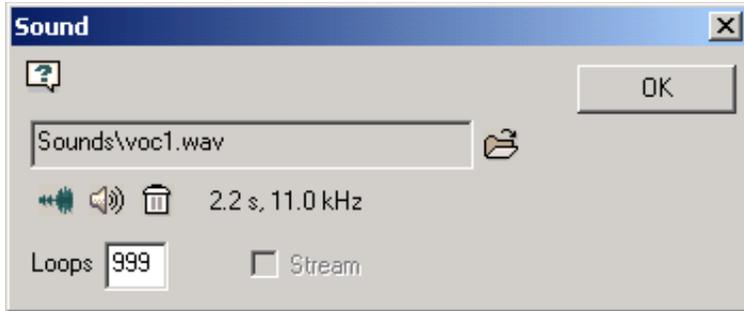
Play Movie, Stop Movie, and Stop Sounds are applied directly to the Frame and require no user inputs. Stop Sounds will stop all sounds currently playing when that Frame is loaded.

Sound files, movies, source text and other external objects defined at authoring time and loaded at movie execution can be quickly identified with the Dependencies window (Movie > Movie Dependencies).

The Dynamic Text commands are grayed out if there are no Dynamic Text objects in the project.

### Start Sound

Sounds (MP3 and .wav PCM) can be added to a specific frame in a KoolMoves movies. When the movie plays that frame the sound will start playing.



MP3 frame sounds can be streamed (this means that the animation can begin playing before the entire sound is loaded). The disadvantage to streaming sounds is that key frame animations can be dropped if they cannot keep up with the sound.



Select a Sound



Edit the sound using audio editor defined in File > Preferences



Play a sound



Remove a sound

Set Loops equal to 999 for continuous looping.

2.2 s, 11.0 kHz  
Sound file's duration and sample rate.

Each Key Frame can have only one sound action. Load sounds into movie clips to have multiple sounds on a Key Frame.

If you use the same sound file multiple times throughout the movie, it will be referenced that many times in the Flash™ movie but only defined once.

Sound files tend to be huge which can make the exported swf file huge. To reduce sound file size, take these steps: in a sound editing program convert from stereo to mono and to the smallest sample rate that gives you the quality you need. Shortening the sound track also helps.

Sounds can also be attached to button states (e.g. moving the mouse over a button or on pressing a button). See [Adding Sound to Button States](#) in Chapter 7 for more details.

Sounds do not play when the movie runs in KoolMoves' internal player.

If a sound plays longer than the movie and the movie loops, the sound from the first loop will continue playing during subsequent loops until it finishes. The sound effect worsens as the movie continues to loop. To solve this problem, add Stop Sound action to the last frame of the movie.

If your sound file is sampled at a rate other than 5500, 11025, 22050, or 44100, the sound will play in the Flash player at a rate nearest to one of the four rates listed above.

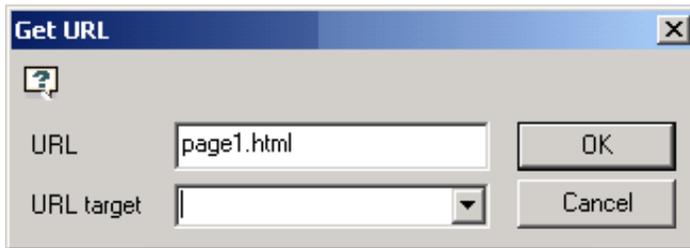
The duration of a sound is indicated in the Actions and Sounds View window and in the Actions and Sounds tab of the timeline by a sound icon followed by small squares.

### ActionScript

See Chapter 13 - [Action Scripting](#).

### Go To URL (URL Link)

The Get URL window is comparable to the `getURL` ActionScript global function. With this you can load a document into a browser or execute server-side scripts. A similar window is available for incorporating this capability into buttons.



The optional URL Target field (comparable to the window argument in `getURL()`) can be used to load the document into a named browser window or browser frame. If this field is left empty, the document is loaded into the current browser window or frame.

The dropdown list has four preset values: `blank`, `_parent`, `_self` and `_top`.

`_blank` – Opens the document in a new, anonymous browser window.

`_parent` – Replaces the frameset that contains the current movie.

`_self` – Loads the results of a script execution into the current browser frame or window.

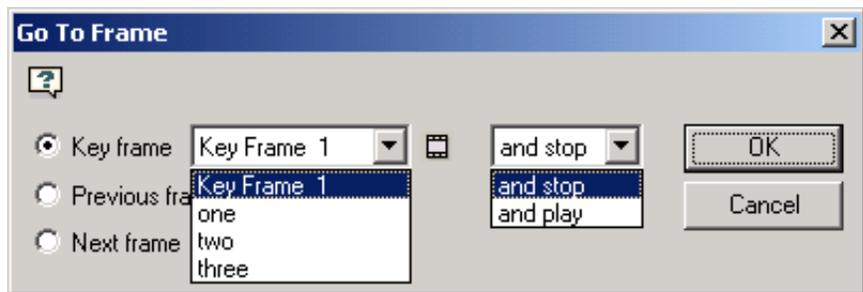
`_top` – Replaces all framesets in a web page with the specified document.

When testing a movie using Play in Browser, any html and swf files referenced by Get URL or Load Movie MUST be in the same directory as the fun file for these button actions to access these files, or set the directory for playing the movie in File > Preferences.

### Go to Key Frame

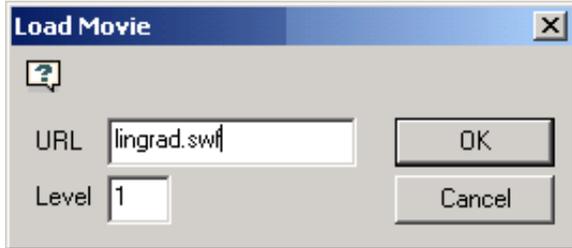
The Go To Frame window is comparable to the `gotoAndPlay()` and `gotoAndStop()` global function. A similar window is available for incorporating this capability into buttons.

With Key Frame Selected, select a key frame number (e.g., Key Frame 1) or a frame label from the drop down list. The Previous and Next Frame controls will direct program control to the most adjacent frame, including tweens.



## Load Movie

Load Movie loads a new SWF into the Flash player. This feature can be used to string together multiple SWF files. Because the project is broken up into multiple movies, the individual SWF files are smaller in size. Smaller files load faster, and manage memory more efficiently. A similar window is available for incorporating this capability into buttons.



Flash can play multiple timelines simultaneously. Each timeline is assigned a level. The main timeline is always Level0. Level0 sets the frame rate, background color and frame size for all the other loaded movies. Additional timelines load into higher numbered levels and are displayed in a stacking order with level0 being the bottom-most

level. When using the Load Movie action, a level must be specified to define where the SWF will be loaded. If loading one movie only, any level 1 or higher will work fine.

For use on the web, the URL for the loaded movie can be an absolute or relative URL, but, for testing purposes, all the movies must be in the same directory without directory or disk drive specifications in their names. When testing a movie using Play in Browser, any HTML and SWF files, referenced by Get URL or Load Movie actions, MUST be in the same directory as the fun file for these button actions to access these files or set the directory for playing the movie in File > Preferences.

A new movie can also be loaded into level0. This will replace the original contents of the main timeline, and is an effective method of stringing multiple SWF files together in a continuously running show.

An alternative method is to have a control movie in level0 that regulates the loading of additional movies/media into other levels.

Using relative paths with load movie can be confusing. Since any timeline can perform a load movie action, what path is the movie being loaded relative to? Is it relative to the main timeline at `_level0`? Or is it relative to the timeline that performed the Load Movie action? The answer is simple: Loaded Movies are always relative to the timeline that loaded them.

When a movie is loaded into a level, it is positioned relative to the upper left corner of the movie that loaded it.

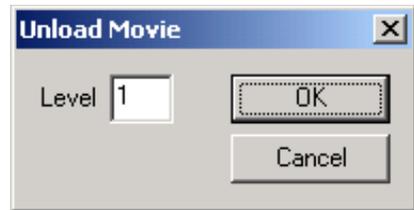
Loading a movie into a level other than `_level0` is much like adding elements to the stage; they will be in a stack. Unless higher level movies have backgrounds of same or greater size, the lower movie(s) (or portions of it) may be visible underneath the higher level movie.

### Daisy Chaining Movies

Use Load Movie to Daisy Chain movies together. Create a base movie that loads in the other media and then loops - make sure that you do not loop the load movie action. Load your main movie into Level 2 and your background animation into Level 1. Adjust your frame durations so your movie in Level 2 runs to the end, then add a new frame and load the next movie into Level 2 (first movie is automatically unloaded). Repeat as necessary for each component movie, then create an additional frame and loop back to Frame 2 (Frame 1 was left blank due to Flash Players tendency to skip the first frame).

### Unload Movie

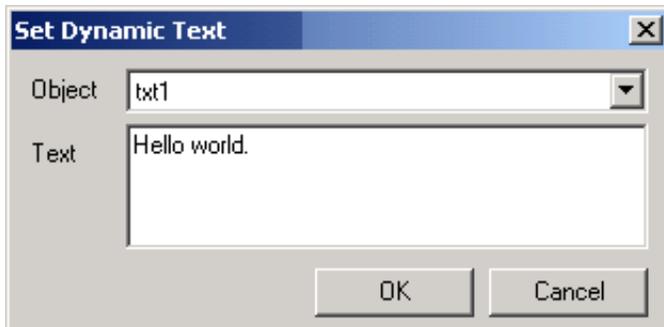
This window, comparable to the unloadMovie() global function, empties the specified level.



### Play Movie, Stop Movie, Stop Sounds

These commands are comparable to the play(), stop(), and stopAllSounds() global functions. No window opens for these commands because no parameters need to be set. Instead, the commands are immediately placed on the timeline.

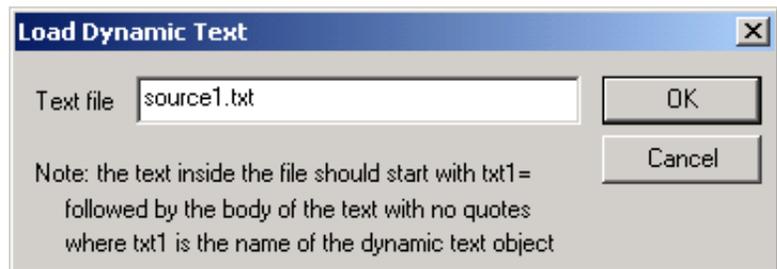
### Set Dynamic Text



This window allows you to set the text in a dynamic text object without knowing ActionScript. The image below is comparable to 'txt1.text = "Hello world.";', in Flash Player 6 and above ActionScript.

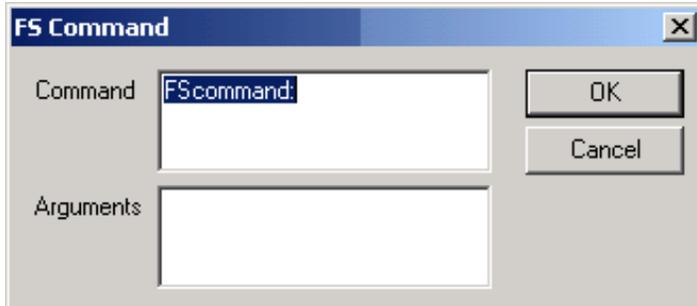
### Load Dynamic Text

Load Dynamic Text loads text into a dynamic text object from a .txt file.



## FS Command

The FS Command action provides a general method for Flash to send a message to whatever program is hosting the Flash player. This is advanced functionality, not for the average user.



FS Commands are not very reliable as they don't work on many browsers. You are actually better passing the info from the SWF to the browser using JavaScript. Likewise, you are better off using ActionScript for interactivity. There are a very select few instances where this is not the case.

In a web browser, FS Command calls the FS Command JavaScript function. In Director, FS Command sends a message interpreted as events by Lingo. In Authorware, FS Command sends commands to the scripting environment. In Visual Basic and Visual C++, FS Command sends a VB Event with two strings.

The FS Command was developed in Flash for 2 reasons.

1. To allow SWFs to talk to Live Connect and Active X.
2. When it was implemented IE did not support JavaScript: Commands as URLs. At that time FS Commands were the only way to pass information to IE 3.0 or lower.

The early versions of Flash, Flash 3 or lower, had a very limited scripting environment, to see just how limited download a shareware trial of Flash 3. If you wanted to script something you had to use FS Commands and Flash Player Methods to do it. Since the authors of the Macromedia Technotes wrote that the way to communicate with Flash was through FS Commands all of the following Flash book authors stuck to that story, which is untrue. Most FS Command tutorials on the web confuse FS Commands with Flash Player Methods.

FS Commands pass a variable and an argument to the native scripting language. With the exception of projector files, which have some limited commands built in, you must script the commands yourself. You need to be aware that there is no Active X support for IE for Mac, Netscape, Mozilla, and most open sourced browsers. Also Linux browsers are dropping Live Connect. Therefore the use of FS Commands for web page usage is problematic.

The most common use for a FS Command is in the use of projectors. These commands are quite easily called by using the following command words in the FS Command box and the following arguments in the argument box. It is important that you start the command with "FS Command:".

Command Arguments:

- Quit - Shuts down the Player.
- FullScreen True/ False - Makes your projector Full Screen if set to true and makes it regular if set to false.

- AllowScale True/False - Allows your Graphics to scale with the projector or not.
- ShowMenu True/False - Shows the Projector Menu or eliminates it.
- Exec “Path of EXE you want launched” - Launches any EXE if you are working on a MAC it launches and apple script. You must know the Path of the program. Save- Only available in the Flash 5 Standalone Player and may be removed at anytime without notice. It saves all of your variables to a txt file in the root directory.
- Save “File Name” – Saves to file name.

Many projector enhancement programs like Flash Tools, SWF Studio, Swish Canvas, Jugglor, Flashants FMProjector have custom FS Commands built into them. Be sure to read the documentation carefully before implementing one of these specialized FS Commands. Outside of browser non-support for FS Commands, there is one good reason to use them to communicate with your HTML. This one good reason is that FS Commands do not cause your web browser to click when the command is issued like JavaScript commands do.

In order to use FS Commands you must have this on your page:

```
<script=;"JavaScript;"> function navmovie_DoFSCommand(command, args) {  
  
if ( command == "init" ) {  
  
var buttons = parent.InternetExplorer ? window.navmovie : window.document.navmovie; if  
(buttons != null ) {  
  
buttons.GotoFrame(parent.targetFrame);  
  
// This function ensures that the navigation bar movie is set to the proper  
// keyframe when it initially loads. The argument init is the name you use  
// as the command parameter for the action attached to the first keyframe // in the navigation bar  
movie.  
} } // Hook for Internet Explorer  
if (navigator.appName &&  
navigator.appName.indexOf("Microsoft") != -1 &&  
navigator.userAgent.indexOf("Windows") != -1 &&  
navigator.userAgent.indexOf("Windows 3.1") == -1) {  
document.write('<SCRIPT LANGUAGE=VBScript>');  
document.write('on error resume next ');  
document.write('Sub fscommand_FSCommand(ByVal command, ByVal args) ');  
document.write(' call fscommand_DoFSCommand(command, args) ');  
document.write('end sub ');  
document.write('</SCRIPT> ');  
}  
//--></SCRIPT>  
</SCRIPT>
```

Notice that you have to have some VB on your page to catch the FS Commands from Active X. Now this is important. A FS Command is only a variable and a value when sent to a program. You must build the proper handler for your command and arguments with JavaScript.

Okay now the final use which is probably the most useful use of FS Commands is for in custom programs that use Flash as the interface such as Authorware and VB projects.

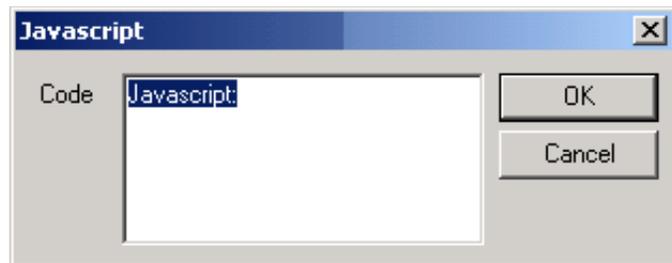
This is handled by the Active X Flash method:

FSCCommand(command, args)

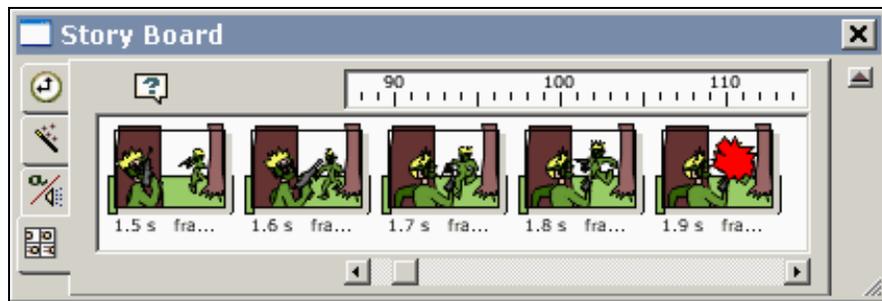
Generated when a FS Command action is performed in the movie with a URL and the URL starts with FS Command :. Use this to create a response to a frame or button action in the Flash movie.

### JavaScript

This window passes JavaScript commands directly to the web browser.



### Story Board tab



The Storyboard view displays a thumbnail and timing information for each frame on the main timeline. See the [Capture Movie Frame](#) entry in Chapter 4 to learn how to create screen captures that can be formatted in a

word processor and printed out

### 3) Movie Overview (Alt-F3)



The Movie Overview provides a detailed overview of all frames, shapes in frames, actions and sounds, buttons, movie clips and ActionScript code in the movie. If an object/visual element has strange behavior, this Data View is a good way to track down all actions and ActionScript in a KoolMoves project. Toggle settings are saved between sessions.



Open the Properties window for the selected shape.



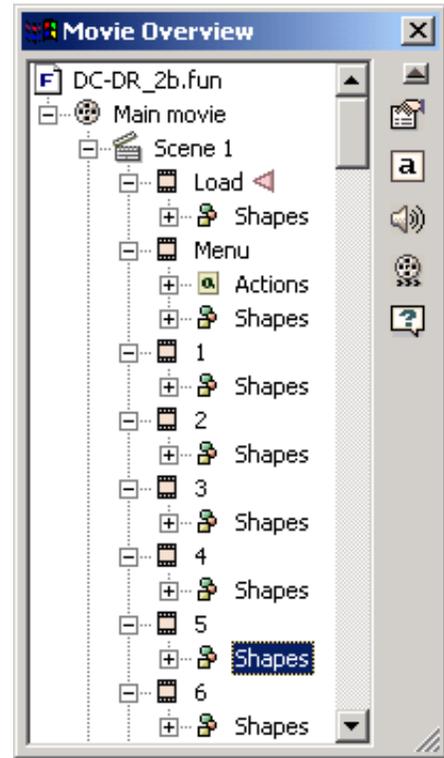
Toggle – display all frames in the movie and expand/collapse all Sound objects in their frame



Toggle – display all frames in the movie and expand/collapse all ActionScript entries in their frame



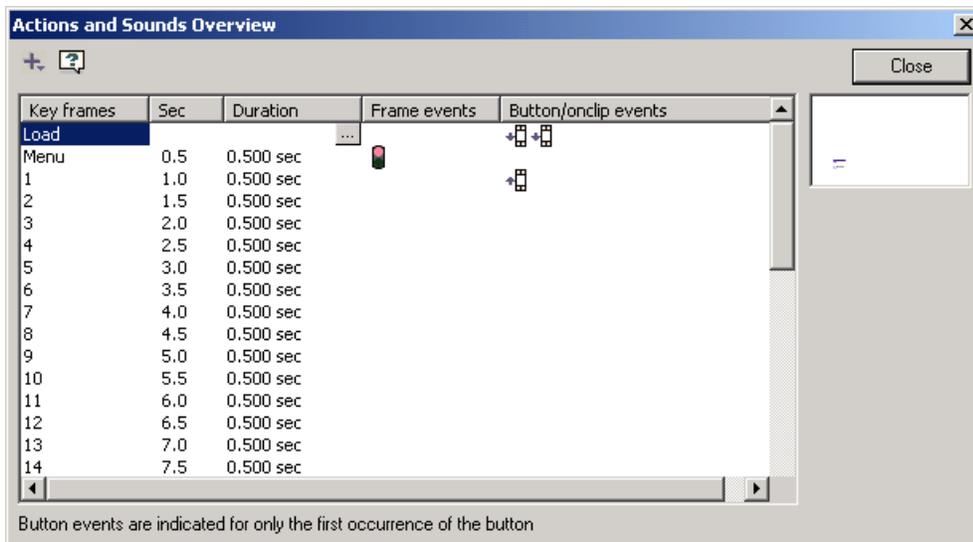
Toggle – display all frames in the movie and display all movie clips in the movie below the frames, at the same outline level of the Main Movie expand/collapse all Movieclips in the Movie



### 4) Actions and Sounds Data View (Ctrl-E)



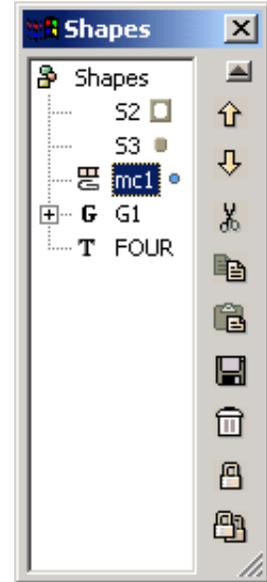
The Actions and Sounds Data View has frame actions displayed in a vertical table format. Left click on the Duration ellipsis (“...”) to open the Duration/Tween dialog, right click on the Frame and Button events to open the Actions and Sounds dialog.



## 5) List of Shapes (F9)



The List of Shapes window displays all visual elements on the active Key Frame. The order of the elements indicates their relative position in the content stack (i.e., their depth). Click on an object's name to select it; click a second time to edit the name. Use the Shift/Ctrl keys while clicking names to select multiple objects.



Paste places the cut or copied object at the top of the stack.



Object is a mask



Object is masked



Object has been selected



Move selected object Up/Down



Lock/unlock selected object/all objects. Locked objects can not be moved, edited, have effects applied to them or have their properties changed.



Cut selected object.

Objects from the last cut operation are stored in a clipboard. The clipboard contents are listed in the Clipboard folder in the Movie Overview Data View.



Delete selected object



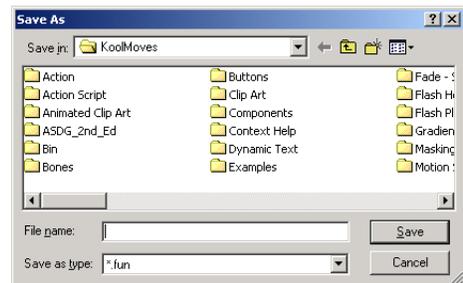
Copy selected object



Paste copied object



Paste copied object to new .fun file.



To lock a selected shape from changes in point position or many other changes, click on the lock button in the shape list popup. To unlock, click again. To lock or unlock all shapes, click on the lock all button.

## 6) List of Frames (Ctrl-K)



The List of Frames can be used for navigating, renaming key frames, and frame editing (cut, copy, paste, and reversing the order of frames). Frame selecting, renaming and same looking buttons are analogous to actions covered in the [List of Shapes](#) entry.



Paste places the cut or copied frames before the currently selected frame.



Reverse order of selected key frames

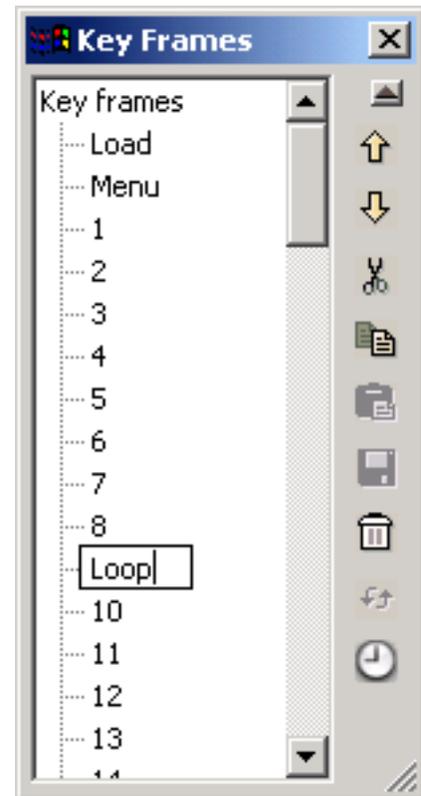


Set Frame Duration (seconds or tweens) for selected key frame(s)

Interesting things to know:

- KoolMoves allows you to copy and paste the Frame Actions associated with a Frame.
- Key frames without labels are automatically renumbered when their relative position is changed.

The second bullet point has a serious implication – name all key frames referenced by action script code to avoid having to recode references to that key frame each time renumbering occurs.



## Other Views

### List of Scenes (Shift-F2)

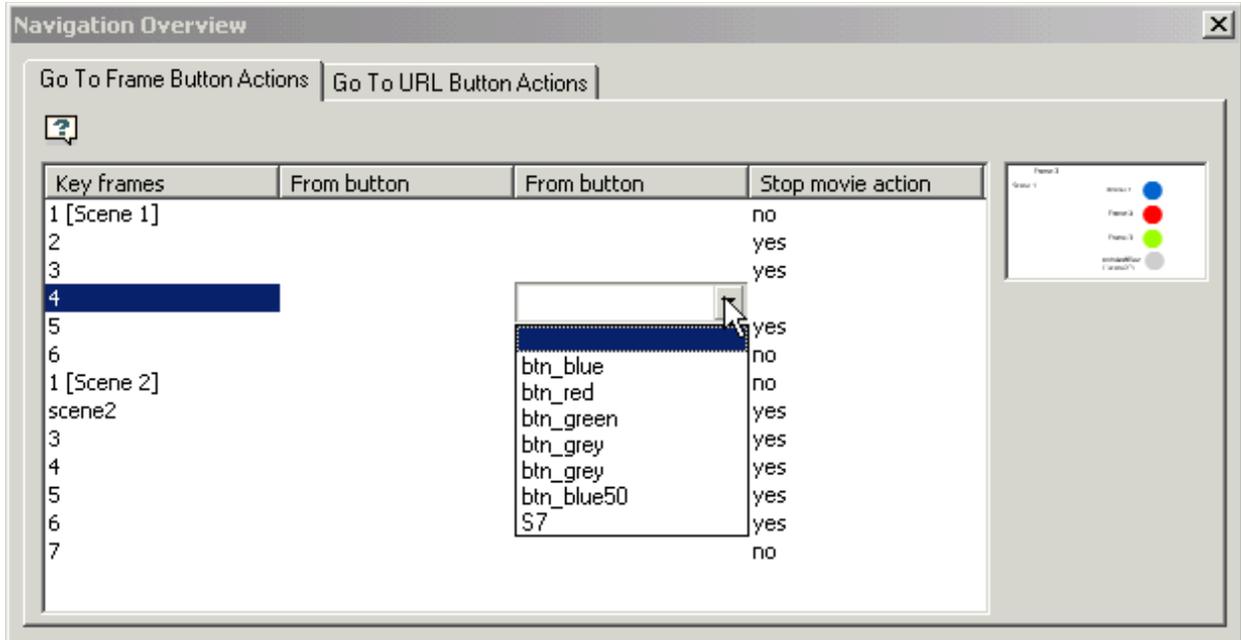
See [Scenes](#) in Chapter 1.

### Navigation Overview

Views > Navigation Overview...

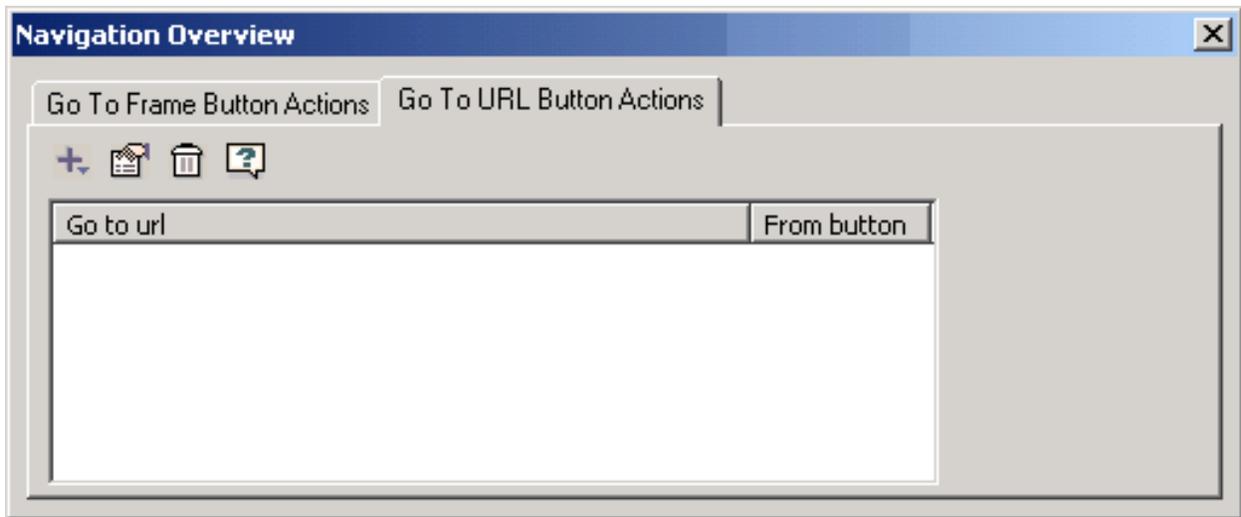
### Go to Frame Overview

This page shows all go-to-frame actions originating from buttons. You can create, edit or remove go-to-frame actions and stop movie actions by making changes in the second through fourth columns.



### Go to URL Overview

This page shows all go-to-url actions originating from buttons. You can create, edit or remove go-to-url actions by using the tool icons at the top of the page or the button selector in the second column.



## **Part III – Tools Toolbar (Toolbox) & Menu Commands**

**Chapter 6 – Drawing / Editing Shapes**

**Chapter 7 – Buttons**

**Chapter 8 – Toolbox Libraries**

**Chapter 9 – Importing Images**

**Chapter 10 – Text**

**Chapter 11 – Movie Clips**

**Chapter 12 – Transforms & Effects**



## Chapter 6 – Drawing/Editing Shapes

Before getting into the drawing tools, let's cover some items you will be using a lot when you start drawing.

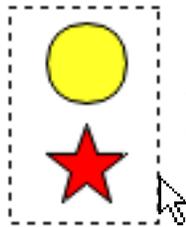
### Selecting Shapes and Points



Shape and point selection tools are located on the top part of the Tools Toolbar. These tools are used a lot. In addition to providing selection capability, they also allow you to move shapes and points. You can move previously selected shapes or points by dragging the mouse with the left mouse button depressed.



One way to select a shape is by clicking on the shape when the shape selection tool is active. If you click on a shape with the shift key depressed, you will add to any previous selections.

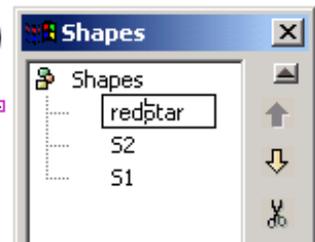
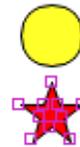


Another way to select one or more shapes is by dragging the mouse across the shapes totally enclosing them. If you drag the mouse with the shift key depressed, you will add to any previous selections.

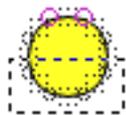


list.

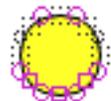
A third way to select one or more shapes is by invoking the list of shapes (under Views) and selecting from that list. Shapes are automatically named as they are created but you can change the name by editing the name in the



A point can be selected by clicking on the point when the point selection tool is active. The selection area associated with a point can be set under Points. If you click on a point with the shift key depressed, you will add to any previous selections.



Another way to select one or more points is by dragging the mouse across the points totally enclosing them. The points can be associated with different shapes. If you drag the mouse with the shift key depressed, you will add to any previous selections.



It is important to appreciate that point and shape selection is very general. You can select multiple shapes or multiple points. The multiple points can even be associated with different shapes. Selected shapes and points can be moved by dragging the mouse or by pressing the arrow keys.

## Shape Menu Commands

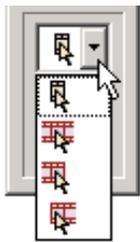
Shape menu commands are used to 1) select shapes, 2) position the selected shape(s) on the stage or in the content stack, and 3) control the scope of changes applied to the selected shape(s). Use the select shape tool to select shapes. 

### Select/Move (Shapes > Select/Move)

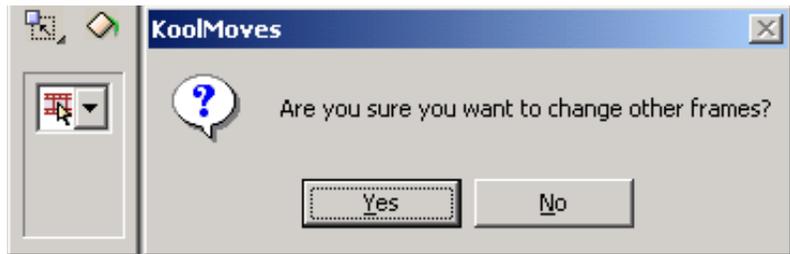
Select/Move applies to shapes, individual shape points, and other visual elements (like text boxes) and can apply over many frames. Selected shapes and points can be moved with the arrow keys. There are four types of shape and point selection (Selection Scope):

- (1) The current frame only (the most common).
- (2) The entire movie.
- (3) All frames from the current frame to the first frame.
- (4) All frames from the current frame to the last frame.

### Selection Scope (Shapes > Selection Scope)



The ability to select over multiple frames is very powerful. For example, if you change the color of a shape in the current frame and the shape has been selected over multiple frames, the color change will be applied to each of the selections. To prevent inadvertent sweeping changes, you are first asked if you want to change the other frames. The ability to change selected shapes and points over multiple frames based on what you are changing in the current frame applies to most of the functions in KoolMoves -- transformations, shape fills, translation, etc. There is an analogous command under the Points menu.



Sometimes you want to exclude some points from possible selection. An easy way to do this is to select the shape associated with the points you want to select and then select the point selection tool. Only the points associated with the previously selected shape will be visible. To hide shapes or points you can use Shapes > Hide Selection or Points > Hide Selection, respectively. Selecting Shapes > Show Shapes or Points > Show Shapes brings the shapes or points back. Shapes > Invert or Points > Invert inverts the current shape or point selection.

### Invert Selection (Shapes > Invert Selection, Ctrl-I)

Inverts the objects selected on the stage.

### Hide Selection

A toggle command that hides/shows the selected objects. No history is stored, so it only toggles once. There is an analogous command under the Points menu.

### Last Selection

Selects the most recent previous selection. No history is stored, so it only goes back one step.

## **Align**

The align commands are used to arrange selected shapes on a common base line. The Left, Right, Top and Bottom baselines are based on the extreme value of the selected shapes, while the Horizontal and Vertical Center baselines are based on an average of the shapes.

### **Left (Ctrl-Shift-L)**

The x-axis value for the selected shapes are all set to the value of the leftmost shape.

### **Right (Ctrl-Shift-R)**

The x-axis value for the selected shapes are all set to the value of the rightmost shape.

### **Top (Ctrl-Shift-T)**

The y-axis value for the selected shapes are all set to the value of the shape with the lowest y-axis value.

### **Bottom (Ctrl-Shift-B)**

The y-axis value for the selected shapes are all set to the value of the shape with the highest y-axis value.

### **Horizontal Center**

The x-axis value for the selected shapes are all set to the average value of the shapes.

### **Vertical Center**

The y-axis value for the selected shapes are all set to the average value of the shapes.

## **Change Depth**

The depth of the selected shape(s) can be changed with the following commands. The depth of a shape can also be changed in the List of Shapes data view. Depths of named objects can also be changed dynamically.

### **Bring Forward (Ctrl+Up)**

Moves shape up one.

### **Send Backward (Ctrl+Down)**

Moves shape down one.

### **Bring to Front (Ctrl+Shift+Up)**

Moves shape to top of content stack.

### **Send to Back (Ctrl+Shift+Down)**

Moves shape to bottom of content stack.

### **Bring Forward Selection... (Alt+Up)**

Lists all shapes above the selected shape and let you select where to position the selected shape.

### **Send Backward Selection... (Alt+Down)**

Lists all shapes below the selected shape and let you select where to position the selected shape.

### **Group**

Combines the selected shapes into a meta-shape, allowing the shapes to be treated as a single shape.

### **Add to Group**

The selected shape is added to an existing group of shapes.

### **Ungroup**

A group is broken up into its component shapes.

### **Select All**

All objects on the stage are selected.

### **Unselect All**

All objects on the stage are unselected.

### **Points Menu Commands**

Points menu commands are used to 1) select points on the selected shape, 2) add, delete, or move points on the selected shape, and 3) control the scope of points changes applied to the selected shape(s). After a shape has been selected, use the point selection tool to select points, then click and drag to move.



### **Commands covered under the Shape Menu**

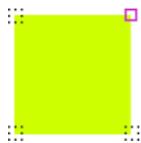
- [Select/Move / Selection Scope](#)
- [Hide Selection](#)

### **Add**

Adding points allows you to smooth a shape, add curves or corners or otherwise modify an existing shape or line. After selecting this tool, click on the outline to add a point.

### **Delete**

Deleting points allows you to simplify a shape, remove a curve or corner or otherwise modify an existing shape or line. After a shape has been selected, use the point selection tool to select the points to delete.

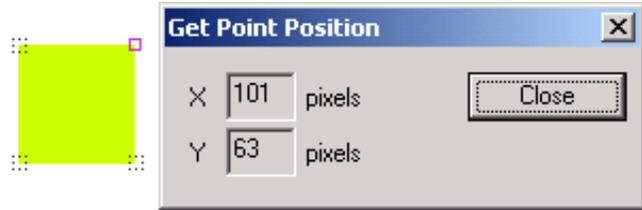


### Append to Line

Allows you to add new line segments to the end of the selected line.

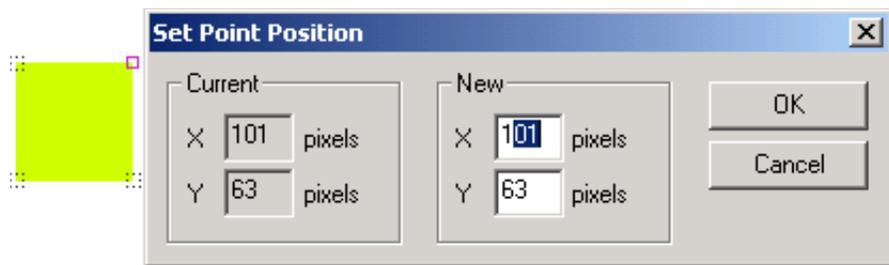
### Get Position

Returns the position of the selected point.



### Set Position

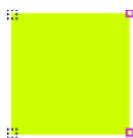
Allows you to change the position of the selected point.



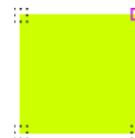
### Selection Area

You have three choices in setting the size of the point location indicators.

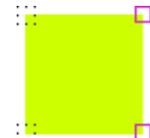
- Small -
- Medium -
- Big -



Small



Medium

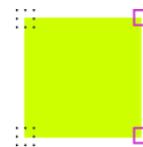


Big

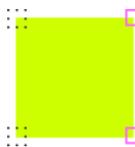
### Color Selected Points

There are three choices in setting the color of the selected point location indicators.

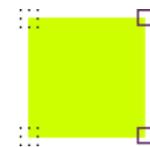
- Default -
- Light -
- Dark -



Default



Light

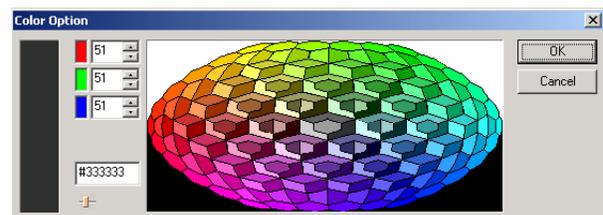


Dark

### Color Unselected Points

Use the color option window to change the unselected points color.

- Default -
- Custom -



## **Edit Menu Commands**

### **Undo/Redo Buttons/History**



The Undo and Redo buttons in the Menubar will undo or redo one action at a time. To view a list of actions, click on Undo History or Redo

History under the Edit menu to display a list of prior/subsequent actions. You can revert to any previous state or move back to a later state by selecting from the appropriate list.

Undo	Ctrl+Z
Redo	Ctrl+Y
Undo History...	
Redo History...	

### **Cut Shapes (Ctrl-X)**

Selected objects are deleted from the stage and stored in the clipboard.

### **Copy Shapes (Ctrl-C)**

Selected objects are stored in the clipboard.

### **Delete (Del)**

Selected objects are deleted from the stage.

### **Paste Shapes**

Objects stored in the clipboard are pasted to the stage.

### **Paste Shapes to Beginning**

Objects stored in the clipboard are pasted to the stage on the current frame and all preceding frames.

### **Paste Shapes to End**

Objects stored in the clipboard are pasted to the stage on the current frame and all succeeding frames.

### **Paste Shapes to File...**

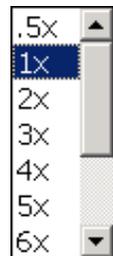
Objects stored in the clipboard are stored to a .fun file.

## **Miscellaneous Commands**

### **Magnification**



In the Menubar, Select a magnification level from the drop down list of 0.5 to 10. Changing the magnification level causes the Canvas to center on the selected visual element(s); when the magnification is set to 1, the Stage position is reset to the last position set by the Reposition Movie Frame command.



### **Trace Image**

File > Trace Image > Display Image – open and display an image (.bmp format) in the background of a frame for the purposes of tracing. The bitmap is centered on the screen. Use the drawing tools to duplicate the lines of the image. When you are finished tracing, clear the image with File > Trace Image > Clear Image.

### **Select Same Color**

File > Select Same Color - selects additional shapes with the same colors as the currently selected shape(s).

## Drawing Tools



You select a drawing tool by clicking its icon in the Toolbox. A small triangle at the lower right of a tool icon indicates hidden tools. Positioning the pointer over a tool displays a tool tip. To select a tool, click its icon. If the icon has a small triangle at its lower right corner, click on the triangle or hold down the mouse button to view the hidden tools; then, click the tool you want to select.

When a drawing tool is active the bottom of the Toolbox displays three options: fill color, border or line color, and border or line thickness. These options are present with any of the drawing tools and once set will remain until changed. Changing these options does not affect existing shapes already drawn.



There is a rich selection of drawing tools in KoolMoves: freehand, primitive or standard shapes, and shapes/lines drawn point-by-point.



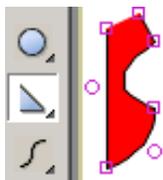
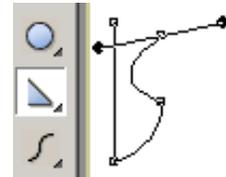
The freehand drawing tool allows you to draw like you would using a pencil. The first and last points are automatically connected when you release the mouse. If you don't want them to be connected, after the shape is drawn change the Closed property of the shape. The points generated by freehand drawing are curve fit to achieve a modest number of points. You can add or delete points later if needed.

The tools for drawing primitive shapes make it easy for you to draw circles, squares, rectangles, ellipses, stars and polygons. You click the left mouse button at one corner of the shape you want to draw and drag the mouse to the opposing corner of the shape. Release the mouse and the shape is drawn.



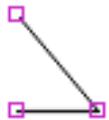
Shapes drawn point-by-point allow you to control the placement and number of points. You click the left button to define each point and continue until you have completed the shape.

To add a curve, drag the mouse after clicking to create a point. This will open a vector tool that allows you to define the angle, direction and amplitude of the curve.



To finish, you double-click. The first and last points are automatically connected. If you don't want them to be connected, change the Closed property of the shape or choose the open version of the tool.

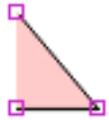
There are situations when you want to have one section of the shape without a border (e.g. in drawing a nose).



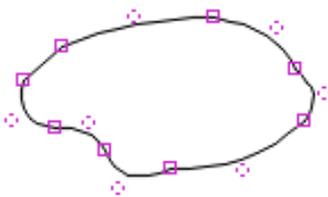
Click the left button to define each point and continue until you have completed the shape. To finish, double-click. After double-clicking, the border is drawn. The fill color is white by default.



Use the Shape Fill tool to color the nose with a flesh color. The border is not connected between the first and last points that were drawn. If we later want to fill in the border between the first and last points, change the Closed property of the shape. To remove the nose color and make the interior of the shape transparent, change the Filled property of the shape. This will give us a line.



Shapes and lines with curves are defined by on-shape points and off-shape points (indicated by dashed circles). Technically, curves are quadratic Bezier splines. You can change the shape of the curve by moving selected points, on-shape and off-shape, by dragging or using the arrow keys.



You can control the appearance of a shape through its:

- 1) Fill
- 2) Border thickness
- 3) Border/line color
- 4) Last border segment

Lines are shapes that are not filled and have a border with the last segment not drawn. Lines can be changed to shapes by setting the Filled and Closed properties to yes. Text does not have a border.

## ***Drawing Lines***

### **Draw Varying Line Thickness**

There are three ways to achieve line thickness variation:

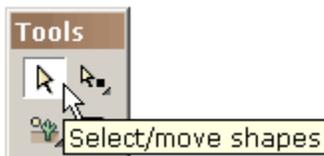
1. In Cartooning mode, you can set a tapered line style in the Shape Properties dialog. It will not work with symbol shapes, shapes with non-linear morphing, or shapes that are filled or have a closed border.
2. Draw two shapes, one on top of the other. Both shapes are set to zero line thickness. The shape behind is then set to the desired line color. By adjusting the points for the shape underneath and making it slightly larger than the shape on top, you achieve the look of a line bounding the shape on top.
3. To achieve a line that looks a little more hand-drawn, use just one shape with the line thickness set to 1. Go along the line and plunk down a few random additional points and intervening curves. Then move these points and curves just slightly out of line, in a random, pencil-like way, until the line looks more hand-drawn. Still relatively straight and all, but with slight variations in placement and thickness so it looks a little more natural.

### **Tapered Lines**

In Cartooning mode, you can set a tapered line style in the Properties dialog. It will not work with symbol shapes, shapes with non-linear morphing, or shapes that are filled or have a closed border.

### **Reshaping**

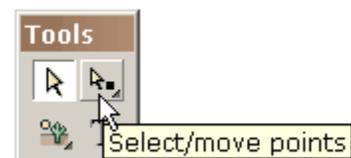
The heart of animation is movement. Movement is achieved by changing shapes from one frame to the next. Change can take many forms: moving a shape from one position to another, moving or transforming points associated with a shape, changing the color of the interior or border of the shape, adding or removing points from the shape. KoolMoves provides a number of tools to accomplish all these tasks without getting in the way of your creativity.

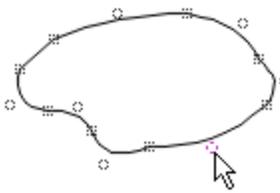


pixels.

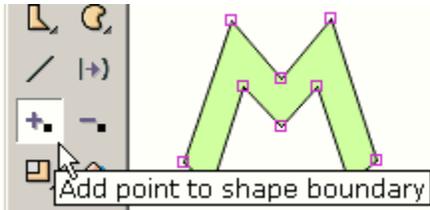
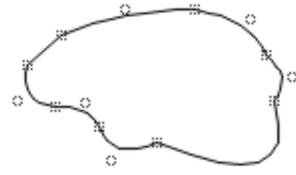
To move a shape or a group of shapes is simple. Select the shape or group of shapes with the shape selection tool and drag the mouse. For fine precision movement, you can use the arrow keys. The combination of shift plus an arrow keys produces a movement of 10

To move a point or a group of points is equally simple. Select the points with the point selection tool and drag the mouse. For fine precision movement, you can use the arrow keys.



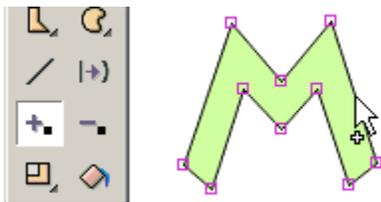
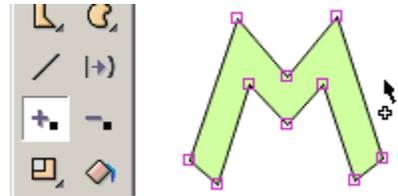


Shapes and lines with curves are defined by on-shape points and off-shape points (indicated by dashed circles). You can change the shape of the curve by moving the points.



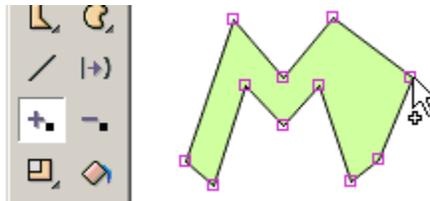
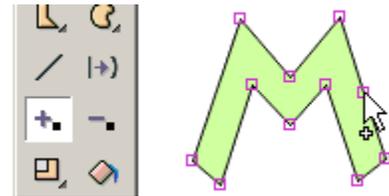
To add a point to the boundary of a shape use the Insert Point tool (whether the shape has a border or not or whether the fill is a color or a bitmap).

You get a cursor that indicates when you are allowed to add a point. It looks like this when you are not on a boundary line.



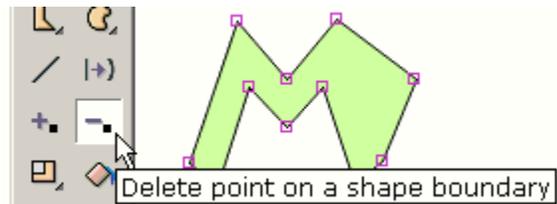
And it looks like this when you are on a boundary line.

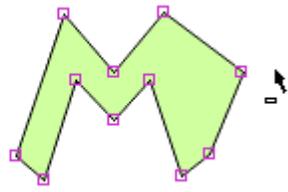
Click the mouse and a point is added.



Without releasing the mouse, you can drag the point to a new position.

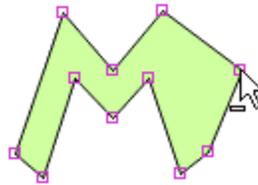
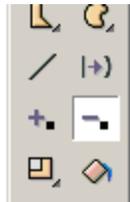
To delete a point on the boundary of a shape use the Delete Point tool.



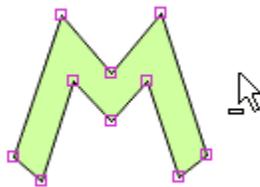


You get a cursor that indicates when you are allowed to delete a point. It looks like this when you are not on a point.

And it looks like this

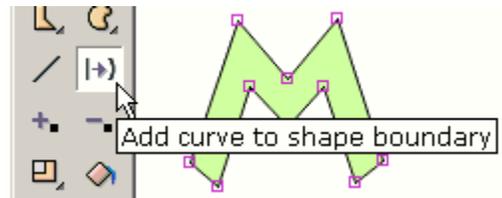


when you are on a point.

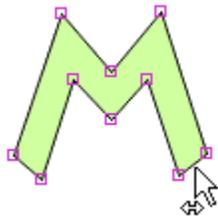
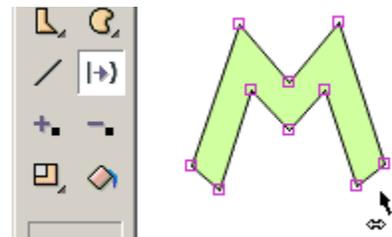


Click the mouse and the point is deleted.

To change the boundary of a shape from a straight line to a curved line, use the Add Curve tool.

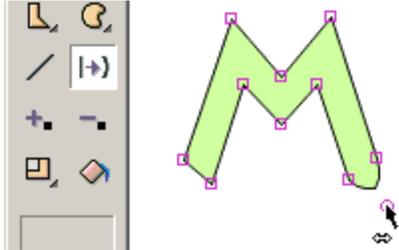
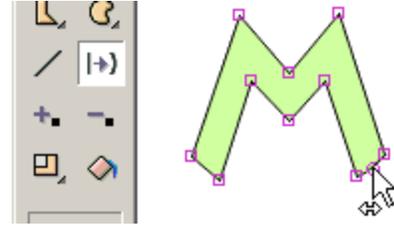


You get a cursor that indicates when you are allowed to add a curve. It looks like this when you are not on a straight boundary line.



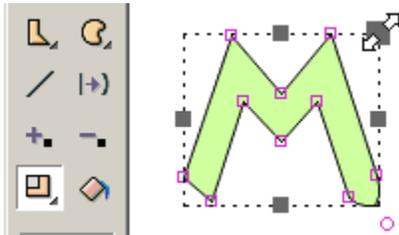
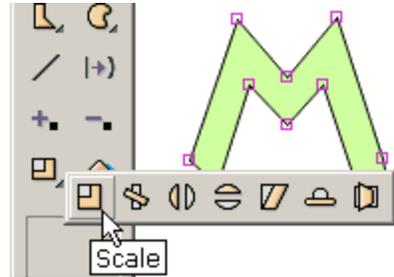
And it looks like this when you are on a straight boundary line.

Click the mouse and a curve is added.

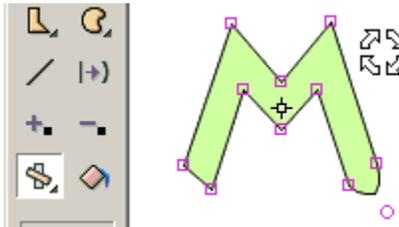


Without releasing the mouse, you can drag the curve point to a new position.

To scale, rotate, flip horizontally or vertically, slant, squish, or create perspective on one or more selected shapes or one or more selected points, use one of the transform tools.

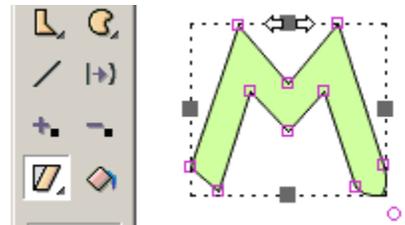


Most of the tools present a set of handles that you can drag in different directions to achieve the effect you want. Shown on the right, the mouse is dragging the handle associated with scaling the same amount in the x and y direction.



The rotation tool lets you define the center of the rotation indicated by the small square plus cross mark. You drag this square to set the center of rotation.

Another tool slants the selection.



Not shown are the flip, squish, and perspective tools. All the tools transform text except slant and perspective.

Squash is the effect used in animation to show the effect of gravity on an object. Stretch is the effect used to show the effect of movement against gravity. A good example would be a rubber ball bouncing. When the ball hits the ground, its shape changes from a circle to an ellipse and as it accelerates back into the air (bouncing) its form stretches slightly. Both squash and stretch can be created by using the squash tool in the effects tool section.

## Bones

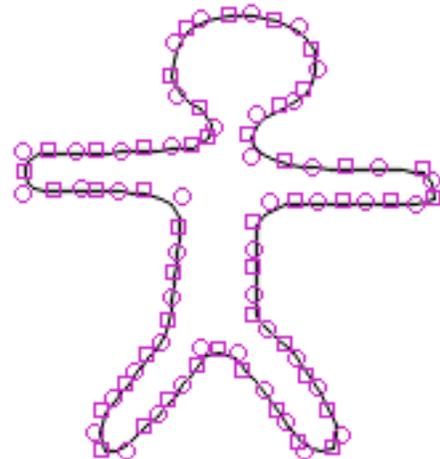
The Bones capability is only available when the Cartooning interface has been selected. Bones allows the designer to attach a skeleton to a figure. In comparison to expensive character animation software, the KoolMoves Bones feature is relatively primitive, but is still a very powerful feature: a shape (or figure) with bones can be fluidly redrawn by manipulating the skeleton. When combined with morphing, Bones allow you to have graceful animations without the tedious task of point by point redrawing.

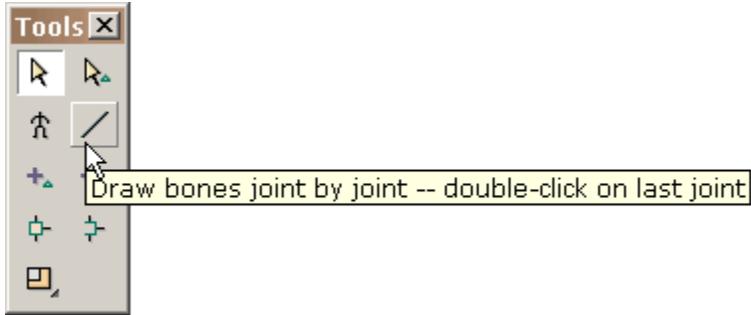
Bones can not be attached to symbol shapes; if you are currently drawing shapes as symbols, you should probably turn off this option in File > Preferences.



When the Interface style is set to Cartooning, the Properties Data View has an additional Behavior field called Bones. This field will only appear when a non-symbol shape is selected.

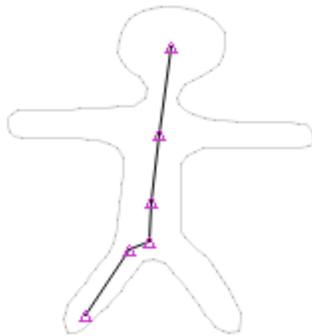
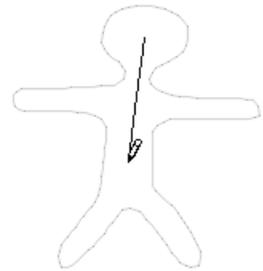
To attach bones to this simple figure, indicate in the properties dialog for the shape that the figure has bones. Bones can be attached to a single shape or a shape group. To add or edit bones, press the Bones button at the top of the dialog.



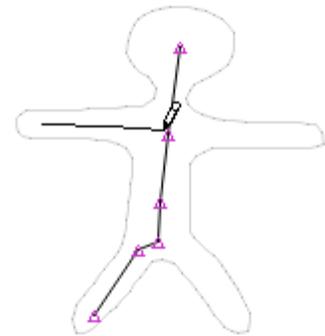


Similar to editing button states, you are given an environment for editing bones against the faint image of the shape to which the bones are attached. The draw bones tool behaves identically to the line drawing tool in the main editing window.

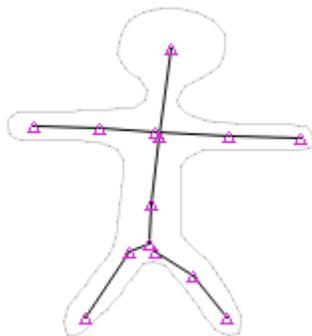
You draw bones by dragging the mouse and clicking the mouse at each joint.



Double-click to finish. When you release the mouse, all the joints are highlighted.

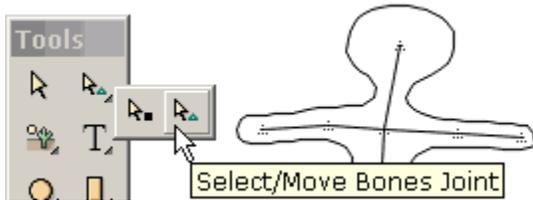
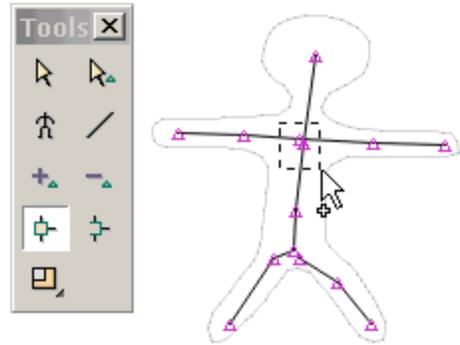


Other limbs are drawn in a similar fashion.



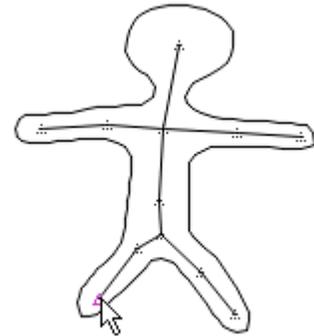
This is what it looks like when all the bones are added.

Certain joints you would like to be shared between different limbs. The way to do that is by using the join tool. Press the join tool and drag a box around the joints you would like to act as a single joint. Under the join tool is an unjoin tool.

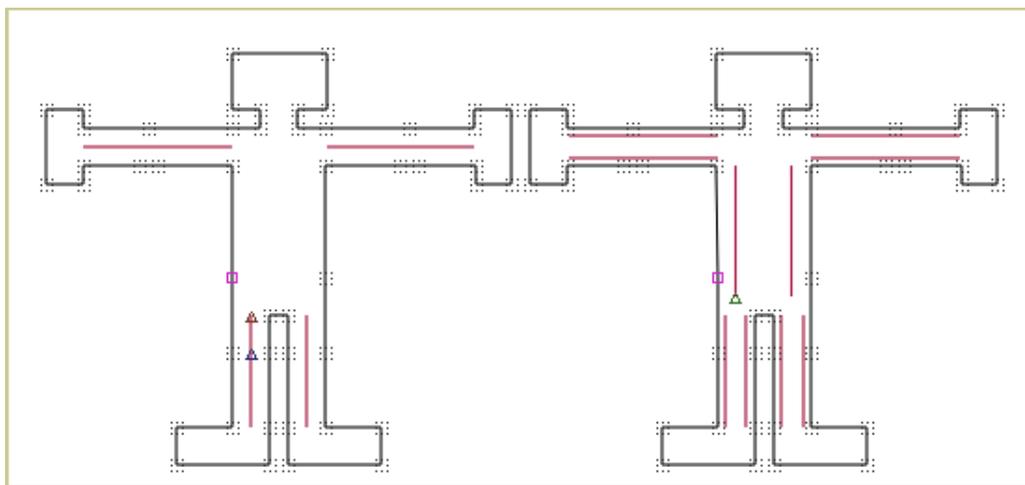


When you are finished editing the bones, press the return to key frame button. This shows the figure with its newly added bones in the main drawing window. You can move joints in this window by selecting the select/move joints tool.

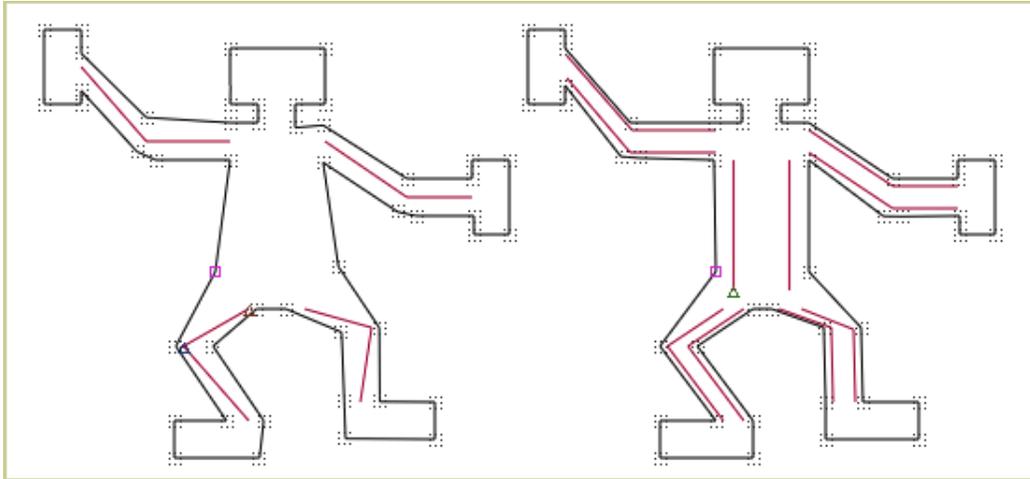
This illustrates selection and movement of a joint. The dynamic interaction between the bones and body of the figure is based on how the joints in the skeleton correlate to the shape points. If more than one joint is associated with a shape point, shape distortion may occur. Shape points that share common joints will tend to move together. The greatest control is achieved when a point is controlled by one joint.



You can view the primary and secondary joints the program uses in controlling each shape point by clicking on Options > View Associated Joints and selecting a shape point (this only works when only one shape point is selected, unlike the modified image below). You can not change the assignment of joints to points.



The figure on the left has two joints controlling the hip point, while the figure on the right has a one to one relation (they are also physically closer). Double Bones, or multiple bones, increase the number of joint locations available for controlling the shape points. The extra bones in the figure on the right decrease distortion and allow for greater control.



## Question to Flashpulse – Comics (flashkit thread 663649)

clejan   
Registered User

Join Date: Dec  
2005

I would like to do comics, so I need to choose from different faces and then easily modify the facial features to express different emotions. I don't need animation at this point, maybe later. Is this possible, and if so, how?  
Thanks,  
-Iuval

FLASHPULSE   
Senior Member



Join Date: Nov 2002  
Location: USA  
Posts: 1,136

Here's how I would do it...

- 1.) The best thing to do is create your character first. (facing forward, left and right)
- 2.) Now delete everything except the eyes and eye-browes.
- 3.) Save this to as FUN file and name it something like "normaleyes\_forward\_character1.fun".
- 4.) Now edit the eyes so they look angry and name it "angryeyes\_forward\_character1.fun".
- 5.) Do this again with different expressions and save them named as they are.

The reason in doing it this way is so when editing your character for expressions, all you have to do is load a FUN file into the scene without editing over and over again.

Now if your eyes for your character were for facing left (for example), I would name it "normaleyes\_left\_character1.fun". And the "character1" stands for the characters name. This makes it faster to read what you want without loading the wrong FUN file.

Now do this for all body parts such as legs, arms, mouth and so forth. I know this sounds like a long process but it will save you alot of time. Hope this helps and let me know if you need more help.

Oh and if you don't add extra "points" when editing and just move the points to create the expressions. Later when you decide to do animation... Example - load "normaleyes\_left\_character1.fun" into frame one and "angryeyes\_left\_character1.fun" in frame two and use the "morph" feature, you will be able to use tweens with your "eyes" (FUN files).

## Chapter 7 – Buttons

Because the interface is WYSIWYG (what you see is what you get), the button appears in the final movie for each frame the button appears in the editing environment. In other words, if the button is missing in a frame, it will be missing in the final movie in that frame. Let's cover some of the facts and vocabulary concerning buttons.

**Button Object:** Buttons are a built-in class of ActionScript. Unlike other built-in classes, there is no button constructor, ie, buttons have to be made manually at author time. In other words, they can not be constructed or removed programmatically. To simulate dynamically created buttons, use movie clips programmed with button like behavior.

The action to be performed by a button, how the action is initiated and the button's appearance is defined either through the properties dialog or programmatically, after the button has been created. Buttons are one of the three classes that KoolMoves can use as symbols (the other two classes are movie clips and KoolMoves components).

**Button States:** A button can have different appearances, reflecting whether the mouse is away from the button ("Up" state), over the button ("Over"), or has clicked (or "pressed") on the button ("Down"). The Up state is created 1) when an item is changed to a button with the Convert to Button tool, or 2) created from scratch when the Create button tool is clicked on. Click on the Over and Down state buttons to create or edit the appearance and actions of the other two states. Once a button exists and is selected, all three states can be accessed by clicking on the states button at the top of the properties dialog.

The Over and Down states can have their own sound, appearance, animation, etc. Many effects are possible, for example, adding a text object to the Over state in addition to the button image has the effect of a Screen Tip. By adding a movie clip to a state you have almost all the effects and features available on the main timeline: add motion scripts and text effects, import swf or flv movies or animated clip art, and so on.

**Copying Buttons:** The actions for a "symbol" button cannot vary from frame to frame. If you want a button's actions to change from one frame to the next, copy and paste the button in the same frame as the original button and then cut and paste the copy to the frame where you want the actions to be different. The procedure of pasting a button to the same frame as its original breaks the link between the two buttons, and allows you to have identically appearing buttons that have different actions (it also has a different name).

**Hit Area:** The button area that is receptive to the mouse. The hit area is expressed as a percentage of the area of the Up state of the button. In other words, a 200% hit area is twice as big horizontally and vertically as the outline of the button when the mouse is not over it.

The Over and Down states do not affect the shape or size of the hit area. This means that if the button appears to have moved when the mouse pointer rolls over it, the hit area has not necessarily moved (however, if someone has embedded the button in a movie clip programmed to move, all bets are off).

**Mouse Events:** there are seven mouse events which can be used by buttons to execute a command:

Press – The button is clicked on.

Release – The mouse button is released after the button is clicked on.

Release Outside - The mouse button is released outside the button after clicking on the button.

Roll Over – The mouse pointer moves over a button.

Roll Out – The mouse pointer moves outside a button.

Drag Over – The mouse button is clicked on the button, and without releasing the mouse button, the mouse pointer is dragged away from the button and then mouse pointer is dragged over the button.

Drag Out - Mouse pointer is dragged outside a button after clicking on the button.

**Text in Buttons:** Button states can contain both static and dynamic text boxes, but the dynamic text boxes should be set to read only. If the mouse pointer clicks on an editable field within a button, the attempt to open and edit the field will interrupt the button execution.

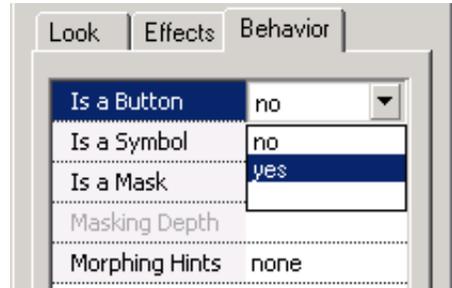
## Creating Buttons

In KoolMoves, any shape, group of shapes, or text box can be turned into a button. Buttons can also be created with an initial empty state, much like movie clips. While buttons can not be created dynamically, after a button is created its appearance and behavior can be modified dynamically with ActionScript.

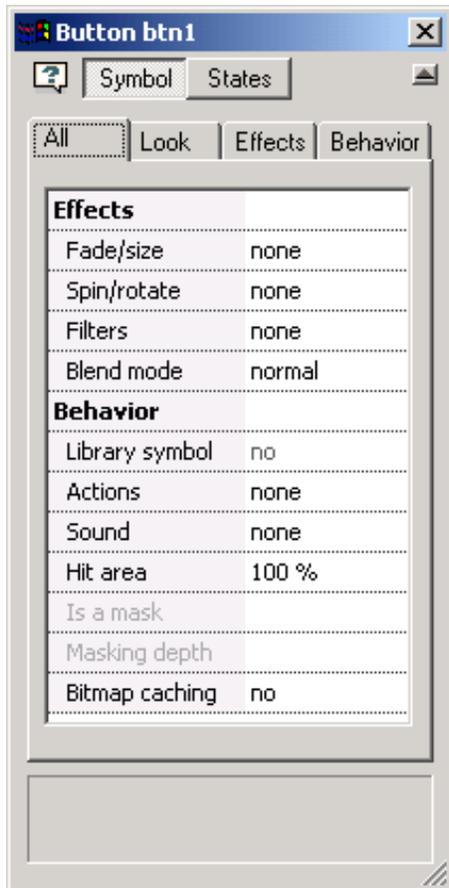
## Converting an item to a Button



To convert the selected shape, symbol or text box (static or dynamic) on the Stage into a button, click on the Convert to Button tool. Shapes, symbols and static text boxes can also be converted to buttons by setting the "Is a Button" Behavior field to yes in the Properties Data View. Objects converted to buttons are placed in the Up state.



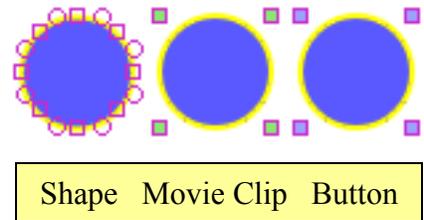
Once an object becomes a button, it can not be changed back; the source object is removed from the stage and placed in the new Up State, much like when objects are converted to movie clips.



Because the source object is no longer on the stage, the Look fields are no longer applicable, and the Effects have changed to Fade/Size, Spin/Rotate, Filters and Blend Modes (the missing effects can not be applied to buttons). The first two effects are covered in Shapes [Effects](#) in Chapter 5, the last two are covered in Movie Clip [Effects](#) in Chapter 11.

The Behavior fields are also fewer, though there is the new Library Symbol field that can be used to add the button to the symbol library and assign a description.

On the Stage, buttons look a lot like movie clips when selected.



## Adding a Button (from Scratch)



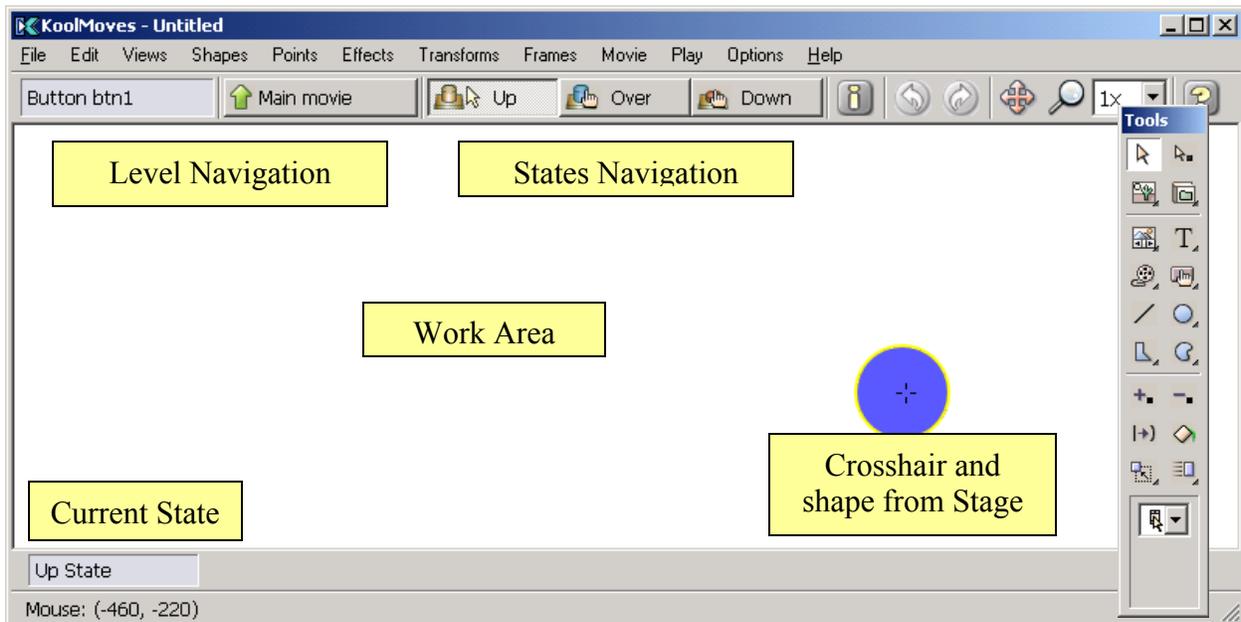
Clicking the Add Button tool creates an empty button and opens the Up State edit screen. Analogous to creating an empty movie clip, returning to the Main Movie without adding any visual elements to the Up State will result in a 10 pixel by 10 pixel empty button

being placed near the center of the Stage. Once the button is not selected it will be completely invisible, and the only evidence of its presence will be its entry in the List of Shapes data view.

## Working with Buttons

### Editing States

With a button selected, click on the States button in the Properties Data View to open the states editing environment. The Up state already exists, and that is where you start. The Tools Toolbar has been moved to the right so as not to obstruct any of the features of this window



If you are starting from the Add (empty) Button tool, the screen will be the same except no shape will be present. The crosshair is either centered on the shape(s) initially selected on the stage or in the case Add Button, the center of the stage in the main movie.

The States editing environment is very similar to the Movie Clip editing environment. Almost all the Tools Toolbars features are functional in a button state, and those that are not can be made functional by first creating a movie clip within the state, and then accessing the function. This is called embedding. You can also use embedding to add Frame Actions to a button.

Within a button state or movie clip, use Options > View Parent Movie to see a semi-transparent overlay of the parent object's contents.

If you press the Over button for the first time, a popup will ask if you want to create the over state from a previously drawn state or if you want to draw it from scratch. If you are only changing the color of the state, you would select copy the up state and edit the color from the value corresponding to the up state to the value you want to correspond to the over state.



You have the option to included embedded actions when copying a state, and the option to decide on whether the actions are linked (share movie clip frames) or completely separate actions.

In the Over State you can change the properties of the over state and edit the shape corresponding to the over state. A faint image of the shape corresponding to the up state is displayed under the shape for the over state for alignment assistance. All of the above information about the Over State are equally applicable to the Down State.

### Adding Sound to Button States

If you want a sound associated with the Over State, return to the Main movie level and add the sound to the Over Sound tab under the Sound field. If you want a sound associated with the Down State, add the sound to the Down Sound tab. Setting the Loops value to 999 will have the sound continuously loop.

The Over State of a button is activated for roll over, roll out, drag over, and drag out mouse events. The Down State of a button is activated for press and release mouse events.

At this time, KoolMoves only supports the PCM wav and the mp3 file types. If you use the same sound file multiple times throughout the movie, it will be referenced that many times in the Flash™ movie but only defined once.

Sound files tend to be huge which can make the exported swf file huge. To conserve resources, take these steps: in a sound editing program convert from stereo to mono and to the smallest sample rate that gives you the quality you need. Shortening the sound track also helps.



Sounds do not play when running your project in KoolMoves' internal player.

If a sound plays longer than the movie plays and if the movie loops, the sound from the first loop will continue playing during the second loop until the sound from the first loop finishes. This effect worsens as the movie continues to loop. To solve this problem, add the Stop Sound action to the last frame in the movie.

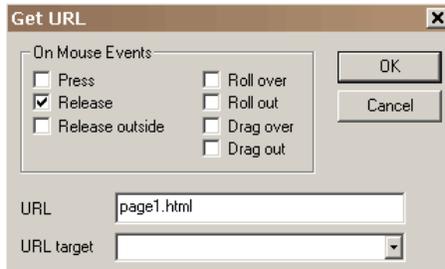
If your sound file is sampled at a rate other than 5500, 11025, 22050, or 44100, the sound will play in the Flash player at a rate nearest to one of the four rates listed above.

## Adding Actions (Go To URL)

Click on the actions field in the properties window to assign and edit the actions for the button.

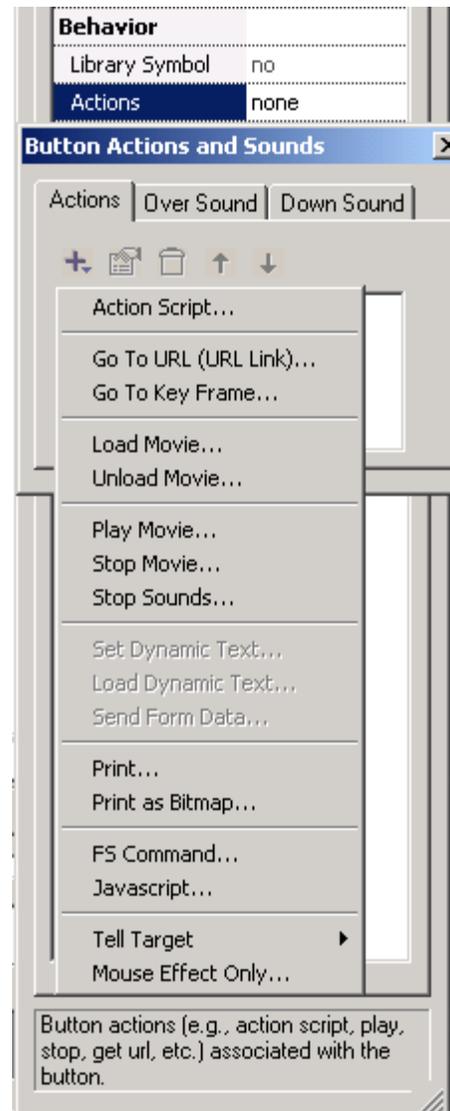
The + button presents you with the possible button actions, most of them self-explanatory. Each of these actions can respond to one or more mouse events. With the exception of Send Form Data, the other available button actions are covered in detail in the [Score Data View > Actions and Sounds](#) tab. Send Form Data retrieves text from dynamic text objects and sends it to a URL.

Selecting Go To URL (equivalent to the ActionScript `getURL()` global function) presents us with this popup. We can select which mouse events trigger the Go To URL action.



For use on the web, the URL for the loaded movie can be an absolute or relative URL, but, for testing purposes, all the movies must be in the same directory without directory or disk drive specifications in their names. Also, there should be no spaces in the name as this will cause problems in Netscape web browser. `Http://` should not be in the URL for a loaded movie because it will also cause problems with Netscape web browser.

The optional URL Target field (comparable to the window argument in `getURL()`) can be used to load the document into a named browser window or browser frame. If this field is left empty, the document is loaded into the current browser window or frame.



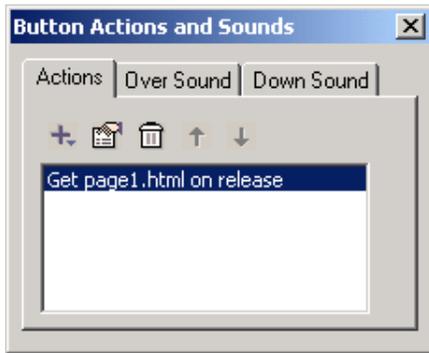
The dropdown list has four preset values: blank, \_parent, \_self and \_top.

\_blank – Opens the document in a new, anonymous browser window.

\_parent – Replaces the frameset that contains the current movie.

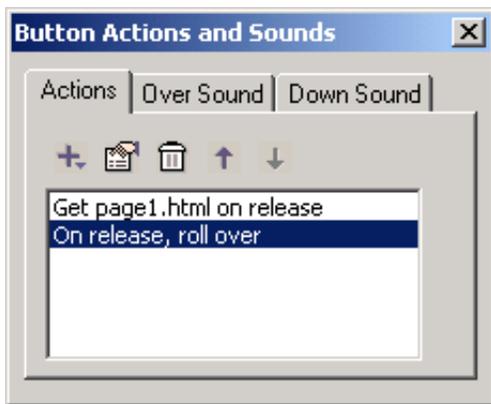
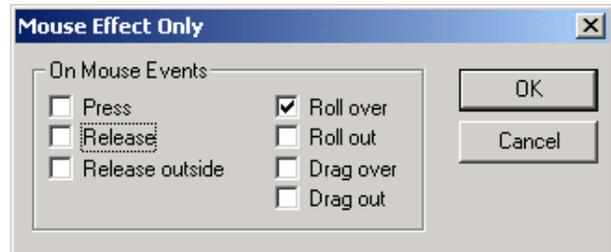
\_self – Loads the results of a script execution into the current browser frame or window.

\_top – Replaces all framesets in a web page with the specified document.



On pressing OK, the action is added to the list of actions associated with this button. Any action can be edited or viewed in more detail by double-clicking on it or by selecting it and then selecting the properties button (second from the left at the top of the popup).

If we want a roll-over effect when the mouse passes over the square in addition to going to the URL on mouse release, we add a second action, Mouse Effect, to our list of actions and select Roll Over as a triggering mouse event. Mouse Effect action is unique in that it only controls the appearance of the button with no resulting action taken.



On pressing OK, we see both actions associated with this button.

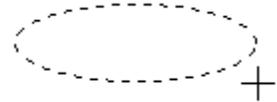
## Address Formats in Actions

When testing a movie using Play in Browser, any HTML and SWF files, referenced by Get URL or Load Movie actions, MUST be in the same directory as the fun file for these button actions to access these files or set the directory for playing the movie in File > Preferences. Load movie loads a movie without closing the current movie. All of the higher levels have transparent

backgrounds which allows to see the layers underneath. Unless you include in the loaded movie a shape under all other shapes that equals or exceeds the size of the movie frame and that is filled with an opaque color or a bitmap, you will see the first movie underneath the loaded movie. The movie background color, size, and speed are set by the movie at level 0. None of the button actions function in the KoolMoves internal player.

### Roll-Over Example

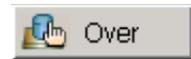
This basic tutorial will define the steps necessary to create a simple roll-over effect using an oval button with a gradient fill. We start by drawing an oval.



The oval is given a gradient fill and the border is removed by setting the border thickness to zero.



Click on the Convert to Button tool. To produce a roll-over effect, the button must have an over button state; click on the Over State button at the top of the screen. Now we are ready to define the over state of the button (the button's appearance when the mouse is over the button).



Because the over state has not been defined, we are asked if we want to create the over state from another state we have created or if we want to draw it ourselves. We select to copy it from the up state. If we had selected the last option to draw it ourselves, we would be free to draw any text or non-text shape. If we position the over state with an offset relative to the up state, when the mouse goes over the button while the movie is playing the button will shift in

position reflecting that offset.

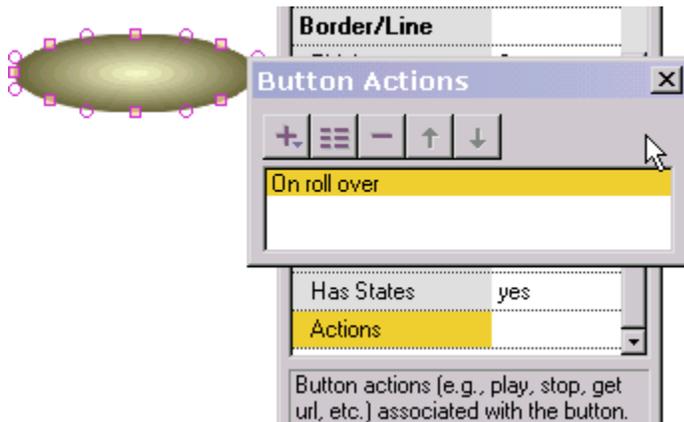
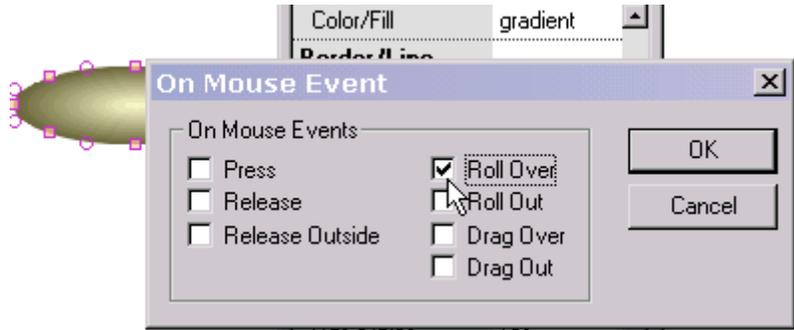


To make the button's over state different than the up state, we change the gradient colors of the button's over state. After selecting the oval shape in the Over State, open the Properties Data View and launch the Color/Fill dialog.

We press the Main movie button at the top of the properties popup to exit the over state and we are ready to test the roll-over effect just created.

If we want an action to occur when the button is pressed, we would select Get URL, Go To Frame or whatever the action is. We can have as many actions occur responding to mouse events as we like. But to achieve a simple roll-over effect which is our current task, we only need to select Mouse Event.

We check Roll Over. If we had a down state defined, we could check Release and/or Press as well.



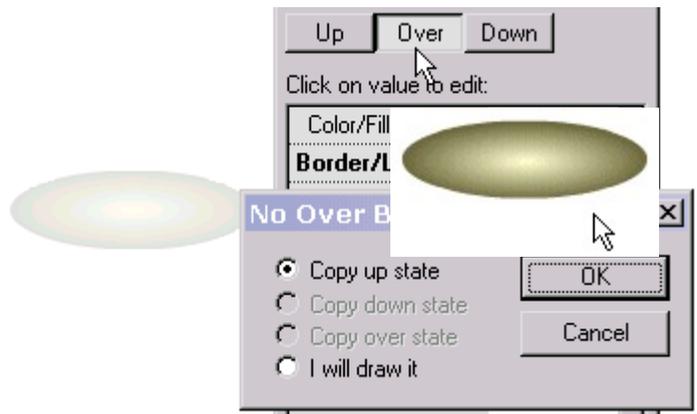
After pressing OK, we return to the prior popup and see the action we just defined in the list of button actions. By double-clicking on it we can edit the action. By selecting "+" we can add another action.

We select play in stand-alone player. With the mouse away from the button we see the gradient color associated with the button's up state.



With the mouse over the button, we see the gradient

color associated with the button's over state. Our test is successful!



To use as a navigation element, we could place text over the oval. On moving the mouse over the text, the background oval would change color.

## Printing from Within a Movie

The **print()** and **printAsBitmap()** ActionScript commands provide the capability to print frames from within a movie. The **print()** function will send vectors to PostScript printers and rasters to non-PostScript printers. **printAsBitmap()** only sends raster information but supports alpha channels and color transformations. **printAsBitmap()** is functionally identical to **print()**.

```
print (level)
print (level, "Bounding box")
print ("target")
print ("target", "Bounding box")
```

Options for the Bounding box :

- bmovie (the bounding box of a specific frame)
- bmax (all of the bounding boxes of all the printable frames)
- bframe (the bounding box of each printable frame)

To limit the printing to specific Key Frames, label the selected key frames to be printed as #P, otherwise all frames (including tweens) will be printed.

## Tell Target Actions

Tell Target is a means to communicate across multiple timelines. Any movie can control another movie. Your movie has multiple timelines when

1. You create a SWF Object
2. You load a movie into a layer

There are three basic Tell Target functions built into KoolMoves.

1. Stop
2. Play
3. Stop Sounds

More will come later. These actions are located in the Advanced section of the Add Action popup menu.

The tell target actions work just like regular actions with one difference - you must define a target in the target box. Inside a SWF each target is laid out much like a directory. To access the current main movie's timeline you can use: /. This can be used anywhere to control the current movies timeline. The ../ syntax can also be used to control a higher movie's timeline. So, for example, lets say you want to control the main timeline from an imported SWF Object with Tell Target you would use the ../ syntax. For loaded movies the syntax is a little different. To use Tell Target on a loaded movie you use `_leveln/` where n is the level of the loaded movie. If, for example, you want to control a loaded movie in level 1 the target would be: `_level1/`. To control a SWF Object from the main movie you can use either `/Object name` or just `Object Name`.

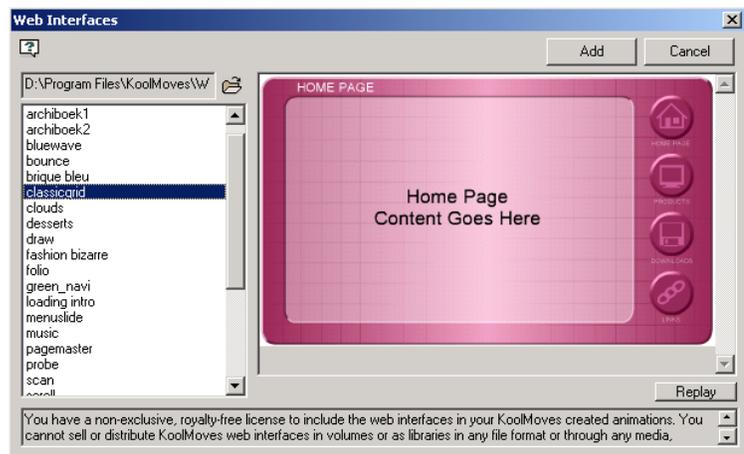
## Chapter 8 – Toolbox Libraries

The Tools Toolbar Library drawer allows users to work with pre-packaged text effects, clip art, web interfaces and buttons. These visual elements are stored in fun files, which means that you can load these files like regular projects and edit them, but we recommend that you use the toolbox to import them into your project. They can also be imported into movie clips.

While Buttons and Clip Art are added to the current frame when imported, Text Effect Templates, Animated Clip Art and Web Interfaces add additional frames to your project. [Appendix E – Toolbar Library Listings](#), has a list of available Motion Scripts, Text Effect Templates, Web Interfaces, Clip Art Gallery Categories, Animated Clip Art and Button Gallery Categories.

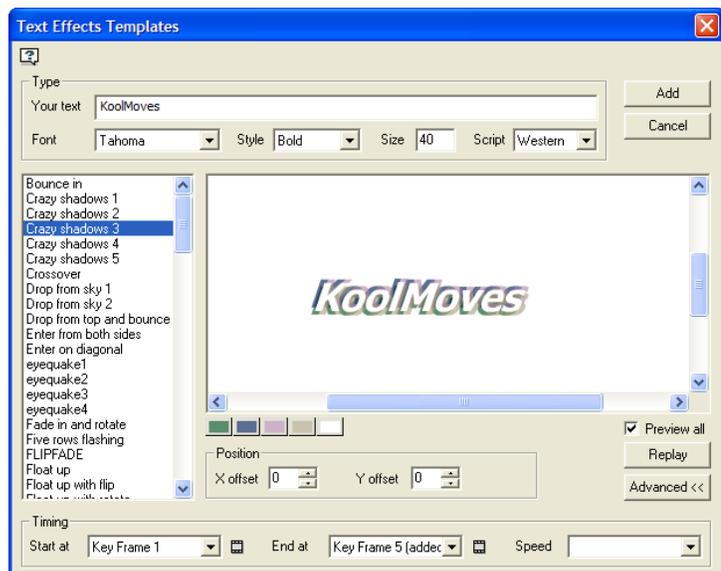
### Web Interfaces

 The web interfaces library was created by several professional graphic designers. You can modify the content, color, and behavior – have a working home page in minutes. They also serve as examples that you can learn from.



### Text Effects Templates

 KoolMoves provides a library of text effects templates. Select Libraries > Text Effects. These are regular KoolMoves .fun files which provide the layout and motion paths for text effects. You enter your text and font selection and the software performs the text and font replacement. Select a text effect from the selection of text effects on the left. You can see the text effect in real-time on the right, superimposed over the background movie.



Under the animated text effect, there are colored buttons. Each button represents a color in the template. You can change the colors of the text effect by pressing these buttons and selecting different colors.

To reposition the text effect on the screen, change the values of the X Offset and Y Offset. Zero offsets correspond to a centered position.

To change where the text effect starts relative to the underlying movie, select a Start At key frame (under Advanced>>). Frames that exceed the current length of the movie have "added" in parentheses. The button to the right allows you to view that key frame. The text effect has a natural frame length which is reflected in the End At key frame. You can override this if you want to stretch the text effect to cover more of the underlying movie. In most cases, you will not need to change the End At key frame. Changing the End At key frame does not affect the speed of the text effect because as you lengthen the text effect, the tweens for each frame are decreased. To control speed independent of number of key frames, set the Speed at a setting between Very Slow and Very Fast or change the speed of the main movie (Movie > Movie Speed). The Speed setting affects the number of tweens between the frames.

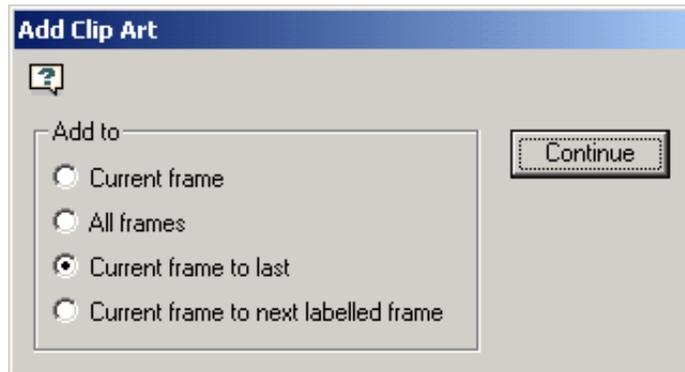
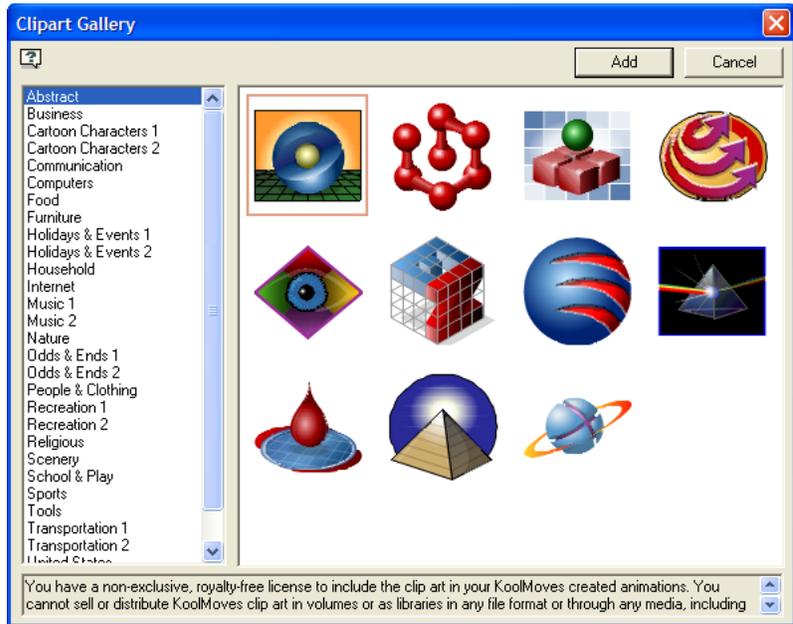
To add the text effect to the underlying movie, press Add. You are free to use any of the positioning, coloring, and transform tools to edit the text effect at this time or you can undo the text effect by pressing the Undo button.

If you want the last frame of the text effect to play for a long time, increase the number of tweens for the previous frame or duplicate the last key frame a number of times using Frames > Insert Copy of Frame. Also, you can use Edit > Paste to End of Movie to paste the text characters in the last frame of the text effect to the remainder of the movie.

## Clip Art

 260 clip art items are included with the registered version of KoolMoves. They were created in KoolMoves and were designed specifically for use in Flash movies. The clip art is found in the Clip Art folder inside the KoolMoves folder.

Click on a folder category to display the contents. Click on an item to select it and then click on Add to choose on which frames to add the image. .

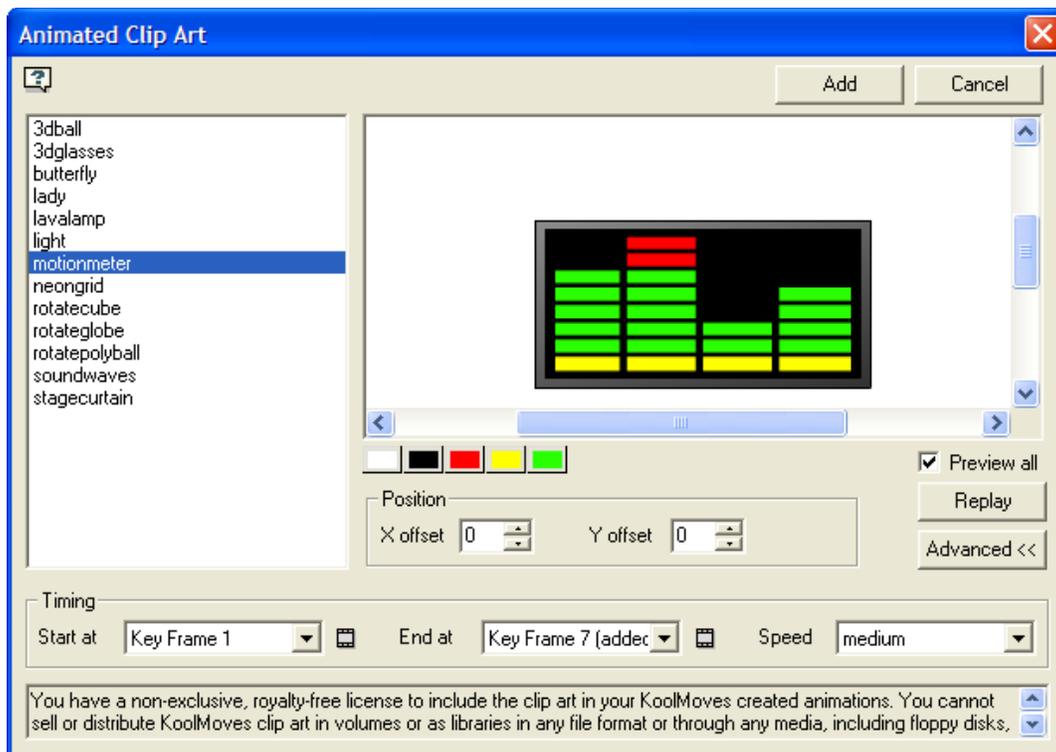


For information on importing third party clip art, see the [Importing Clip Art](#) entry in Chapter 9. For a listing of available clip provided with KoolMoves, see the [Clip Art Gallery Categories](#) entry in Appendix E.

## Animated Clip Art

 Animated Clip Art can be viewed as independent animation projects, developed with their own movie speed and tween settings. Additionally, unlike Clip Art, which consist of vector graphics stored in a single frame .fun file, Animated Clip Art are multi-frame .fun files. When animated clip art is imported, the host project's speed setting applies so adjustments may be necessary; frame duration/tweens will be based on the destination projects default setting. If the animated clip art requires more frames than are available at its insertion point, KoolMoves will add additional frames. It is often simplest to import animated clip art into movie clips where they can have their own tween setting separate from the main timeline.

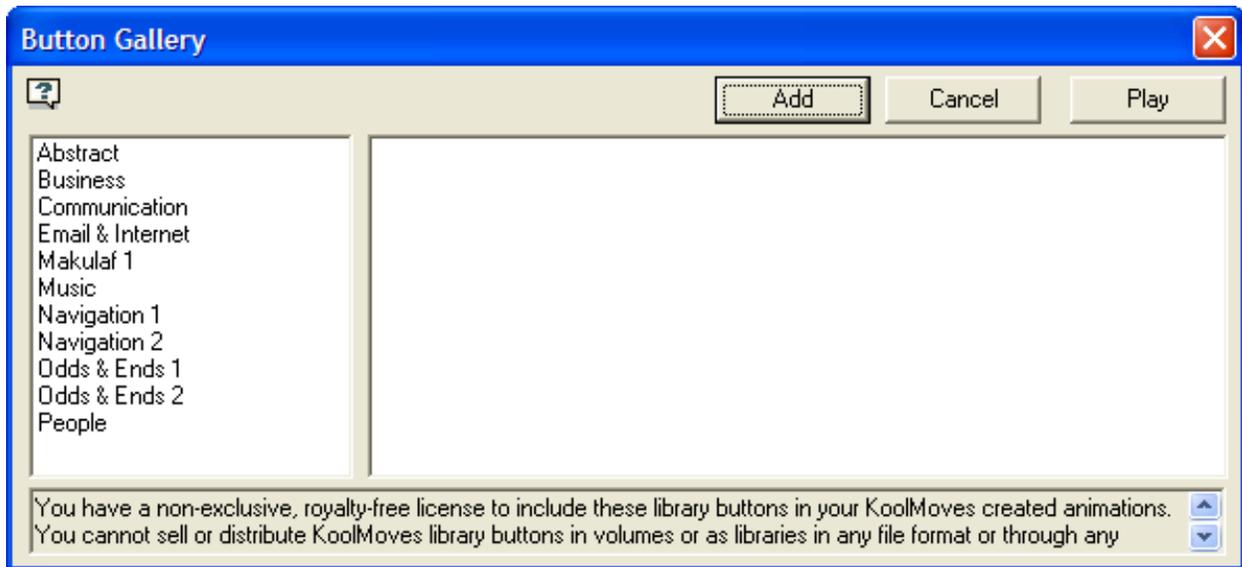
Select from the names of animated clip art in the left list box and press Add to add to your animation. To preview the entire movie, check Preview All. Otherwise, only the animated clip art will be previewed. The default directory is \KoolMoves\Animated Clip Art\.



- Color Boxes – User can change solid color fills used by animation.
- X & Y Offset – Adjust the placement of the clip art relative to the center of the frame.
- Timing, Start at – Select an insertion point for the animation – defaults to active frame.
- Timing, End at – The process will add frames to the current project if required.
- Timing, Speed – adjusts the number of tweens used by the animation, based on the default tween setting of the destination project.

## Button Gallery

 About 70 buttons are included with the registered version of KoolMoves. They were created in KoolMoves and were designed specifically for use in Flash movies. The buttons are found in the Button folder inside the KoolMoves folder. Click on a folder category to display the contents. Click on an item to select it, press Play to see how it responds to mouse actions, and press Add to add the button to your movie. Upon pressing the Play button, the button will display in the Flash stand-alone player if you have one; otherwise, your default web browser will display the button. Move your mouse over the button and click on the button to see the different behaviors of the button.



Buttons created with KoolMoves prior to version 5.1 are grouped shapes, with one of the shapes designated as the button and the rest of the shapes just regular shapes. The button shape is probably at the top or bottom of the shape list. See the [Button Gallery Categories](#) entry in Appendix E for a list of buttons provided with KoolMoves.



## Chapter 9 – Importing Images

### **Static Images (BMPs, GIFs, JPGs, PCXs, PNGs, and TIFs)**



This function, accessible through File > Import Image or the Image Import tool, creates a rectangular shape filled with the image file you select. The initial image shape and size is based on the image file, but the four points of the rectangle can be individually moved or more points can be added. Alternatively, you can create a shape and fill it with a bitmap using the shape fill tool – the image will tile starting from the top left corner.



Some things that are generally true: 1) bmps are not the most efficient format; see if the image is available in other formats, 2) gifs are smaller than pngs and adequate for screen based purposes, but pngs of the same image may be crisper and often look better in print media.

Problem – jpeg loads as blank grey rectangle. This can occur with jpegs used in components. Try saving the jpeg in non-progressive mode or in a different format before importing.

Problem – png is distorted. If this occurs, try saving the image as an interlaced png before importing.

### **Vector Graphics**



Vector Graphics are imported using File > Import Vector Graphic or the Vector Graphic Import tool. They fall into three categories: EPS, SVG and Clip art.

#### **EPS files**

Encapsulated Postscript (EPS) files can be difficult to import as vector shapes into KoolMoves. It should be at least EPS version 3.0, and it should contain vector shapes, not just bitmaps.

#### **Import and Export SVG**

Scalable Vector Graphics (SVG) is only partially supported at this time. The current implementation is primarily focused on importing and exporting outlines.

#### **Importing Clip art (DXF, EMF, and WMF)**

WMF, AutoCAD DXF, EPS, and EMF files are converted into KoolMoves shapes with points when they are imported. There are several classes of clip art including bitmap and vector. KoolMoves only supports vector clip art in which each shape is defined by boundary points and is usually filled with a uniform color. Color fill information may not import with DXF files.

Vector clip art often has a high density of shape points so you are given the option when you import of reducing the number of points through curve fitting, automatically performed by KoolMoves. If detail is lost or there are wild shapes, increase the number of points on importing (i.e., decrease the amount of point reduction).

Not all clip art objects are imported. For example, text objects are not imported at this time.

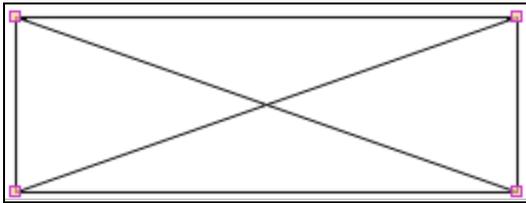
You can get free vector clip art from [www.openclipart.org](http://www.openclipart.org), [www.arttoday.com](http://www.arttoday.com) and [www.clipartdownload.com](http://www.clipartdownload.com). You can purchase single clip art images from [www.eyewire.com/clipart/](http://www.eyewire.com/clipart/). Corel publishes several large clip art collections. Or, you can draw with a number of vector graphics applications and export in a format compatible with KoolMoves.

To edit clip art in order to impart motion for example, lasso the part of the clip art to be edited with the point selection tool. The selected points can be translated using the arrow keys or dragged using the mouse, rotated, slanted, squished, scaled, or put into perspective. See [Reshaping](#) in Chapter 6. Typically, you reshape or recolor the clip art in the current frame, add another frame, make changes in that frame, add another frame, and so on to create a movie. KoolMoves morphs the shapes between frames to create smooth animation. The speed of the motion is set by the overall speed of the movie and the number of tweens between frames.

### **SWF Flash Movie**



A Flash™ movie (swf) can be imported as an object. To Import a SWF Object either select File > Import Flash Movie as Object or use the import button located with the drawing tools (the last choice in the submenu "drawer"). The SWF Object can be located anywhere inside a KoolMove's movie and manipulated like any other symbol. Tell Target actions can issue stop, play and stop sound commands to the imported movie.



The movie has its own timeline and cannot be viewed or edited in the KoolMoves editing environment. It will display as a rectangle to show the size and position of your imported SWF and you won't be able to see what the complete movie looks like until it is previewed in a web browser or stand-alone Flash

player. You may scale, squish, rotate, move, apply an effect, and flip a SWF Object. Wherever the SWF placeholder rectangle is moved is where the SWF Object will play. The imported SWF will retain all of its sounds and actions of the SWF that you imported.

The imported swf must be in the same directory as the swf referencing it. The imported swf can also be imported as an editable movie; however, not all of the elements are currently imported.

### **Flash Live Video**



To import AVI, MPEG, QuickTime (MOV) and WMV, first convert the video to Flash LiveVideo (FLV) format. You can find a free encoder available on the internet (search on "free flv encoder"). At this point you can import the FLV file with the FLV import tool or through File > Import FLV.

### ***Animated GIFs (Work Around)***

KoolMoves supports import of static GIFs with and without transparency. It does not support import of animated GIFs at this time.

For transparent images exported from PhotoShop, use 255 color PNG or GIF (with transparent color, NOT transparent layer!).

Use an application found at [www.segon.com](http://www.segon.com) to convert your animated GIF into a SWF file. Then use import swf as an object to bring the converted gif into your KoolMoves project.

KoolMoves does not directly support output of animated GIFs at this time. To convert a KoolMoves created swf to an animated gif, you might try Magic Swf2Gif at [itinysoft.com](http://itinysoft.com).

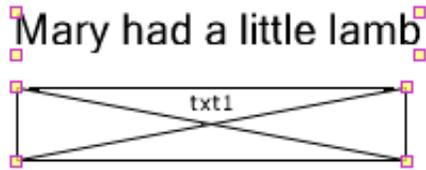
There are a number of other specialized products for creating animated GIFs. Ulead makes a nice product called Gif Animator. There are also free products. See <http://msdownload.microsoft.com/msdownload/gifanimator/gifsetup.exe>, a simple free GIF animator from Microsoft, and from the makers of Game Maker, a very easy to use game creation and development environment, <http://www.cs.uu.nl/~markov/kids/gmaker/index.html>.

Capture the KoolMoves animation as a series of screen images using File > Export Movie > Capture Movie Frame and import the image files into your favorite gif creation tool. With some of the free GIF animators you may need to convert the captured images into GIF format using a graphic editor.



## Chapter 10 - Text

Text as displayed in Flash can be split into two groups: Static and Dynamic. Static Text is created and formatted at authoring time and, except for effects that can be applied to the text box, does not change throughout the movie. Dynamic Text is stored in ActionScript TextField class objects, and can be created and formatted dynamically; however, the KoolMoves Properties Data View allows you to access many of these properties in your project without requiring any knowledge of ActionScript.



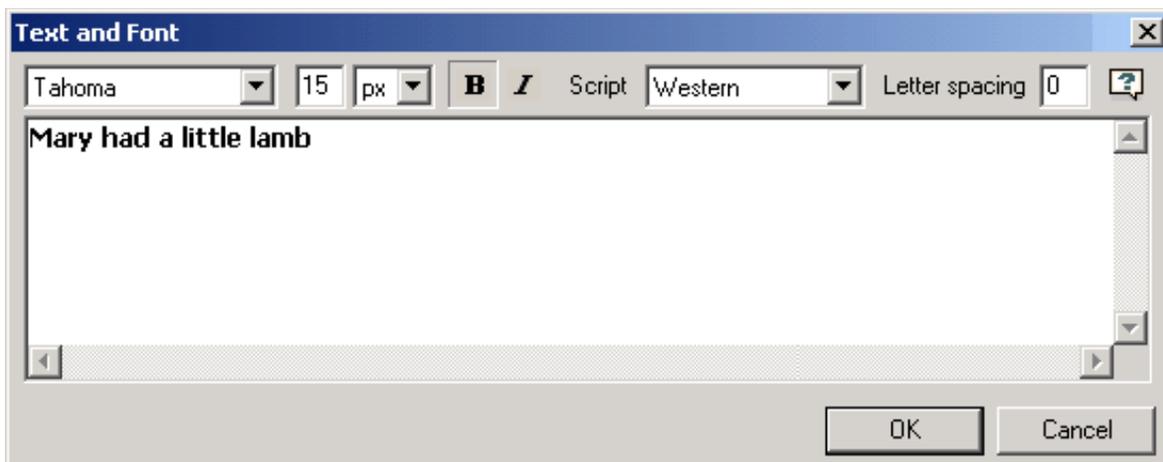
Static and Dynamic text objects have the same bounding box markers, but are displayed differently. Static text is displayed as it will appear in the export movie, while with dynamic text only the physical dimensions and name of the object are displayed at authoring time on the Canvas.

The Text tool on the Tools Toolbar allows you to place text on the Canvas. You have the choice of placing Static text or Dynamic text. The default choice is static. Static text is heavily anti-aliased in the Flash player and is blurry at small type size. This problem cannot be controlled by KoolMoves. FFF Compact is a good font for small type. Dynamic text is lightly anti-aliased and is rendered exactly like text in HTML pages.

### Static Text

**T** Static text is entered through an interactive window that allows you to customize the text: font, font style, size, and letter spacing: type text in the large edit box and make choices. You can edit selected text by double-clicking on the text or right-clicking and selecting Properties. Scale, rotate, flip, and squish transformations will affect text. You cannot manipulate the points associated with a text object. If you desire to do that, convert the text to shapes with points (in the Transforms menu). Static text can not change during movie execution.

Certain format aspects can only be set through the Text and Font dialog, while others are handled through the Properties data view.



## Text and Font Window Properties

- Font – Only a TrueType non-symbol font can be selected for text.
- Size – Enter an integer value.
- Size unit – Choose between Pixel or Point sizes.
- Font Formatting – Turn Bold or Italics on or off for entire text box contents.
- Script – This parameter can affect how the font prints out.
- Text Letter Spacing - Powerful yet simple way to add elegance to text.

## Properties Data View

### Look

#### *Type*

Opens the Text and Font dialog window.

#### *Color*

Set the font color.

#### *Alignment*

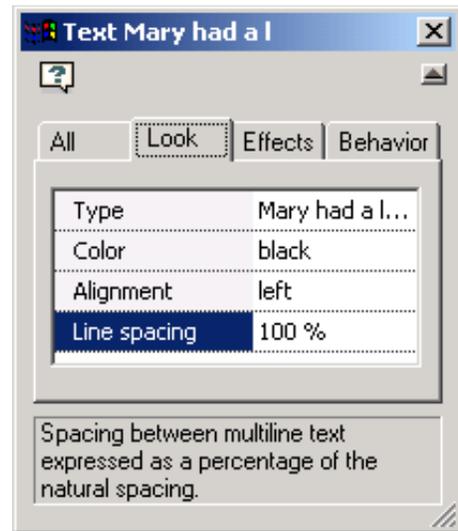
Multi-line text can be left, right, center, or full justified. In the case of full justification, the last line of text is left aligned.

#### *Line Spacing*

Set the line spacing as a percentage of the natural spacing for that particular font type and size.

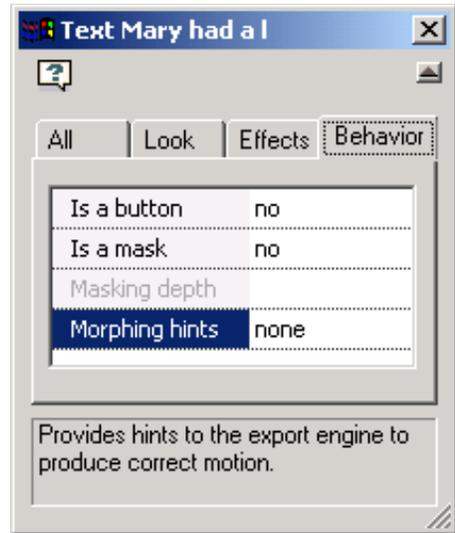
### Effects

See Chapter 5 – Data Views, Properties Data View, [Effects](#).



**Behavior**

See Chapter 5 – Data Views, Properties Data View, Behavior – Shapes.



**Creating Your Own Text Effect**

It is easy to create your own text effects. The basic principle is to create the stages of the text effect in a series of key frames. This example illustrates a series of letters which write on the screen starting very large and transparent.



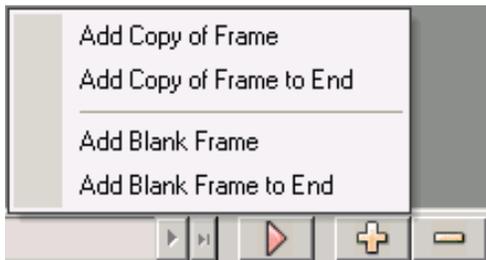
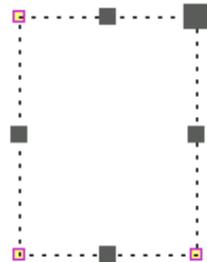
In step 1, the Static Text tool is clicked and a single letter is entered. The size of the letter does not matter because we are going to scale it. The letter is shown on the right.



In step 2, the letter is made transparent using the paint bucket tool.



In step 3, the letter is scaled to a very large size using the scaling tool.



In step 4, the current key frame is copied and added to the end of the movie.



In step 5 in the last key frame, the letter is scaled to the final size using the scaling tool and the transparency is removed using the paint bucket tool.

The current key frame is copied to the end of the movie and the process is repeated for the next letter in this case "o". And so on for each letter. Some fiddling of the sizes of each letter will be necessary to produce a polished look.

When the movie is played, the effect is that each letter appears in sequence. The text effects you can create are endless. See [Text Effect Templates](#) in Appendix E for a complete listing of similar text effects provided with KoolMoves.

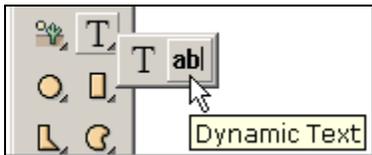
### Transforms > Text to Characters

This menu command breaks apart a text string into a series of text strings for each character.

### Transforms > Text to Shapes

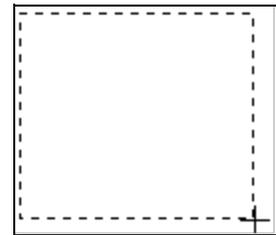
This menu command converts a text string into a shape with points. You would probably only want to do this if you need to apply certain transformations to the text such as slant or perspective, to manipulate the individual points of characters comprising the text, or to use a bitmap or color gradient fill. Text in the form of a text object is very efficient because the characters can be shared with other text strings with the same font.

### Dynamic Text

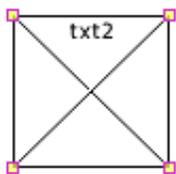


Dynamic text allows user input or can be changed within the movie. It is the second choice in the text tool "drawer".

The dynamic text field is created by clicking on one corner, dragging the mouse and then releasing to define the opposing corner. There are a number of properties unique to dynamic text for specifying the look of the dynamic text.

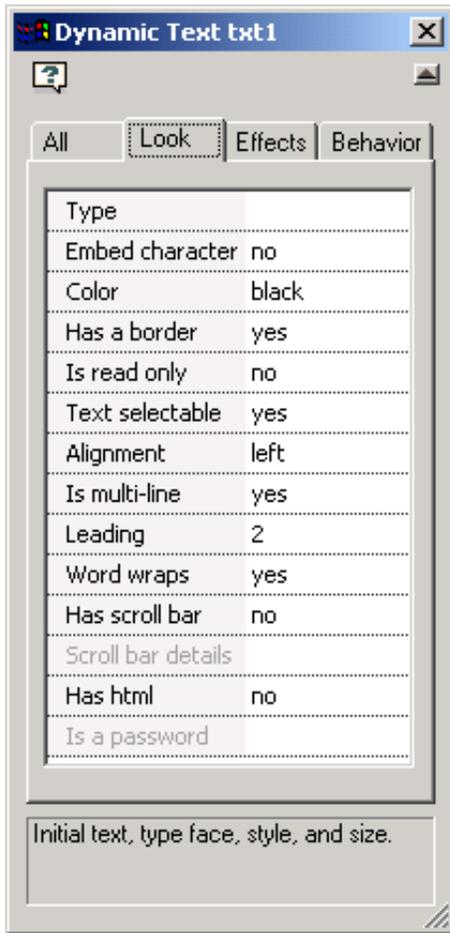


Unlike the Static Text tool, the act of creating the dynamic text box does not open the and interface. Because the dynamic text box can be initialized dynamically, initial text is not required and for many applications is not desired. The static and dynamic text boxes share many features in common – only the unique dynamic text properties will be addressed below.

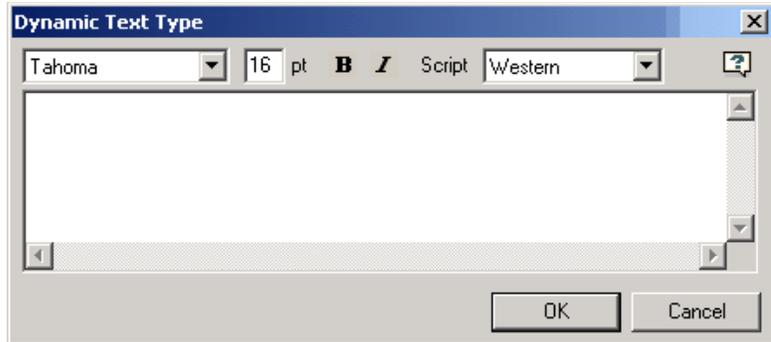


Unlike the Static Text box, no text is displayed in the editing environment. In the editing environment the text field area is defined by a box with crossed lines.

## Look Tab



The Color and Alignment properties are identical for static and dynamic text boxes, and the Type property is almost identical. The interface for setting the initial message font, style and font size is called the Dynamic Text Type window.



### Embed character

Embedding fonts gives the text a consistent appearance regardless of the PC the movie runs on. Embedded fonts are also antialiased. However, the default value of no allows for greater robustness: antialiased fonts below 10 points are usually unreadable, embedding a font adds 20 to 30KB to a movie, and if the font can't be found, no text will print in the box. With font embedding disabled (the default), something will always print, even if its in the PC's default font.

### Has a border

Adds an outline to the text box. Set the border color dynamically with the `textField.borderColor` property. If a border is set, the background of the text field is white with a black border. To have a different background color, you will need to turn border off and create a shape behind the dynamic text object with the color and border of your choice. Dynamically, backgrounds and background colors are controlled with the `textField.background` and `textField.backgroundColor` properties.

**Is read only**

When set to “no”, the text box can be edited and / or used for user input.

**Text selectable**

Determines if the text box contents can be selected and copied to the clipboard.

**Is multi-line / Is a password**

“Yes” enables multiple-line user input, however, the Password field requires the multi-line capability set to “no”. Is a password = “yes” replaces each character in the field with an asterisk (“\*”).

**Leading**

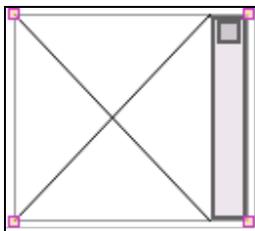
Where static text used a percentage of the normal line spacing, dynamic text has a range (0 to 10 in increments of 2) of points for adjusting line spacing. Default value is 2.

**Word wraps**

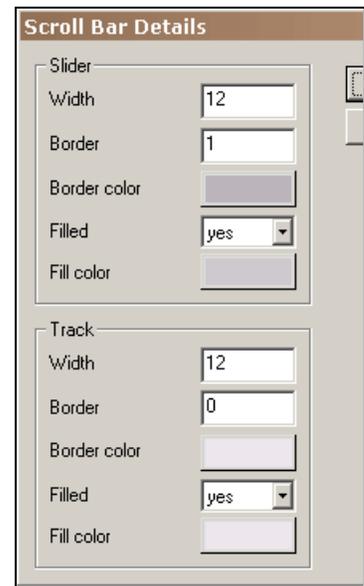
Same as in word processing – automatic line breaks are inserted when the text line length exceeds the edge of the text box.

**Has scroll bar / Scroll bar details**

The two properties enables the display of a vertical scrollbar in the dynamic text box, and open its parameters interface, respectively. If scroll bar is set, a vertical scroll bar is added on the right inside of the text field. The details of the look of that scroll bar is set in scroll bar properties. The first group box defines the properties of the slider and the second group box defines the properties of the track.



This is a scroll bar with customized settings. The border to the text field has been turned off and the borders to the slider and track have been set to 2. A no border text field is displayed with a faint border in the editing environment.



**Has html**

If Has Html is set the text can contain a subset of HTML formatting tags: <B>, <I>, <U>, <LI>, <FONT>, <P>, <BR>, <A>, <IMG>, <SPAN>, <TAB> and <TEXTFORMAT>.

## Effects Tab

Only one effect for dynamic text at this time.

### Filter Effects

The filter Effect is described in Chapter 11 – Movie Clips.



## Behavior Tab

Only one behavior for dynamic text at this time.

### Variable name

The Variable Name is editable and allows for easy dynamic changes to the text.

The text can be changed or retrieved while the movie plays using Set Dynamic Text, Load Dynamic Text, the Variable Name and Send Form Data actions. The use of Send Form Data requires knowledge of GET and POST data sending.

<http://www.pppmail.com/flash4cgi.htm> offers some information on this topic.



If you use & in the text, the Flash player interprets it as the start of a new variable. Replace the & in the text with %26 (Unicode mapping). For more details, see

[http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_14143](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_14143).



## Chapter 11 - Movie Clips

A Movie Clip is a movie within the main movie with its own canvas and timeline. Movie clips and the main movie have many commands and features in common. For example, a movie clip can be previewed with the internal player the same way a movie is previewed.

In object oriented programming, movie clips are used to organize visual elements and sounds, while the main movie is used to organize movie clips (among other objects). A clip within another clip is called a nested or child clip; a clip that contains another clip is the host or parent clip. The display properties of movie clips can be dynamically changed via ActionScript by the host or root timeline(s). Likewise, being movies in their own right, movie clips can contain frame actions and execute ActionScript within their own timeline, independent of or coordinated with actions on the main timeline or in other clips.

Movie clips can be copied and pasted like shapes, or manually added to the Symbol library.

Clips can be used to organize code, contain sound files, group shapes, emulate buttons (but with more capabilities), add behavior to objects, and act like Layers in a Cel Animation. All these capabilities make Movie Clips the building blocks for advanced animations. Almost all of these capabilities require a knowledge of ActionScript.

### Creating Movie Clips



Click on the triangle or click and hold to display the two ways to create a movie clip (these commands are also available under the Shapes menu):

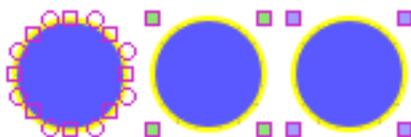
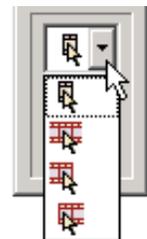


press the new movie clip button to create an empty movie clip, or



press the convert shapes to movie clip button to move the selected shape(s) or objects into a new movie clip.

In converting shapes to a movie clip, the selection scope -- this frame only, all frames, etc. -- determines the composition of the movie clip. If you select a number of shapes in the current key frame and select all frames for the selection scope, all those shapes in whatever frames they appear will be converted into a movie clip.

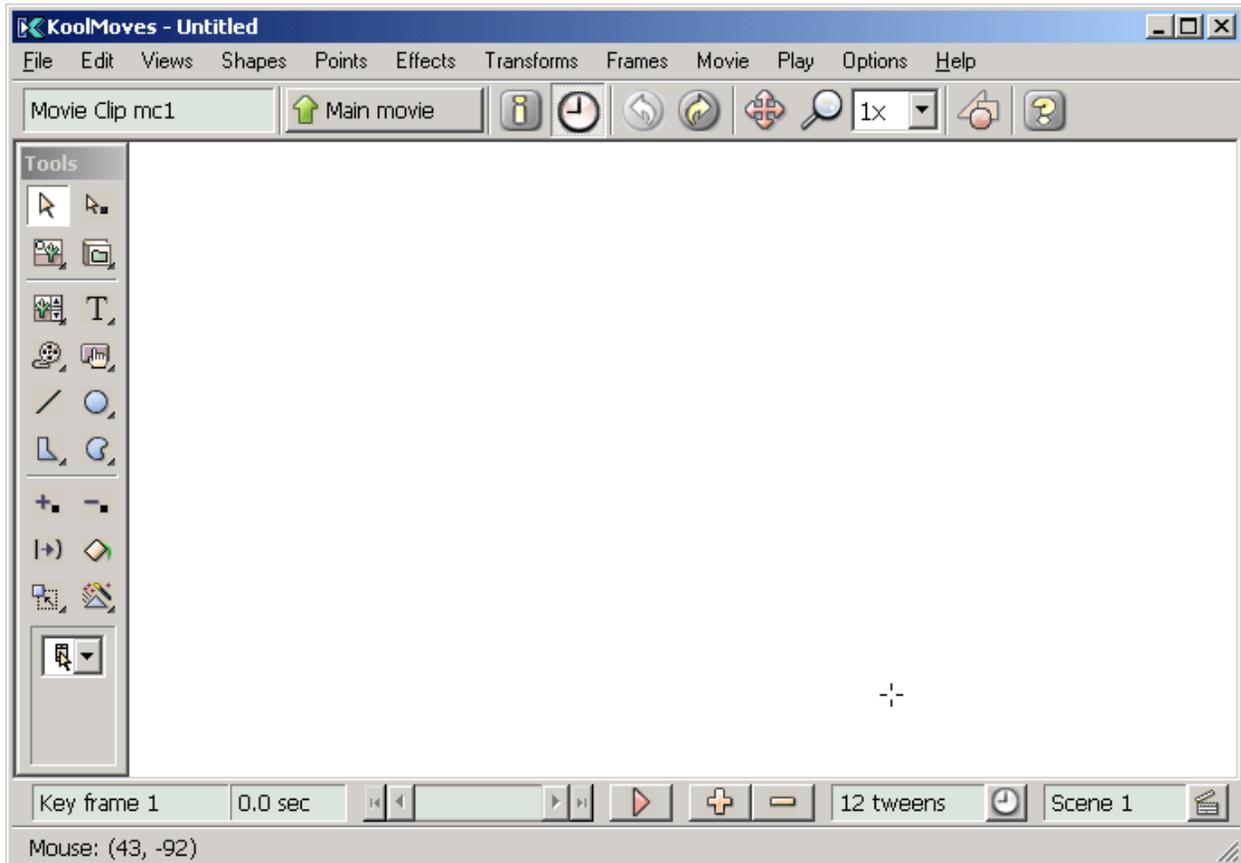


Shape Movie Clip Button

A selected movie clip symbol has a similar bounding box to a button except for the shading color in the corners.

## Movie Clip Timeline

Click on the Frames button at the top of the properties dialog to reveal the visual elements and frame actions of the movie clip. Within its own timeline, the movie clip has an environment analogous to that of the main movie and has over 90% of the same capabilities.



Within a button state or movie clip, use Options > View Parent Movie to see a semi-transparent overlay of the parent object's (`_root` in the example above) contents. To return to the `_root` level of the movie, click on the Main movie button.

## Movie Clip Properties Data View

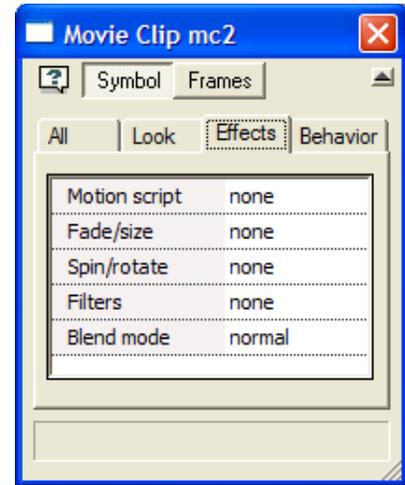
In Chapter 5, Section 1), we covered the Properties Data View and the properties of a shape. Not surprisingly, movie clips have a slightly different set of properties from simple shapes, but the GUI has them organized in much the same way. We will be looking at the Look, Effects and Behavior Tabs for a movie clip.

## Movie Clip Look

This tab is empty – the appearance of the movie clip is tied entirely to its contents, effects and runtime behavior.

## Movie Clip Effects

Movie Clips have in common with shapes the motion script, fade/size, spin/rotate effects. And while movie clips may not have 3D or Drop Shadow effects, these are quite easily duplicated, and surpassed, with the Filters and Blending Modes effects.



## Filters

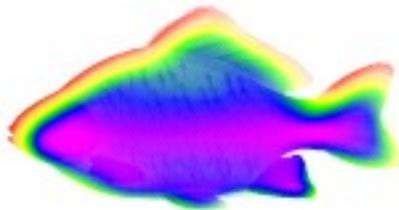
Flash 8 filters allow for the quick creation of compelling designs, and unlike imaging applications (like GIMP or Photoshop) that have traditionally been used to apply filter effects, in Flash the filtering can actually be animated. Because the filters are natively supported and rendered in real time by Flash Player 8, there is an associated processing cost; if your movie's performance is suffering, consider preprocessing static images before loading into your movie, and rethinking the parameters on your animated filter effects.

Filter effects can be applied to movie clip, button and dynamic text objects. Other object types, shapes, and imported images can be placed in a movie clip and filtered indirectly.

Available filters include:

- Adjust Color: Modify an object's Hue, Saturation, Brightness and Contrast.
- Bevel: Commonly used with buttons and user interfaces, beveling gives a sense of depth to an object.
- Blur: Define X and Y pixel offsets to smudge an object.
- Drop Shadow: Applies a shadow, and more if you wish.
- Glow: Creates a glowing outline around an object.
- Gradient Bevel: Same shape and appearance as Bevel, except a gradient is used.
- Gradient Glow: The glowing outline has a gradient color scheme.

The brief descriptions above don't do justice to this powerful KoolMoves feature. For details on how to use this feature, and some sample images, please see the KoolMoves Filters User Guide.



## Blend Mode Effect



Blend modes involve combining the colors of one image (the base or destination image) with the colors of another image (the blend or source image) to produce the final image on the screen. Each pixel value in an image is processed with the corresponding pixel value of the other image to produce a pixel value for that same position in the result, but actual math is performed on the component RGB values.

Blend modes can only be applied to movie clips or buttons, either through ActionScript or through the Blend Mode window (accessed via the Effects menu or the Properties Data View).

Available blend modes include:

- Normal: No blending at all, actually.
- Layer: A “composition” layer used to contain 1) a masking gradient and 2) an Alpha or Erase layer. By itself the layer mode does nothing, but used as directed it’s gradient provides a feathering mask effect applied to the destination image.
- Add: Source and Destination pixel color values are added – tends to lighten/brighten.
- Subtract: Source pixel color value is subtracted from destination pixel.
- Multiply: The two pixels are multiplied – resultant pixel is as dark or darker than original pixels.
- Screen: Opposite of the multiply mode.
- Difference: The darker of the two pixels is subtracted from the lighter.
- Darken: Combines the darker pixels of the source image with the destination image.
- Lighten: Opposite of the darken mode.
- Overlay: If the destination RGB pixel is less than 127, then multiply mode is applied, else screen mode is applied. In plain English, the source highlights and shadows are overlaid on the destination image.
- Hardlight: Same as Overlay, except source pixel is used
- Invert: The source image acts as a mask – overlapped destination image area is inverted.
- Alpha: Transparent areas of the Alpha mask block destination image.
- Erase: Opaque areas of the Erase mask block the destination image.

As with the filters, the brief descriptions above don’t do justice to this powerful KoolMoves feature. For details on how to use this feature, and some sample images, please see the KoolMoves Blend Mode User Guide.



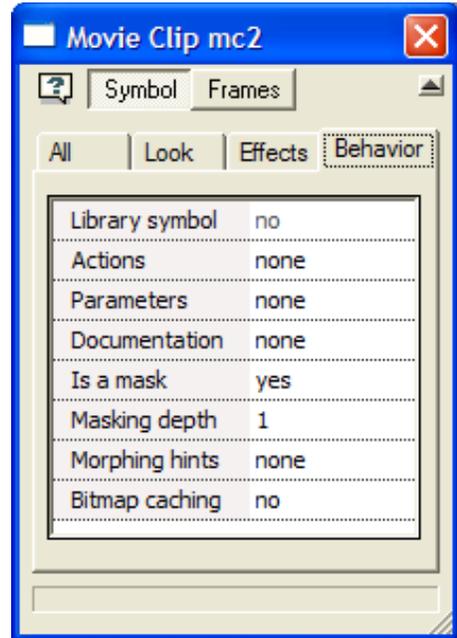
## Movie Clip Behavior

Movie clips are the workhorse object of the Flash environment, and the range and scope of the Behavior properties reflect their flexibility in handling a variety of roles and actions.

However, some of the behavior properties like library symbol and masking have been covered in other sections, so we will focus on the properties unique to movie clips.

### Movie Clip Actions

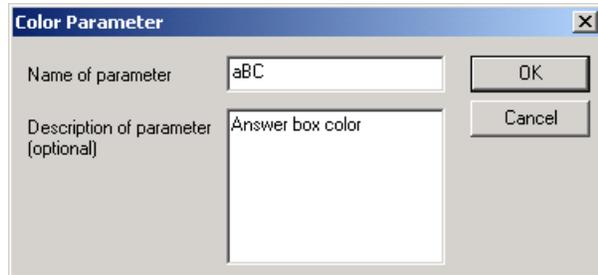
Click on the Actions field to open the ActionScript editor and define onClipEvents. These pieces of code travel with the movie clip they are attached to, and are not visible from the main timeline data views. If your movie is complicated and you can't recall where some code is located (which can happen when you use this field), use the Movie Overview data view to track it down.



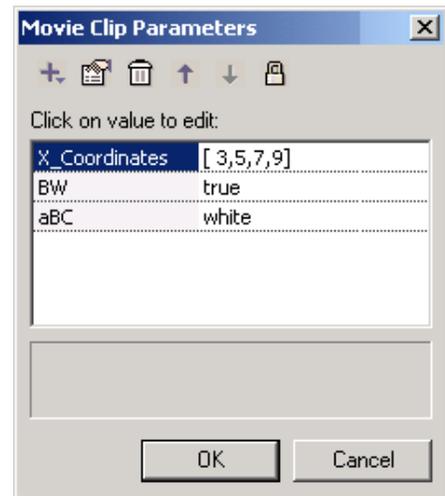
### Movie Clip Parameters

In the Properties popup, you can create and assign values for parameters to the movie clip. These parameter assignments are exported in the swf to the top of the first frame of the movie clip. Parameters offer the opportunity for reusable movie clips that can be customized by the end user.

 Click on the Add Parameter button to display the available parameter types. Selecting a type opens a dialog box.



Click on a parameter to change the value.



 Click on Edit to reopen the dialog box for the selected parameter.

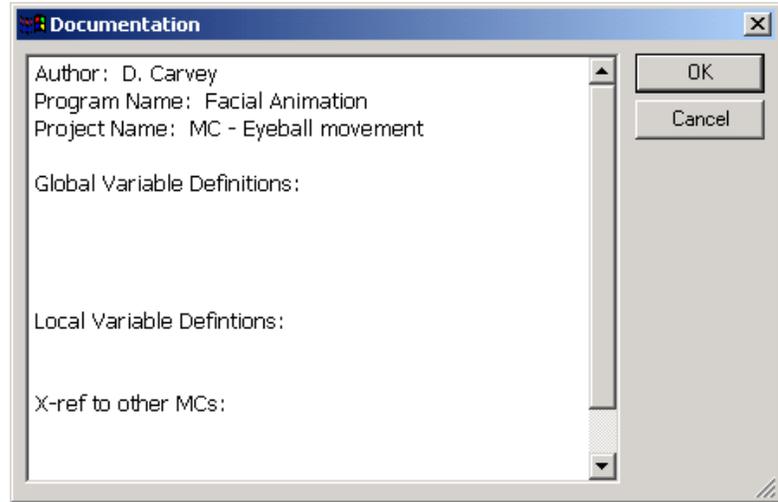
 Deletes selected parameter.

  Shifts selected parameter up or down in list.

 Click on Lock to prevent changes to the parameters (disables edit, delete, move up and down buttons).

### Movie Clip Documentation

A very simple text editing function allows you to add documentation/comments to your movie clip, enhancing its portability and maintenance. Use Ctrl-M to insert line feeds.



### Morphing Hints



Activate this parameter if the movie clip is going to be moving in a fashion other than pure rotation.

### Bitmap Caching

A major performance limiting factor has been Flash's requirement to update vector graphic images with each screen refresh, even if nothing has changed. In movies with many objects on the screen, this brought down the practical frames per second performance capabilities. Bitmap caching is intended to reduce the load on the renderer by storing copies of selected movie clips in the Flash Player.

Bitmap caching has been implemented as a movie clip property. This property can be accessed through the Properties Data View.

Bitmap caching is also accessible in ActionScript through the `mc.cacheAsBitmap` property (true = caching turned on, false = caching turned off). You can also directly access this property to determine a movie clip's bitmap caching status. Bitmap caching also works with nested movie clips.

The Flash Player will convert the visual contents of movie clips with bitmap caching on into bitmaps and store them in runtime Flash Player memory. These copies are used for screen

refreshes instead of rendering the images from scratch, until the contents of the movie clip change, at which time a new bitmap copy is rendered and stored.

So as long as the movie clip doesn't change, caching can seriously reduce the load on the renderer, especially if the movie clip happens to contain complex vector images (lots of lines, curves, gradient fills, etc.). However, what constitutes a change? If the movie clip moves up or down or sideways – no problem. On the other hand, rotations, scaling, flipping, animated filters, basically any effect, will require the movie clip to be rendered. For you ActionScripters, these are the properties that cause the bitmap to be regenerated:

- `_xscale`
- `_yscale`
- `_rotation`
- `_alpha`
- `_width`
- `_height`
- `filters`
- `blendMode`
- `opaqueBackground`
- `transform`

It follows that, since two steps are involved (rendering and storing) in caching as opposed to only one step without caching, and one of the steps is tying up memory (you're storing both the vector data and the bitmap), that turning bitmap caching on unnecessarily for a movie clip that needs to be rendered frequently can actually hurt movie performance. And it does.

There are a few limitations, quantitative, qualitative, and performance inherent in bitmap caching:

- The largest bitmaps that can be stored are 2,880 by 2,880 pixels,
- Bitmaps do not store vector images as efficiently as vector data,
- Bitmap caching is turned on when a filter is applied to a movie clip, and
- Bitmap caching is turned off when content (external Flash movie or image) is dynamically loaded via ActionScript.



## Chapter 12 – Transforms & Effects

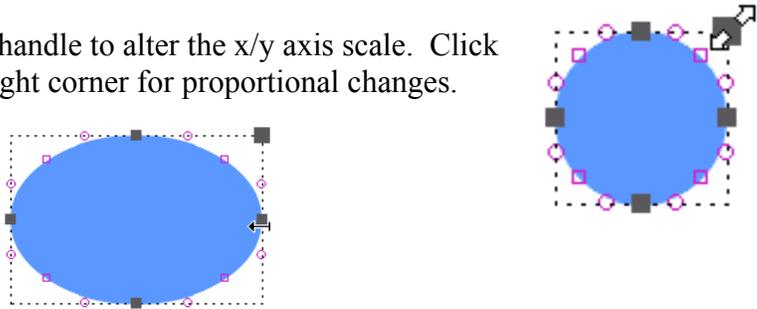
Toolbox Effects are covered in Chapter 5, Data Views > Properties Data View > [Effects](#) .

### Toolbox Transforms

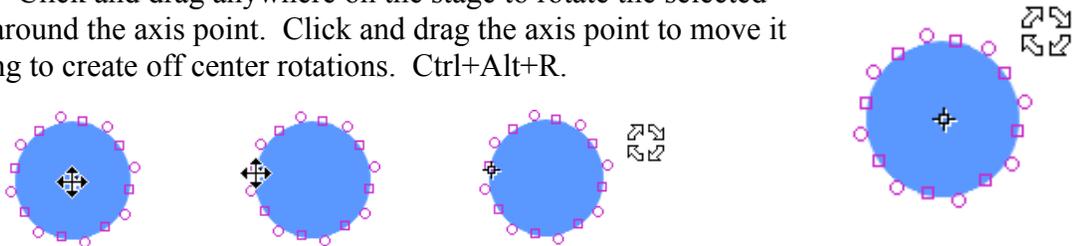
 Click and hold to open the Transform tool drawer. With the exception of the flip, squish, and perspective tools, these tools were covered in Chapter 6 in the [Reshaping](#) entry. These tools operate in the same or similar fashion, so we will just provide a brief description of the tools here.



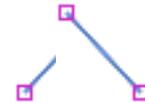
 Scale/Resize. Click and drag a handle to alter the x/y axis scale. Click and drag the handle in the top right corner for proportional changes. Ctrl+Alt+S.



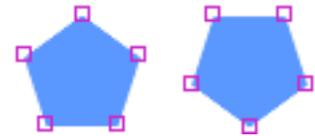
 Rotate. Click and drag anywhere on the stage to rotate the selected shape around the axis point. Click and drag the axis point to move it before rotating to create off center rotations. Ctrl+Alt+R.



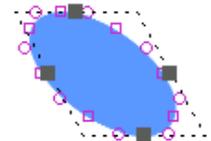
 Flip Horizontal. Turns the selected object into a mirror image along a vertical axis. Ctrl-Alt-H.



 Flip Vertical. Turns the selected object into a mirror image along a horizontal axis. Ctrl-Alt-V.



 Slant. Click and drag a handle to distort the selected object in a diagonal fashion. Ctrl+Alt+T.



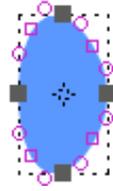
 Squish. Click and drag a handle to selected object in an oblong fashion.



compress or stretch the Ctrl+Alt+Q.



Perspective. Click and drag a handle to change the object's shape consistent with the effects of distance on different points on the object. Ctrl+Alt+P.

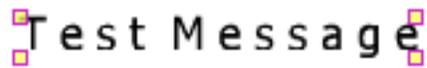


### ***Transforms Menu (other commands)***

Following are the Transforms Menu commands not already covered in [Toolbox Transforms](#).

#### **Break Apart Text**

The Break Apart Text commands allow you to take static text and convert it into either individual characters or shapes defined with points. The first step is to select your static text object,



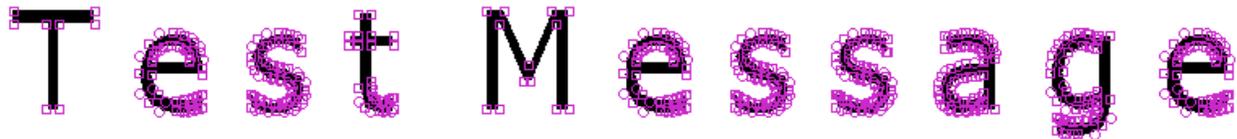
then select the appropriate menu command.

#### **Break into Text Characters**



Each letter in the test message has been broken out into its own static text object. This means that each character can be selected and modified individually, with its own effects, parameter settings and position.

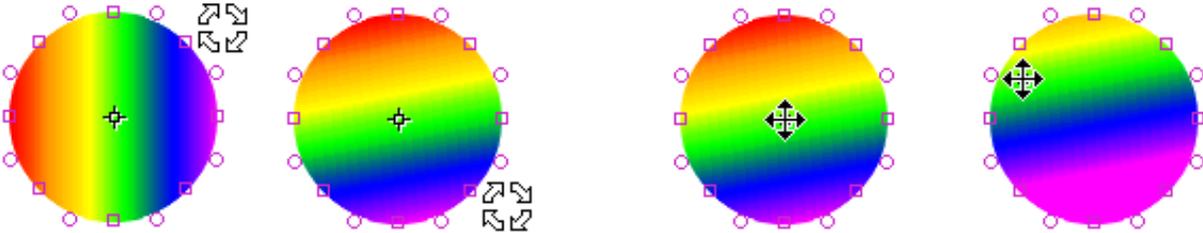
#### **Convert to a Shape with Points**



Oddly enough, all the letters above are considered one shape. For example, clicking on one of the letters above selects the entire message. Use the Select Point tool to edit the placement of the individual points.

### Rotate/Shift Gradient Fill

This command allows you to rotate and/or shift the gradient fill of a shape.

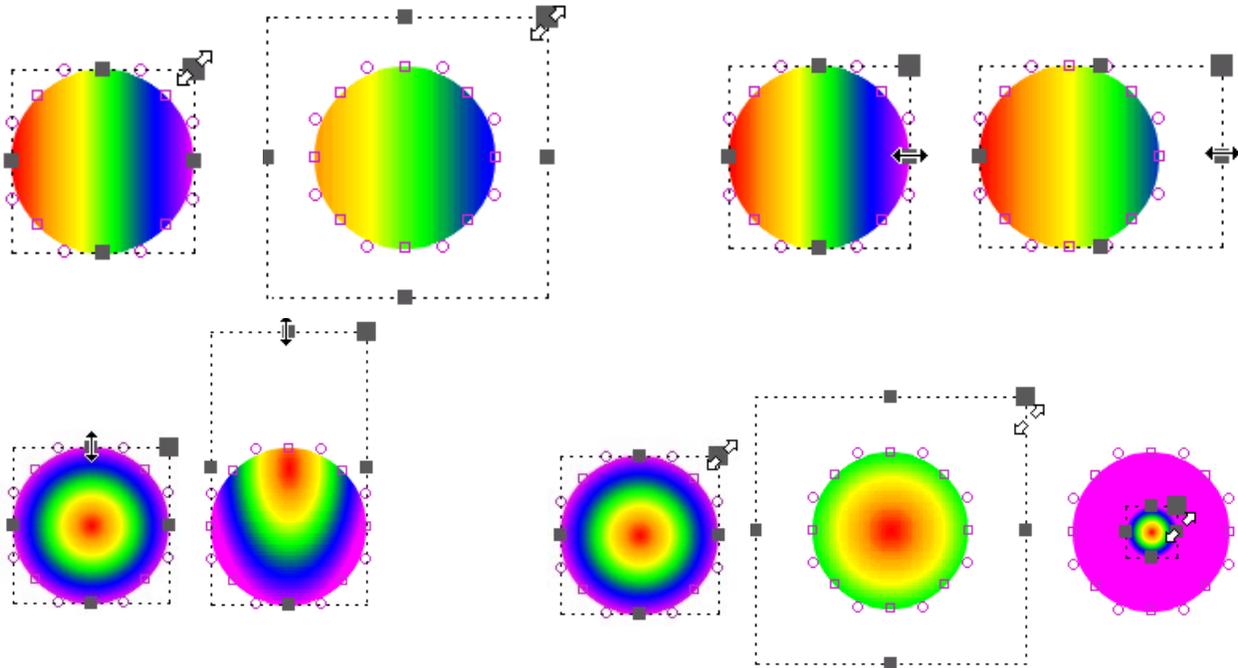


A reference point is placed at the center of the selected shape and the mouse pointer is changed to four arrows arranged in a circle. Click and drag the mouse around the center to rotate the gradient fill.

Placing the mouse pointer over the center will change the mouse pointer again, this time into four arrow pointing out from the center. Click and drag the center to shift the gradient fill.

### Scale Gradient Fill

This command resizes the gradient fill without affecting the shape's size. Click and drag the handles as with the normal scale tool.



### Rotate Image Fill

Analogous to Rotate Gradient Fill, but applied to shapes with image fills.

### Shift Image Fill

Analogous to Shift Gradient Fill, but applied to shapes with image fills.

### Scale Image Fill

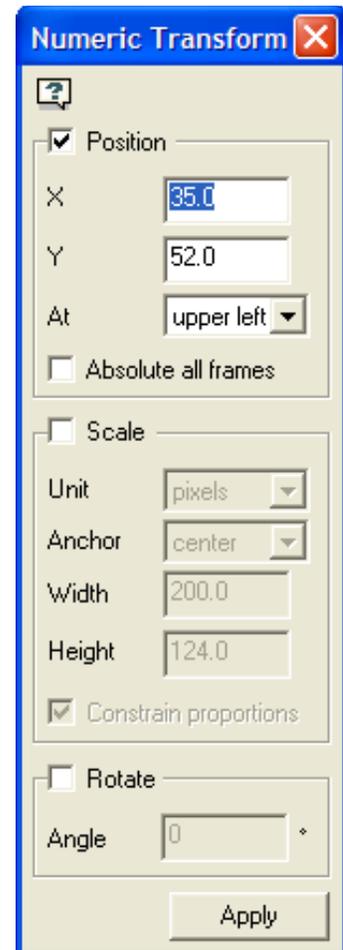
Analogous to Scale Gradient Fill, but applied to shapes with image fills.

### Numeric... (Ctrl+Alt+N)

Use the Numeric Transform tool to modify a selected object's size, initial rotation and position. Click on the Scale, Rotate or Position boxes to activate those fields – only one set of fields is active at a time. Click on the Apply button to effect changes to the selected object – only one set of changes (Scale/Rotate/Position) can be applied at a time.

- Scale Units - Pixels or Percent.
- Scale Anchor Points – Center, Upper Left, Upper Right, Lower Left, Lower Right, Top, Bottom, Left, Right.
- Check the Constrain proportions box to maintain original width to height ratios.
- Angle field only accepts integers.
- The X and Y position fields are used to set a coordinate point on the selected object relative to the stage.
- Position coordinate point choices are Upper Left, Upper Right, Lower Left, Lower Right and Center

Set the Anchor field prior to changing the X and Y fields.



### Center (Ctrl+Alt+C)

This command moves the selected object to the center of the stage.

### Rotate Left 90°

This command rotates the selected object 90° to the left

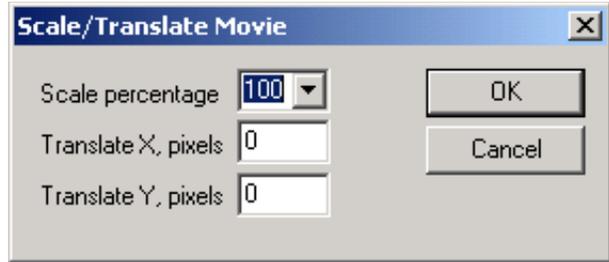
### Rotate Right 90°

This command rotates the selected object 90° to the right

### Scale/Translate Movie...

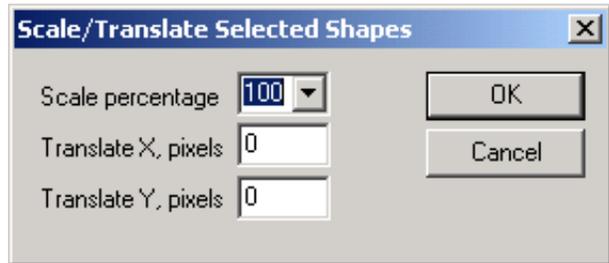
Use this window to

- Scale all objects within a movie using the drop down list of ratios (30% to 300%, in increments of 10), and
- Shift all object within the movie the designated number of pixels.



### Scale/Translate Selected Shapes...

Same as Scale/Translate Movie..., except applied only to the selected shapes.



### Toolbox Effects

 Click and hold to open the Effects tool drawer. These tools were covered in Chapter 5 in Data Views > Properties Data View > Effects, so we will just list the tools with their icons.



 Motion Script

 Filter

 3D

 Blend Mode

 Spin / Rotate

 Drop Shadow

### Creating Your Own Transition

In Chapter 5 while discussing the Effects tab, several built-in transition effects were discussed (recapped below). It is easy to create your own transitions using the tools discussed in this and preceding chapters. The basic principle is to create the stages of the transition in a series of key frames. This example illustrates an image that comes from the left into the movie frame as a small transparent image and becomes more opaque and larger as the image moves to the center of the movie frame.

 In step 1, an image is imported using File > Import Graphic Image. To start the transition, the image is dragged to the left of the movie frame.

  In step 2, the image is made small and transparent using the scale and paint bucket tools.

 In step 3, the current key frame is copied and added to the end of the movie.



In step 4, the image is dragged to the center of the movie frame in the last key frame.



Finally in the last key frame, the paint bucket and scale tools remove the transparency and return the image to its original size.

When the movie is played, the image slides in from the left from a transparent, small state. The transitions you can create are endless.

KoolMoves has 6 built-in transitions -- initial fade in, final fade out, initial grow from nothing, and final shrink to nothing, initial zoom in and final zoom out. They take effect at the beginning or end of the shape's life in the movie depending on whether the transition is called initial or called final. These transitions can be set in the Properties popup for the selected shape or through the Effects menu bar. Transitions occur over a set number of key frames. When a transition is added to a shape, if there aren't enough frames for the transition to occur, frames will be added if the "Add frames if needed" check box is checked. Transitions can be combined, although the fade transitions are not a good combination with drop shadow.

## **Part IV – Advanced/Additional Features**

**Chapter 13 – Action Scripting**

**Chapter 14 – KoolMoves Components**

**Chapter 15 – Symbol & Link Library**



## Chapter 13 - Action Scripting

The Action Script Editor (ASE) is KoolMoves' ActionScript programming interface.

### Opening the Action Script Editor (ASE) for the first time

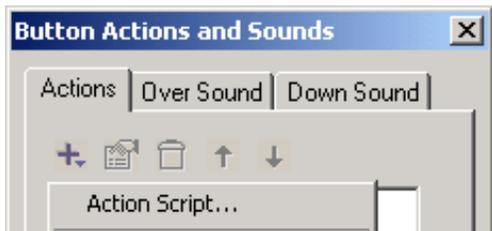
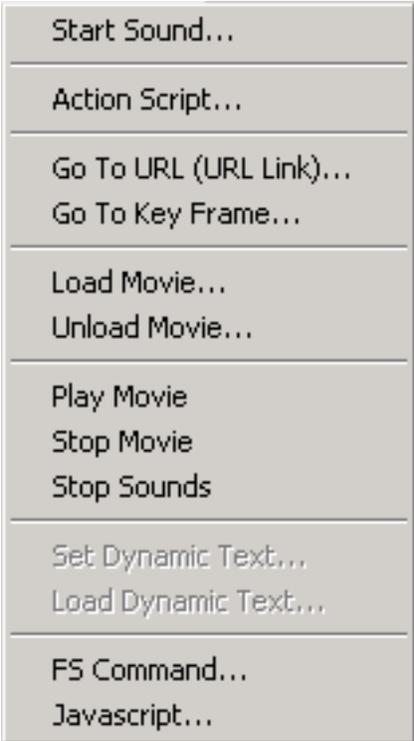
How the ASE is accessed will depend on where the ActionScript is being placed. For example, to place code on a frame in a timeline click on "ActionScript..." in the popup menu for either the Actions and Sounds Data View or the Actions and Sounds Overview window.

- 1) 
  - 2) 
  - 3) 
- 1) Click on the Score/Timeline Menu Toolbar icon to open the Score/Timeline Data View, 2) select the Actions and Sounds tab, 3) click on the popup menu button, and then \*) click on ActionScript...

\*)

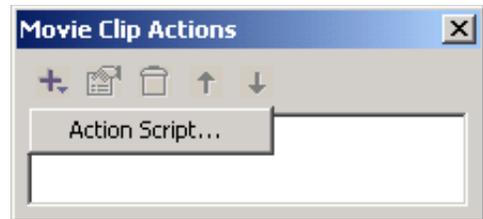
or

- 1) 
  - 2) 
- 1) Click on the Actions & Sounds Overview Menu Toolbar icon to open the Actions & Sounds Overview, 2) click on the popup menu button, and then \*) click on ActionScript...

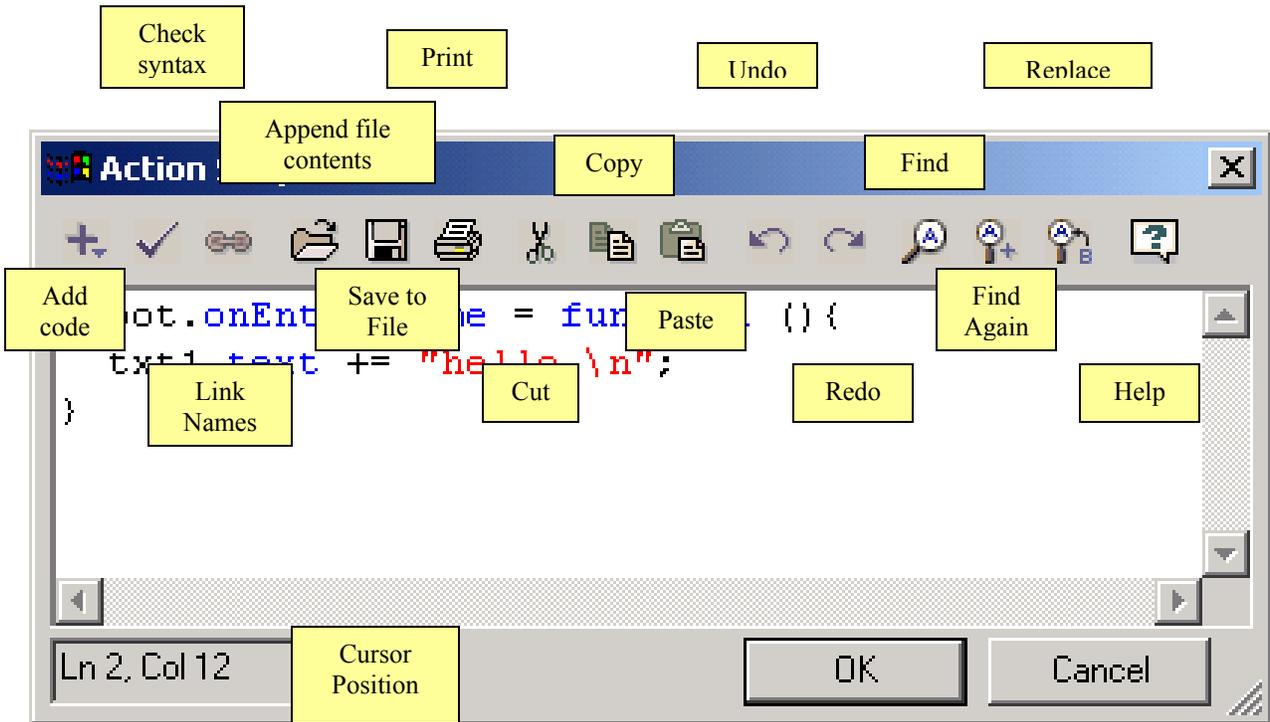


To place an event handler (`on(event) {}`) on a button, select a button and in the Properties window click on Actions, click on the plus button and then click on ActionScript in the popup menu.

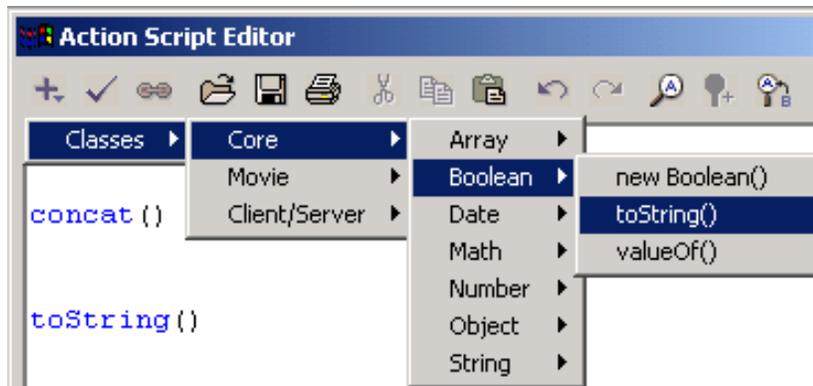
To place an event handler (`onClipEvent(event) {}`) on a movie clip, select a movie clip and in the Properties window click on Actions, click on the plus button and then click on ActionScript in the popup menu.



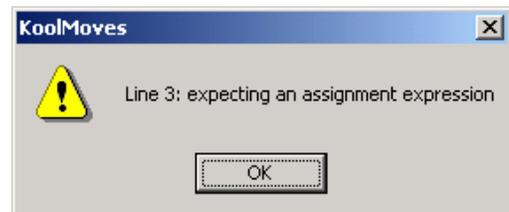
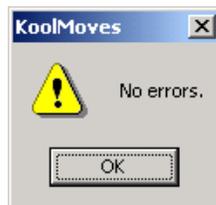
## The Action Script Editor (ASE)



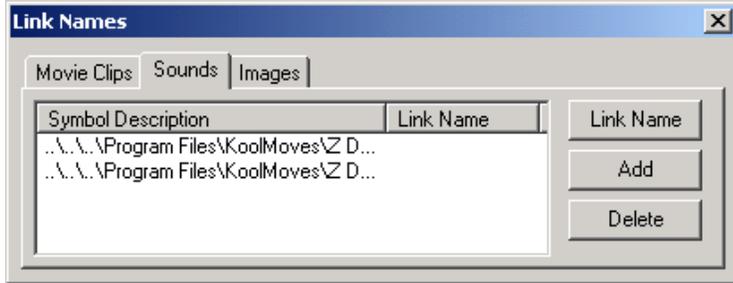
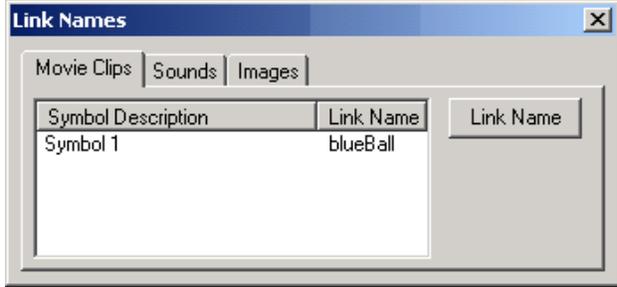
 Click on the plus button to open a popup menu of available reserved ActionScript words and phrases that can be added to the current cursor position.



 Clicking the checkmark button will return one of two messages after checking the code for syntax errors.

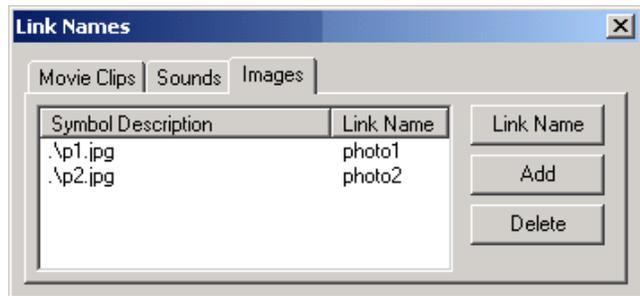


 The link button opens the Link Names window. In the Movie Clips tab are displayed available symbol library movie clip links created in the main gui with the Symbol Library interface. These links work with the mc.attachMovie() action script command.

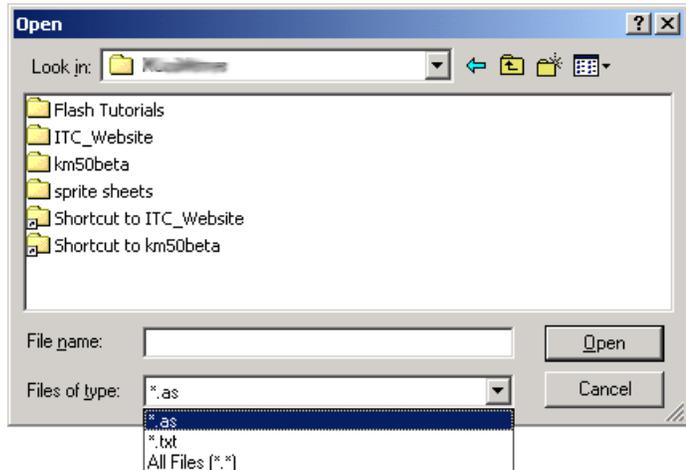


Under the Sounds tab, links to external sound files can be created. These links can be used with the sound.attachSound() action script command and the KoolMoves [Slideshow](#) component.

Under the Images tab, links to external image files can be created. These links can be used with the loadBitmap() action script command and the KoolMoves [Slideshow](#) component.

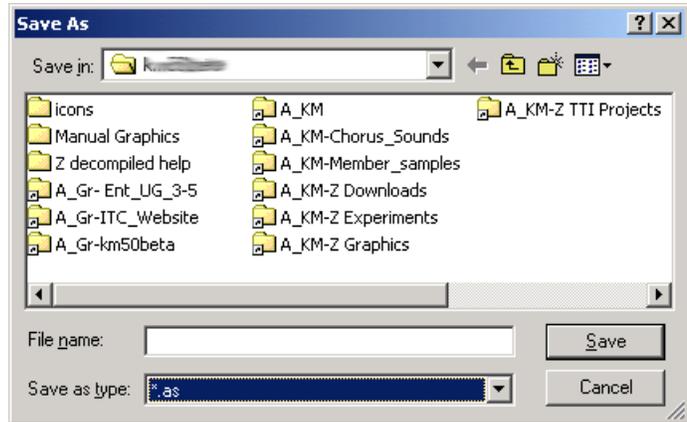


 Open a text file and paste the contents to the current cursor position. Default file extension is as.





Save contents of coding area to a text file. Default file extension is as



Print contents of coding area.



Cut, Copy, and Paste, respectively. Works with Windows™ clipboard.



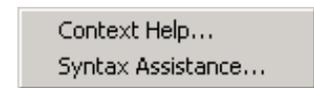
Undo and Redo last action, respectively.



Find, Find Again, and Replace, respectively. Works with Windows™ clipboard.



Popup Help menu with Syntax Assistance and Context Sensitive tips.

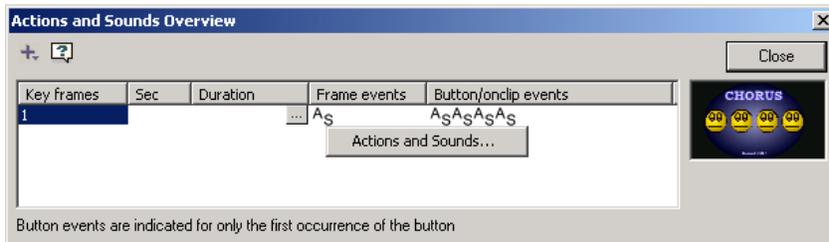
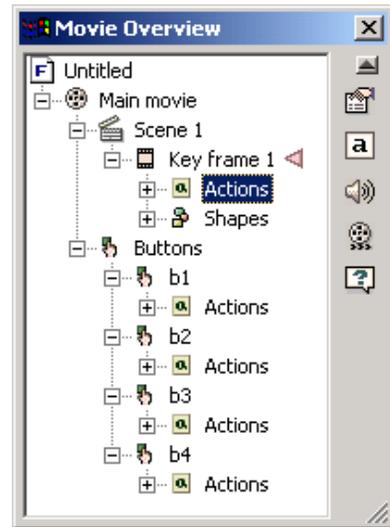
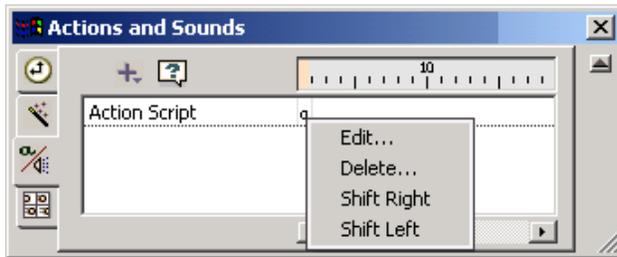


Click on Cancel to exit without saving changes. Clicking OK will return KoolMoves to the main movie environment if there are no syntax errors; if there are errors, you have the option to correct the errors or save the ActionScript code in its current state.



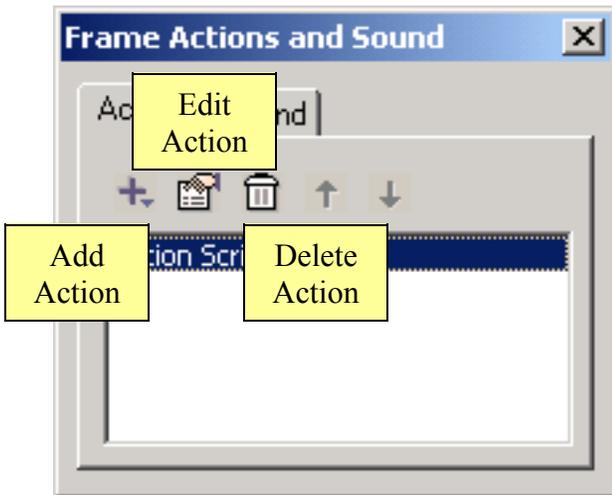
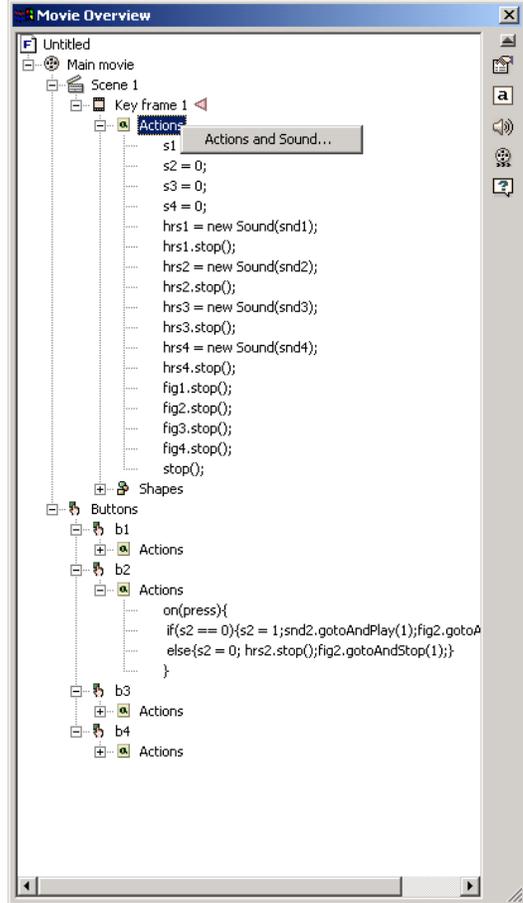
## Editing ActionScript code

Editing event handlers placed directly on buttons or movie clips through the Properties window is the same process as initially creating them. However, editing code that has been placed on a frame differs from the initial creation process, as KoolMoves Allows for multiple independent pieces of code or commands to exist on the same frame. (However in the sample screenshots below, there is only one piece of code on the Frame). The code to be edited must be selected by right clicking on it and then navigating the popup window. There are three ways to find the code to edit: the Action and Sounds view in the Score window, the Actions and Sounds Overview window, and the Movie Overview window. The last two methods also displays code placed directly on buttons and movie clips.



Clicking on the “+” next to an Actions entry in the Movie Overview window will expand the entry to display the code.

In the Actions and Sound Overview or the Movie Overview, after clicking on the popup Actions and Sound..., click on the edit symbol in the Frame Actions and Sound window to edit the highlighted ActionScript object.



## ***Unimplemented ActionScript***

ActionScript 2 (syntax and constructs) is unimplemented in its entirety.

Unimplemented Flash Player 4 through 6 commands:

```

%=      (modulo assignment)
&=      (bitwise AND assignment)
|=      (bitwise OR assignment)
~       (bitwise NOT)
<<=     (bitwise left shift and assignment)
<>      (inequality)
>>=     (bitwise right shift and assignment)
>>>=   (bitwise unsigned right shift and assignment)
    
```

switch - case conditional

\* a workaround for switch/case is to rewrite the code with if/else if

do - while loop

nextScene(), prevScene()

super

\* advanced command for use with the symbol library.

void

\* not used very often. Void(x) returns 'undefined'.

throw, try, catch, finally

\* used for error handling.

trace

\* used to output a result for debugging.

The following invalid operators have been deprecated since Flash Player 5 and have never been implemented in KoolMoves. If you are following an old ActionScript tutorial / language reference, please use the identified alternative operators.

Invalid	Valid	Invalid	Valid	Invalid	Valid
add	+	and	&	or	
gt	>	ne	!=	<>	!=
lt	<	not	!		

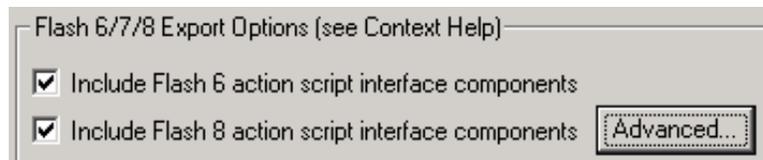


## Chapter 14 – KoolMoves Components

Created by a senior member of the KoolMoves community, the components in KoolMoves are portable object classes written in ActionScript. Components are used to expand the core capabilities of the Flash language, and can add a new dimension of interactivity when used in animations.

Technically subclasses of movie clips, they are handled in largely the same way. For example, implementing the action associated with an Interface Component is analogous to setting up a clip event. Likewise, many movie clip properties and methods work with components.

The code for the components are stored in library files, and are automatically added to a movie if components are added through the Toolbox. If the components are being added dynamically through ActionScript coding, you will need to set the inclusion of the library files manually. Check the appropriate box(es) in Export Settings and the compiler will then know to add the appropriate components code. For example, when you movie has Flash 6 components, check the "Include Flash 6 action script interface components (list box, etc.)" box. There is a similar box for Flash 8 components.



### ***Application Programming Interfaces (APIs)***

APIs allow you to change a component's parameters interactively or to initialize the component if it has been attached to the movie via the symbol library. Many components have advanced features that can only be accessed through their API.

Some of the components were written in Flash 6, some were written in Flash 8, and some components have both Flash 6 and 8 versions. The components were coded using the latest available features at the time of their respective development. This means that:

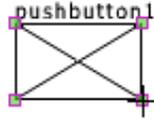
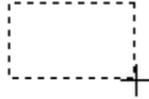
- components with two versions will have separate APIs for each version,
- the Flash 6 and Flash 8 API descriptions have different formats,
- The Flash 8 contentPane, listBox and comboBox components can control vertical scrolling via a mouse wheel (while the mouse is over the component), and
- the Flash 8 components were designed with and are available in the Symbol Library.

There is no conflict between the Flash 6 and 8 versions – your movie can have components of both versions. However, there are distinct differences in how you work with the two versions dynamically. For example, while all components can be added to a movie dynamically with ActionScript, Flash 6 component APIs have the `mc.createKComponent` function, while Flash 8 components use the standard ActionScript `mc.attachMovie` function. Applicable examples can be found in `\KoolMoves\Examples\Components\Flash N\`, where N is the flash version.

## Interface Components in the Toolbox



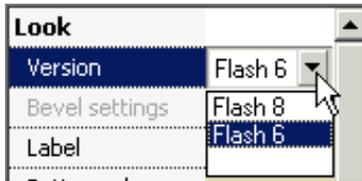
Click and hold on the Components tool to display the available components. Interface components allow movies to interact with users. KoolMoves has seven interface components, six of which can be added through the Toolbox. Example files included with the KoolMoves installation package (referenced earlier) introduce the minimal knowledge of ActionScript needed to use these components.



After selecting a component, the mouse pointer changes to a cross hair; click and drag to set the size and location of the component. Most components have a minimum size, while the check box defaults to 12 by 12 pixels (but the

check box can be resized with ActionScript coding).

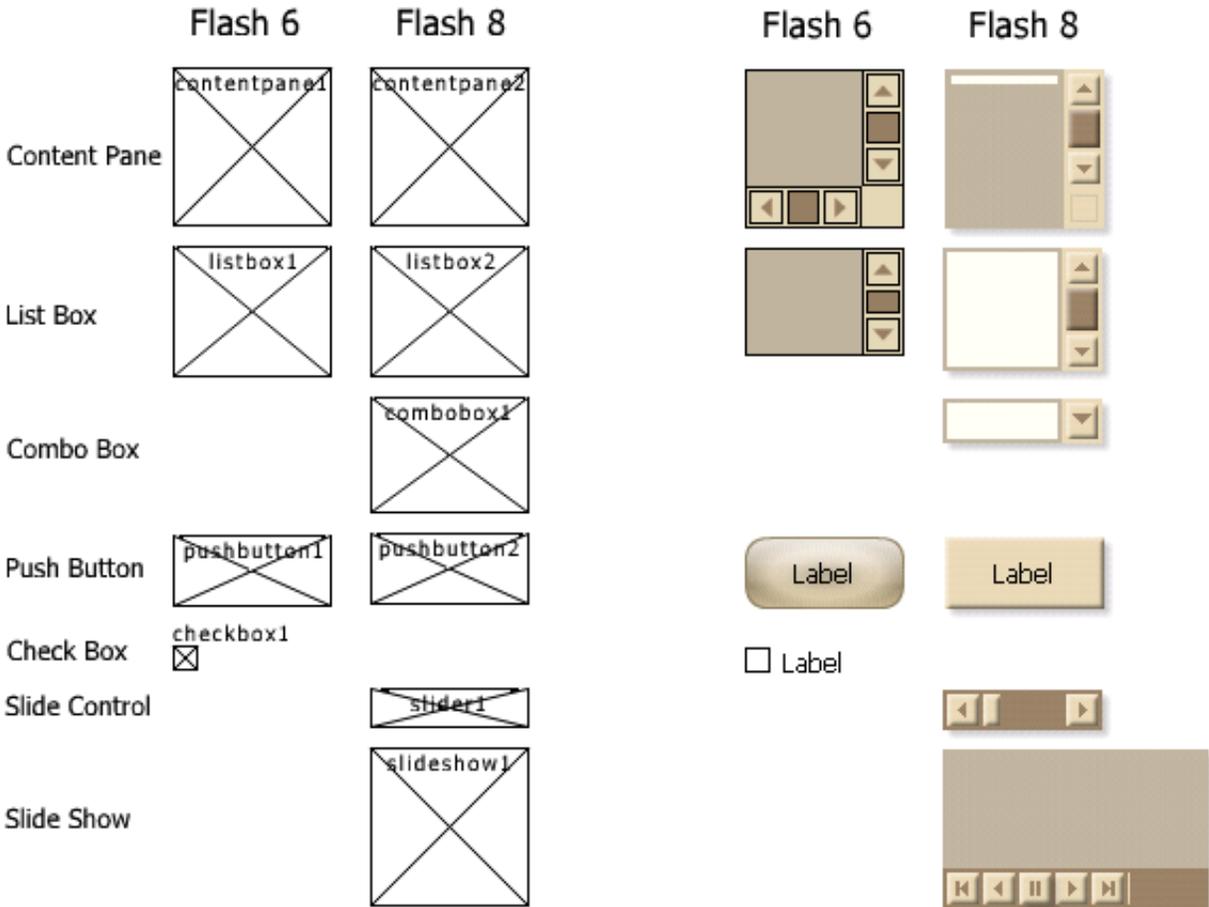
Other than resizing, moving or aligning actions, components added through the Toolbox can not be manipulated (i.e., no shape transforms or effects) at author time. The version, contents, parameters and actions associated with these components are defined in the Properties window at author time, and most of these can be changed dynamically with ActionScript through the Flash 6 APIs (Flash 8 components would need to be recreated from scratch).



same movie.

Once a component has been sized and placed on the stage, open the Properties window and set its properties. Some of the components have two versions (Flash 6 and 8) The Flash 8 version of components have fewer appearance settings, but allow for beveled borders. Both versions of a component can be used in the

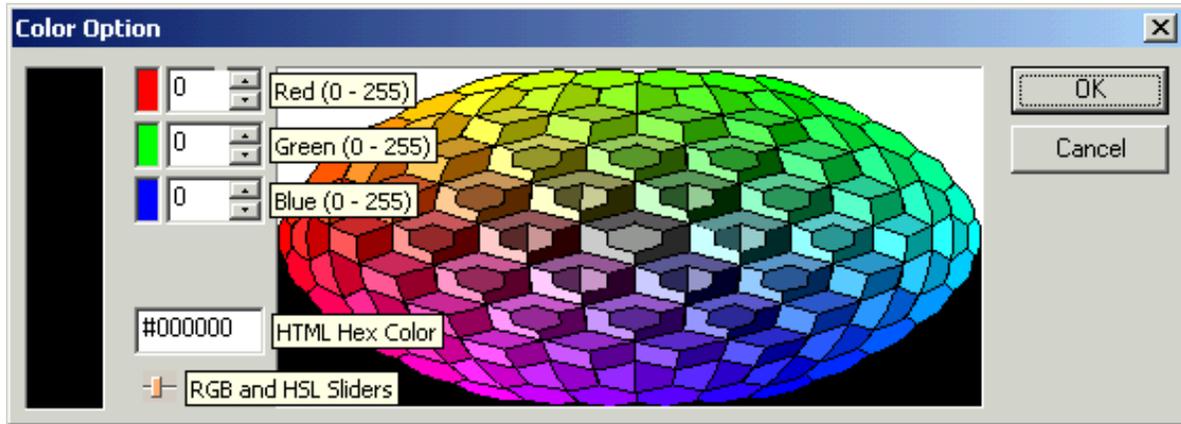
Below are typical components as displayed on the stage and in a web browser.



In terms of the Properties data view, the biggest difference between components is in the content handling and Appearance options. The more common parameters and their values are covered below.

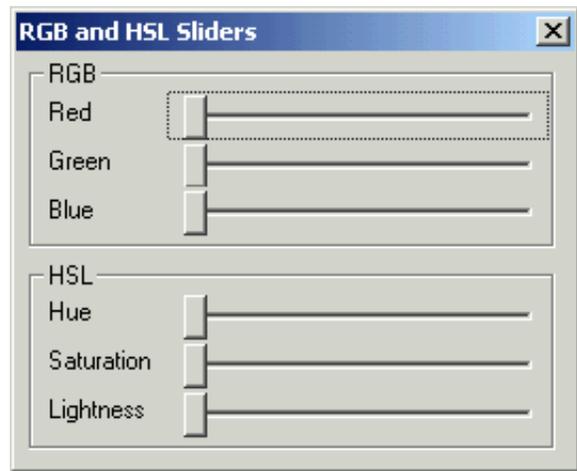
### Color Dialogs

The Arrow, Background, Border, Button, Scroll Bar, Scroll Track, Selected Text, Selection Bar, Text Background, and Text Color fields all have the same color picker. There are three ways to select a color from the initial Color Option window: 1) Enter a value between 0 and 255 for the Red, Green and Blue values, 2) Enter a value between #000000 and #FFFFFF in the Hex Color box, or use the Color Picker to select a color from the palette. 



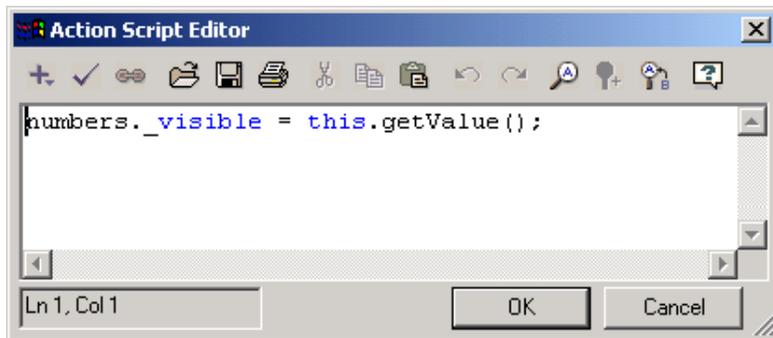
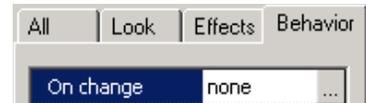
 Click on the RGB/HSL icon to open the RGB/Hue/Saturation/Lightness Slider window (a fourth way to select a color).

The color dialog is applied to control surfaces and does not allow for the use of patterns or images as fills, nor does it have a transparency setting. These settings can be dynamically changed with ActionScript for Flash 6 components.



### Behavior Actions & Dynamic Components

Most of the interface components have a default event (on change, on mouse click or on mouse drag) that can be used to execute ActionScript code. In the Properties window, after selecting the event under Behavior, click on the ellipsis (“...”) to open the ActionScript Editor.



The code embedded in a component is not limited to updating the movie as to the changes made to the component, but can also be used to control the flow of the movie.

Code is present on a component when the event field has changed from “none” to “action script”.



**Note:** When coding behavior for components through the Properties window, the event (e.g., onChange) has already been defined and a function block does not need to be specified. Also note that initializing a property not present in the Properties window is not complicated. In \KoolMoves\Examples\Components\Flash 6\Other examples\TellingTime.fun, checkbox1 (created at author time) is dynamically initialized with:

```
checkboxbox1.setValue(true);
```

**Note:** As mentioned earlier, Flash 6 and 8 components differ internally. Because Flash 8 components were designed with the use of the Symbol Library, their behavior code is placed within the component. This means that when dealing w/ external objects, the component's behavior code has to allow for its relative starting position. Flash 6 component behavior code, on the other hand, is positioned relative to the parent movie clip. In the sample code below, both components update a text box on \_root when their value changes.

Valid code for checkBox1 (Flash 6 component):

```
txt1.text += this.getValue();  
    or  
txt1.text += checkbox1.getValue();  
    or  
_root.txt1.text += _root.checkbox1.getValue();
```

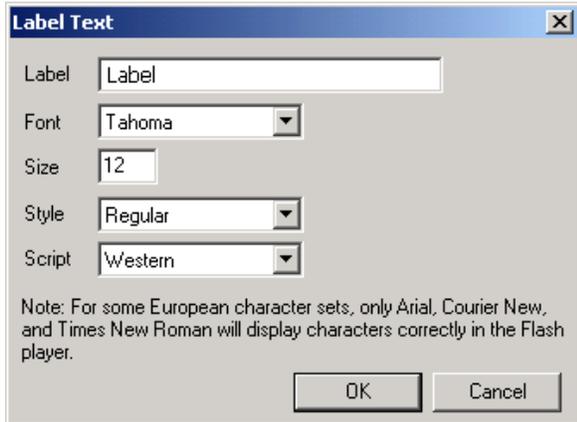
{Since the Flash 6 component behavior code is positioned relative to the parent movie clip, in this case \_root, both the code and the text box are on the same level and relative path information is not required.}

Valid code for comboBox1 (Flash 8 component):

```
_root.txt1.text += this.getValue();  
    or  
_root.txt1.text += _root.combobox1.getValue();
```

{The Flash 8 component behavior code's parent is the component and not the creating movie clip \_root, so even though the absolute path for the two components are analogous and the getValue() function look the same, the internal workings differ. For this reason, "combobox1.getValue();" returns "undefined". Likewise, sending the value to "txt1.text" doesn't work for Flash 8 components as the behavior code ends up looking for txt1 on the component object.}

**Label**



Labels are available for Push Buttons and Check Boxes. The fonts are limited to the fonts installed on your computer. Push button text is centered on the button while check box text is left justified to the right of the check box. Both are limited to one line.

Label settings can be changed dynamically with ActionScript for Flash 6 components.

**Bevel Settings**

Beveling is only available with Flash 8 components. The default values are an angle of 45 degrees and a width of 2 pixels. The range of valid values for angle and width are 0 to 60 degrees (in increments of 15 degrees) and 0 to 5 pixels, respectively. These settings can be changed with ActionScript.



		<b>Angle - Degrees</b>		
		30	45	60
Width - Pixels	0	Label	Label	Label
	2	Label	Label	Label
	5	Label	Label	Label

### Gradient, Liquid and Shaded (GLS) Styles

The Gradient, Liquid and Shaded Styles are only available with the Flash 6 components.

- Gradient style = Yes, the affected interface element is lightest at the top left corner and darkest at the bottom right corner.
- Shaded style = Yes, the component's border is replaced with a beveled edge.
- Liquid Style = Yes, a borderless, radial gradient is applied to the interface element (tends to overwhelm the Gradient and Shaded styles when applied to the same element).

		Gradient Yes	Gradient No
	Shaded	Yes <input type="checkbox"/> Label	<input type="checkbox"/> Label
	No	<input checked="" type="checkbox"/> Label	<input checked="" type="checkbox"/> Label

Check Box.

In the List Box you can see the Liquid style overriding the others on the scroll bar appearance.

		Gradient Yes		Gradient No	
		Shaded Yes	Shaded No	Shaded Yes	Shaded No
Liquid	Yes				
	No				

Ditto in the Content Pane.

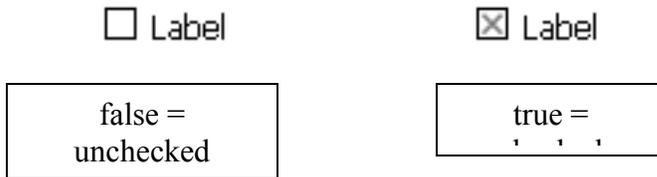
		Gradient Yes		Gradient No	
		Shaded Yes	Shaded No	Shaded Yes	Shaded No
Liquid	Yes				
	No				

The Push Button has only one visual component, so the Liquid style dominates.

		Gradient Yes		Gradient No	
		Shaded Yes	Shaded No	Shaded Yes	Shaded No
Liquid	Yes				
	No				

## Check Box

 Flash 6. Check boxes are best suited for turning features on or off and dealing with variables that have only two values.



The check box component was largely covered when it was used as the example in the section on Behavior Actions & Dynamic Coding. In addition to the examples already covered, check boxes are also used in `\KoolMoves\Examples\Components\Flash 6\Styling demo\KcStyling.fun` and `\KoolMoves\Examples\Components\Flash 6\Tool Tip example\tipstest.fun`.

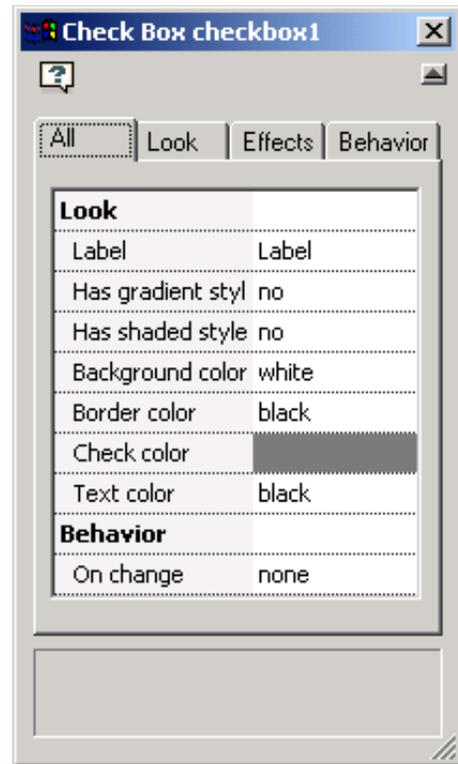
Below is the programming interface for the check box component.

### KCheckBox API (Flash 6):

```

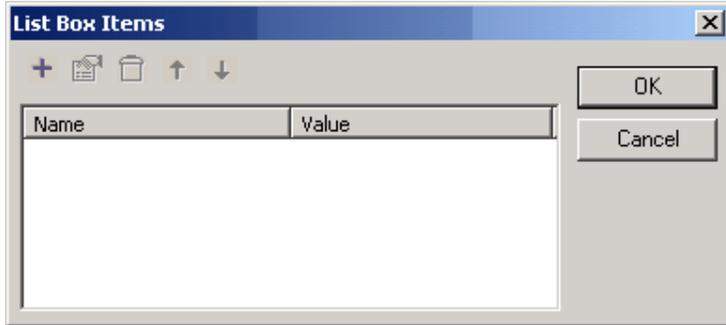
MovieClip.CreateKCheckBox (instanceName, depth, initObject);
KCheckBox.destroy ();
KCheckBox.setEnabled ();
KCheckBox.getLabel ();
KCheckBox.getStyleProperty (styleProperty);
KCheckBox.getValue ();
KCheckBox.onChange = OnChangeFunction;
KCheckBox.setEnabled (enabled);
KCheckBox.setLabel (label);
KCheckBox.setLabelPlacement (placement); // "right" or "left"
KCheckBox.setSize (size);
KCheckBox.setStyle (KStyleFormat);
KCheckBox.setStyleProperty (styleProperty, value);
KCheckBox.setValue (value);
  
```

For more details on dynamically setting Flash 6 component styles, please see `KStyleFormat` under Non-Interface Components.

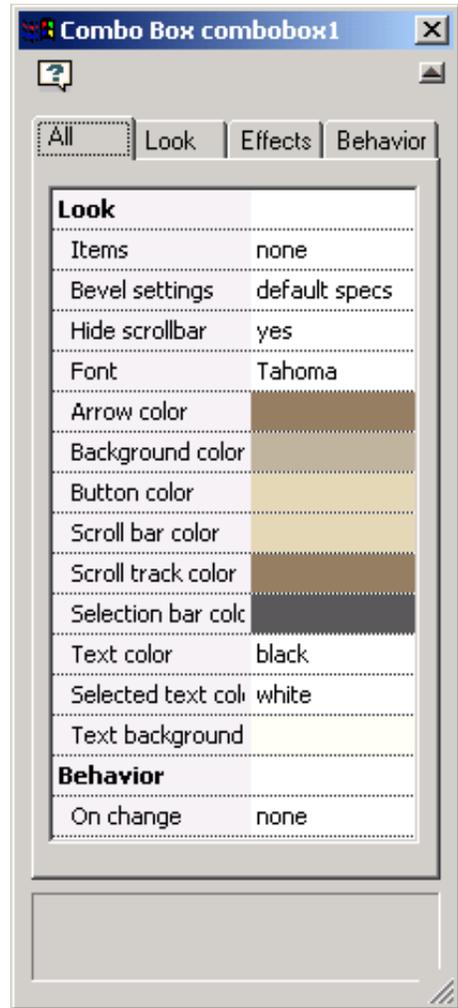
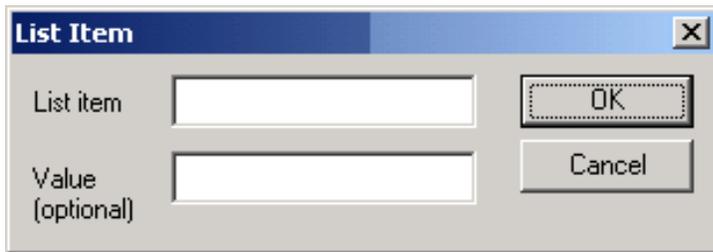


## Combo Box

 Flash 8. The Combo box allows you to display items in a drop down list. Click on Items and then the ellipses to open the List Box Items dialog window.

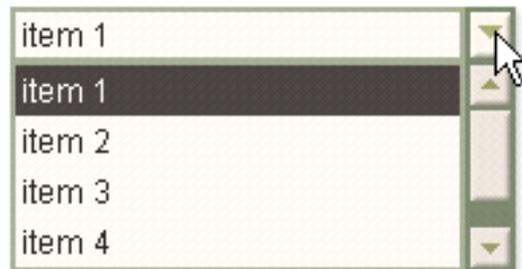


Click on the plus button to add items and their respective values to the list.



Just click on the down arrow to toggle the display.

Once the list is displayed, you can select from the list by clicking on an entry, or by using the arrow keys. Combo boxes can be added dynamically, through the toolbox, and through the Symbol library. When used dynamically, comboBoxes can be changed into listBoxes that allow multiple selections. For further details, see the comboBox implementation notes at [KoolMoves\Examples\Components\Fash 8\ComboBox\](#).



comboBox API (Flash 8):

### Functions

setItems(items,multiple,textFormat)

items - Array of items. Each item has to be an object containing label, value and selected (optional).

multiple - Boolean value indicating if multiple selections are allowed.

textFormat - textFormat object reference. Use this to set the font and font size. Font size freedom is limited because the height of a collapsed combo is always the same. If you specify a large value, you will see only part of the text. Example:

```
myTF = new TextFormat();
myTF.font = 'Arial';
myTF.size = 12;

items = new Array();
items.push({label:'line 1',value:1});
items.push({label:'line 2',value:2});
items.push({label:'line 3',value:3,selected:true});
items.push({label:'line 4',value:4});
items.push({label:'line 5',value:5});

myCombo.setItems(items,false,myTF);
```

setSelected(index / [indices]) - Set the selected item to the specified index / [indices]. If multiple selection is allowed, all other selected values are cleared.

getSelectedItems() - Returns an array of selected items.

getValues() - Function to be used when multiple selection is allowed. Returns an array of values of the selected items.

getValue() - Function to be used when multiple selection isn't allowed. Returns the value of the selected item.

### Properties

autoHide – Boolean value. When set to true, a scrollbar is only used if necessary. This property can only be set when setting up the component or just before setting new items

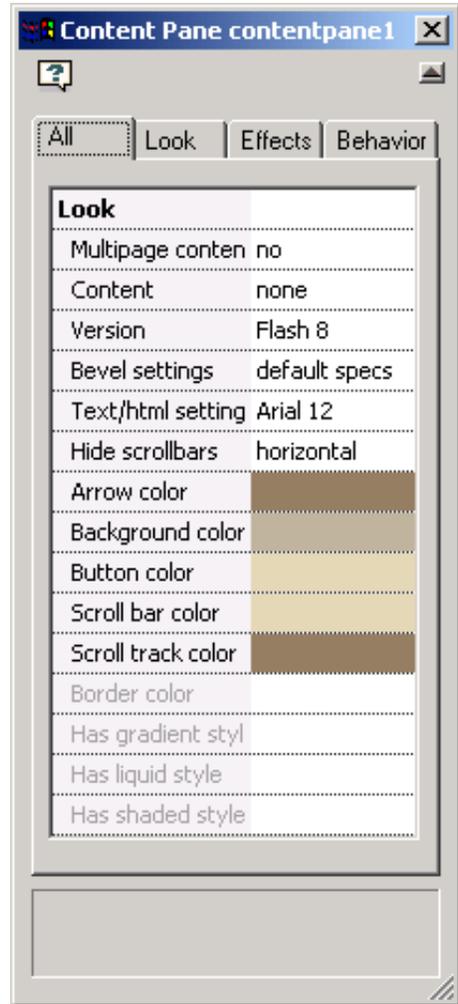
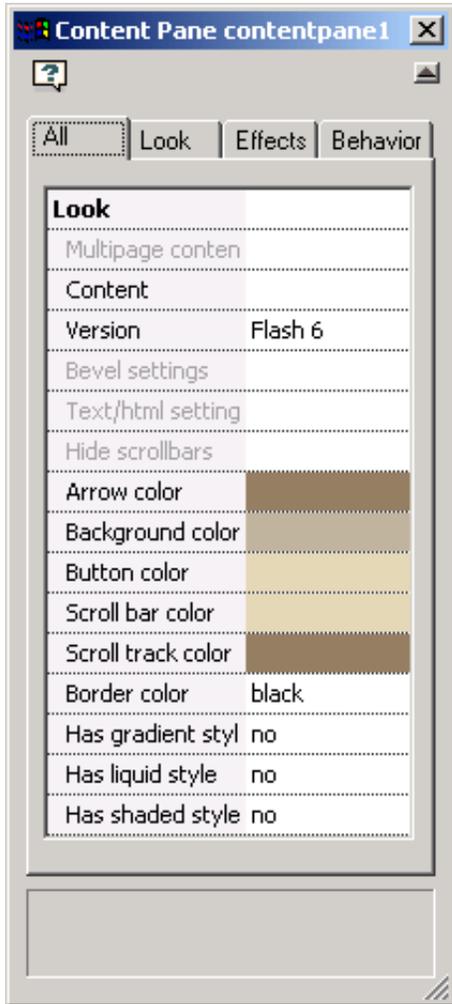
embedFonts - Analogous to textfield.embedFonts.

### Events

onChange(value) - Event occurring when another item is selected. If multiple selection is allowed, nothing is passed, otherwise the value of the selected item is passed, making it unnecessary to use the getValue function.

## Content Pane

 Flash 6 & 8. The Flash 6 version is also referred to as the scrollPane. ContentPanes are the most versatile and possibly the most useful Interface Component. The minimum size for a contentPane is 75 x 75 pixels. Scrollbars on the x and y axis are automatically available.

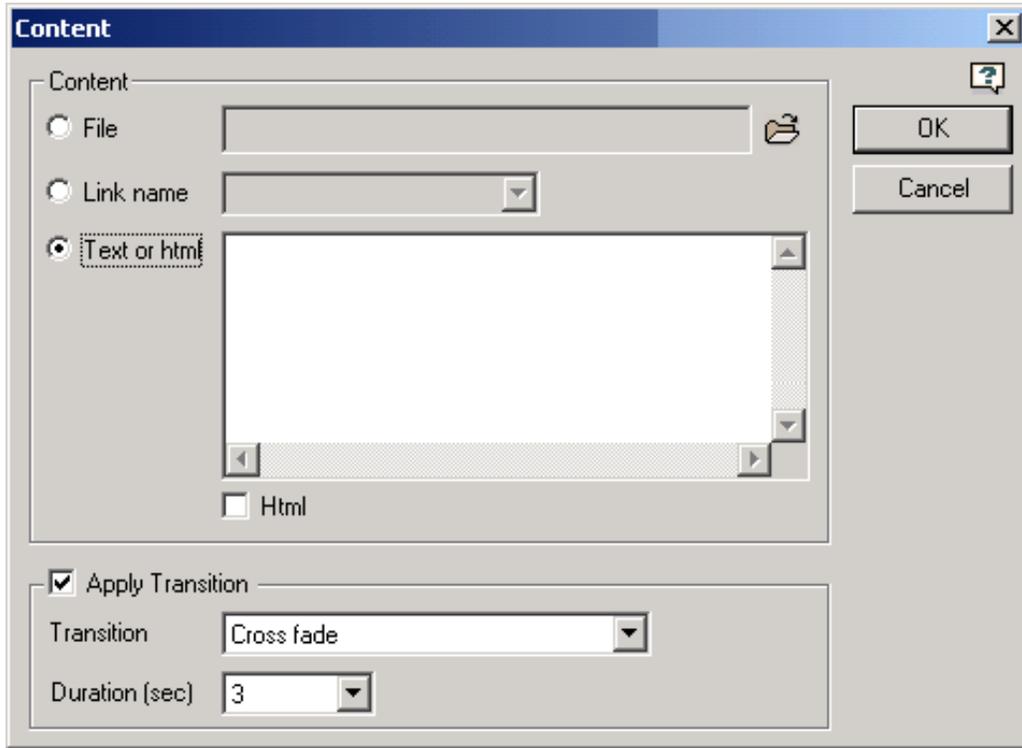


Unlike the comboBox and listBox, which display lists, contentPanes display information in other formats. A Flash 6 scrollPane can display a jpeg or flash (swf) file. A Flash 8 contentPane can display multiple items, treating each item as a separate page (Multipage content field equals yes). ContentPanes can display text strings, text files, graphic (gif, jpeg, png) files, flash files, bitmaps, and objects stored in the symbol library. Right clicking the component gives you the ability to zoom in on the content.



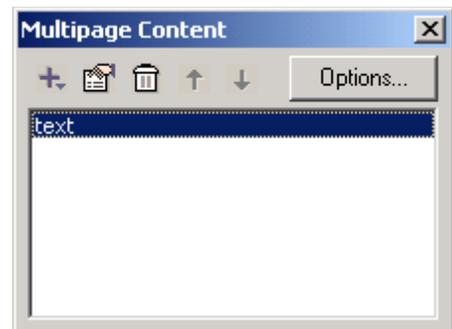
Clicking on the scrollPane Content field lets you identify the file to load.

Clicking on the contentPane Content field, before any content has been specified or when Multipage content field is no, opens the Content window. Use this dialog screen to specify a single page of content and its transition. Available transition styles are Cross Fade, Dissolve, Left to Right, and Right to Left.



- The combination of very long texts and font embedding may result in rendering problems. This can be resolved by breaking up the text or by turning off font embedding.
- Setting a movie(clip) with dynamic textfields as content may result in problems rendering the dynamic text.
- When using html text with images inside it, it is important to specify the width and height of the images, otherwise the width and the height of the text can not be detected right.

Clicking on the contentPane Content field, before any content has been specified or when Multipage content field is no, opens the Multipage Content window. Click on the plus to add additional content, click on the sheet-hand to edit the highlighted content (opens the Content window shown above), click on the trashcan to delete and the arrows to change the content's relative position in the stack.



Due to Flash Player security restrictions, a content transition will not work when content is an external graphic stored on another domain.

Two excellent examples of dynamically using scroll/contentPanels are at  
 KoolMoves\Examples\Components\Flash 6\Other examples\kcdemo.fun and  
 KoolMoves\Examples\Components\Flash 8\Examples\KC8 Demo\kc8.fun.

**KScrollPane API (Flash 6):**

```
MovieClip.CreateKScrollPane (instanceName, depth, initObject);
KScrollPane.destroy ();
KScrollPane.setEnabled ();
KScrollPane.getHeight ();
KScrollPane.getStyleProperty (styleProperty);
KScrollPane.getWidth ();
KScrollPane.loadContent (fileName, hscroll initial position,
    vscroll initial position);
KScrollPane.setEnabled (enabled);
KScrollPane.setSize (width, height);
KScrollPane.setStyle (KStyleFormat);
KScrollPane.setStyleProperty (styleProperty, value);
KScrollPane.setHeight (height);
KScrollPane.setWidth (width);
```

**contentPane API (Flash 8):**

**Functions**

getPage() - Returns the current page of multi page content.

setContent (type, source [, transition, transitTime [, clipRect, initObject ]]) – use to load content.

<u>Type</u>	<u>Source</u>
0 text	string containing text
1 html text	string containing html text
2 external text	filename of text file
3 external html text	file name of html text file
4 external graphic	file name of swf, gif, png or jpg
5 symbol library mc	linkage ID of movieclip
6 symbol library bitmap	linkage ID of bitmap
7 bitmap	bitmap

Transition – defines transition appearance.

- 0 Crossfade
- 1 Dissolve
- 2 Left to right
- 3 Top to bottom

transitTime - Value between 0 and 5, sets the number of seconds for the transition duration.

**ClipRect** - Rectangle of the `flash.geom.Rectangle` type representing the area to clip. When the content is a movie(clip), objects sometimes start outside the stage area resulting in a problem when trying to recognize the width and height of the movie(clip). This problem is solved by the ability to manually set the region that should be displayed.

**initObject** - Specified after `clipRect`; when content type is 'symbol library mc', you can pass an `initObject` to the specified symbol library mc.

Alternative syntax for multipage documents:

```
setContent (pageArray, startPage [,transition,transitTime])
```

`pageArray` is an array of pages that make up the content. Each array item is a sub array containing the parameters for a normal `setContent`.

`startPage` is the number of the page to display in the beginning.

The transition parameters can be used to set an initial transition that is used instead of the normal one specified for the first page to display. Example:

```
pages = new Array();
pages.push([3, 'firstPage.html']);
pages.push([2, 'secondPage.txt', 0, 3]);
pages.push([4, 'thirdPage.jpg', 1, 2]);
myContentPane.setContent(pages, 1, 0, 2.5);
```

**setContentFilters (filterArray)** - Sets the filters of the content. When using content filters, text background will be ignored and set to transparent. Example:

```
myContentPane.setContentFilters([new
flash.filters.DropShadowFilter(2)]);
```

**setNewTextFormat ( textFormat )** - Analogous to `textfield.setNewTextFormat`.

**setPage ( pageNumber )** - Sets the current page when multi page content is displayed.

**setScrollPos ( hsPos , vsPos )** - Sets the position of the scroll bars. Both can have a value between 0 and 1.

`setScrollPos(0,0)` sets the scrollers to the top left.

`setScrollPos(1,1)` sets the scrollers to the bottom right.

`SetScrollPos(0.5,0.5)` sets the scrollers to the middle.

**setTextFormat ( [beginIndex], [endIndex], textFormat)** - Analogous to `textfield.setTextFormat`.

**Properties:**

autoHide - Possible values are 0 through 3.

0 means hide no scrollbars

1 means hide the horizontal scrollbar if possible

2 means hide the vertical scrollbar if possible.

3 means hide both the horizontal scrollbar and vertical scrollbar if possible.

Notes:

- The vertical scrollbar is never hidden when content is word wrapped text.

- This can only be set when setting up the component or just before setting new content

backgroundColor - Analogous to textfield.backgroundColor; gets / sets the background color of the text when content is text.

barTrans - When set to true, only scrollbars and buttons are visible during a transition. When set to false the tracks are also visible.

embedFonts - Analogous to textfield.embedFonts

htmlText - Analogous to textfield.htmlText. Content has to be text.

ignoreCR - Boolean value, true by default. Windows textfiles often use both cr and lf resulting in blank lines between each line of text. When set to true, cr is ignored when using external text content.

ignoreLF - Boolean value, false by default. When set to true, lfs are ignored when using external text content.

length - Analogous to textfield.length

styleSheet - Analogous to textfield.styleSheet

text - Analogous to textfield.text. Content has to be text.

wordWrap - Analogous to textfield.wordWrap

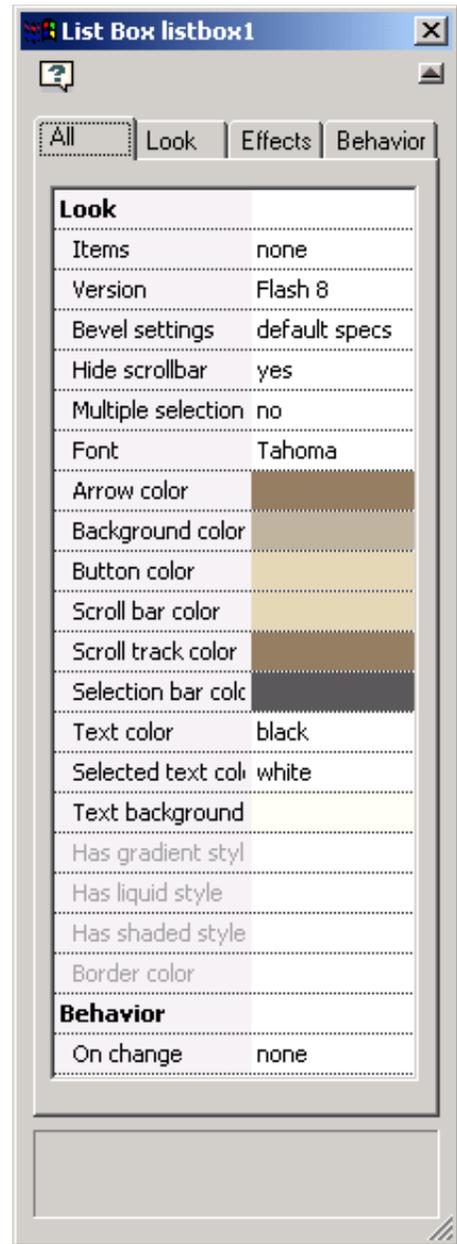
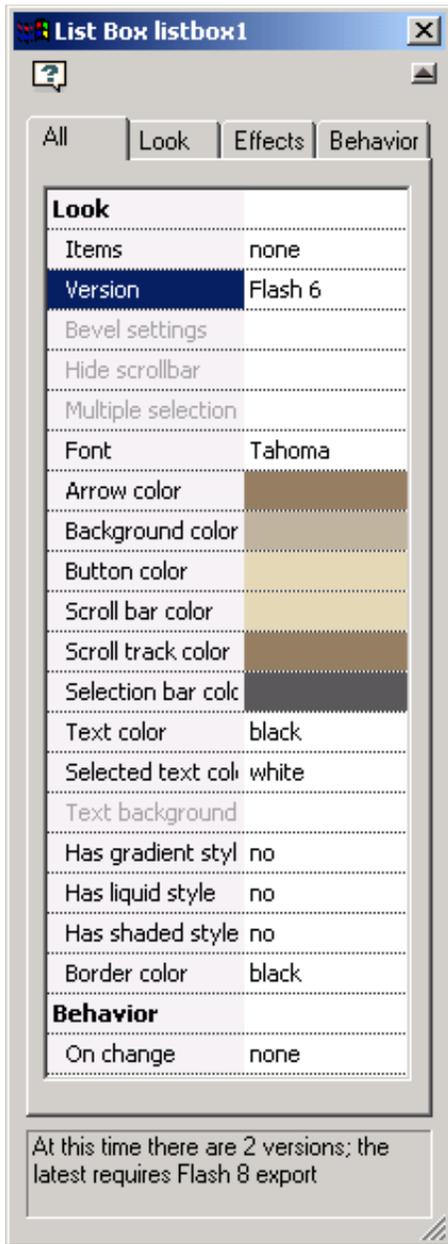
**Events**

onScroll ( hsPos , vsPos ) - Event occurring when content is scrolled.

onTransit - Event occurring just before the content transition takes place. This gives you, for example, the ability to see the position of the scrollbars before the content is first displayed.

## List Box

 Flash 6 & 8. Works pretty much the same as the comboBox, except that there is a Flash 6 version, it is a list instead of a drop down box, and there is an option to allow the component to have more than one item selected at a time. A Flash 8 dynamic example is located at KoolMoves\Examples\Components\FIash 8\Examples\ListBox\KC8\_Example\_ListBox.fun.



### KListBox API (Flash 6):

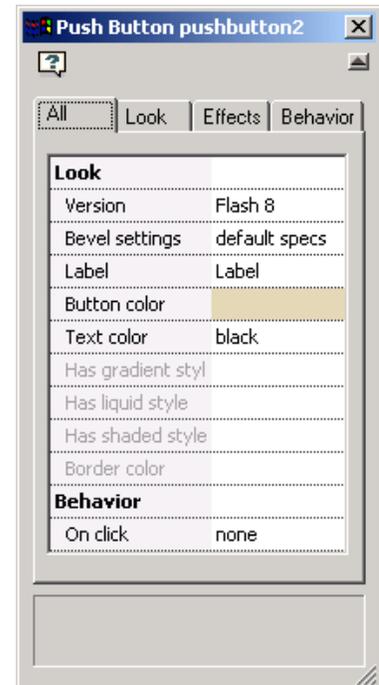
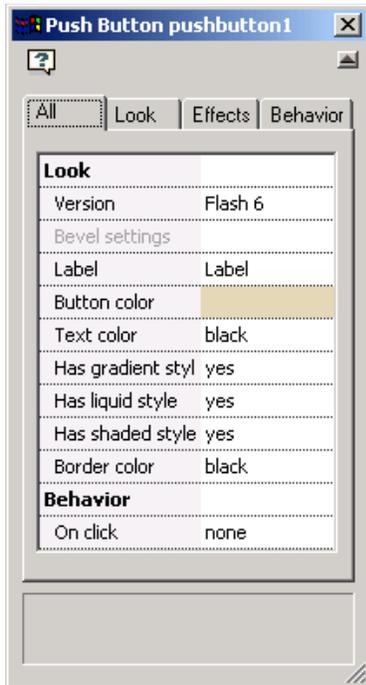
```
MovieClip.CreateKListBox (instanceName, depth, initObject);
KListBox.addItem (label, value);
KListBox.addItemAt (index, label, value);
KListBox.destroy ();
KListBox.setEnabled ();
KListBox.getItemAt (index);
KListBox.getItemLabel (index);
KListBox.getItemValue (index);
KListBox.getItemID (index);
KListBox.getLength ();
KListBox.getRowCount ();
KListBox.getScrollPosition ();
KListBox.getSelectedIndex ();
KListBox.getSelectedIndices ();
KListBox.getSelectedItem ();
KListBox.getSelectedItems ();
KListBox.getSelectMultiple ();
KListBox.getStyleProperty (styleProperty);
KListBox.getValue ();
KListBox.getWidth ();
KListBox.onChange = OnChangeFunction;
KListBox.removeAll ();
KListBox.removeItemAt (index);
KListBox.removeItemByLabel (label, isSubstring);
KListBox.removeItemByValue (value, isSubstring);
KListBox.replaceItemAt (index, label, value);
KListBox.setAutoHideScrollBar (autohide);
KListBox.setEnabled (enabled);
KListBox.setRowCount (rowcount);
KListBox.setScrollPosition (scrollPosition);
KListBox.setSelectedIndex (index);
KListBox.setSelectedIndices (selectedIndicesArray);
KListBox.setSelectMultiple (multiple);
KListBox.setStyle (KStyleFormat);
KListBox.setStyleProperty (styleProperty, value);
KListBox.setWidth (width);
KListBox.sortItemsByLabel (ascending);
KListBox.sortItemsByValue (ascending);
```

ListBox API (Flash 8): see the Flash 8 ComboBox API under ComboBox.

## Push Button



Flash 6 & 8. Click on the Label field to add a label as covered in [Check Box](#). Click on the On click field to add behavior as covered in [Check Box](#).



### KPushButton API (Flash 6):

```

MovieClip.CreateKPushButton (instanceName, depth, initObject);
KPushButton.destroy ();
KPushButton.setEnabled ();
KPushButton.getHeight ();
KPushButton.getLabel ();
KPushButton.getStyleProperty (styleProperty);
KPushButton.getWidth ();
KPushButton.onClick = OnClickFunction;
KPushButton.setAutoResize (autoResizeButton);
KPushButton.setEnabled (enabled);
KPushButton.setLabel (label);
KPushButton.setSize (width, height);
KPushButton.setStyle (KStyleFormat);
KPushButton.setStyleProperty (styleProperty, value);
KPushButton.setHeight (height);
KPushButton.setWidth (width);

```

Button API (Flash 8):

**Functions:**

setLabel(text, textFormat)

text - Label text in quotes.

textFormat - reference to textFormat object specifying font properties and alignment.

example:

```
myTF = new TextFormat();  
myTF.font = 'Arial';  
myTF.size = 12;  
myButton.setLabel('Caption', myTF);
```

**Properties:**

embedFonts - Analogous to textfield.embedFonts

**Events:**

onClick() - Event called when button is clicked.

## Slider

 Flash 8. Use the Slider API to control a parameter or change a value interactively. A straightforward example is available at `KoolMoves\Examples\Components\F8\Slider\KC8_Example_Slider.fun`.

Slider API (Flash 8):

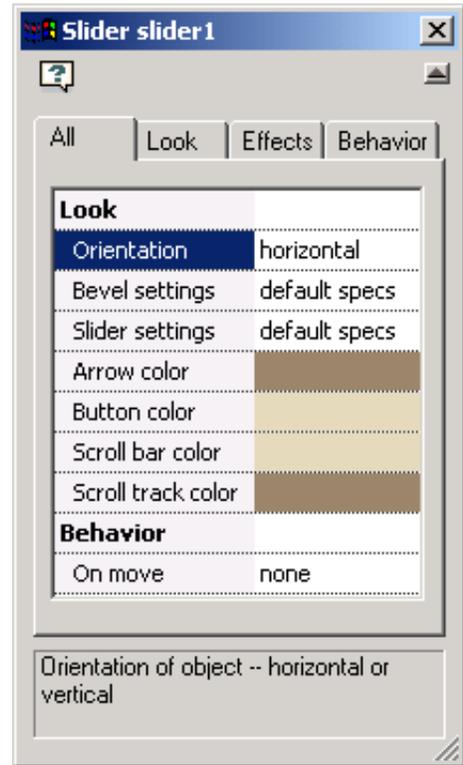
### Functions

`setSlider (min, max, current )` - Set the min, max and current value of the slider. All values have to be of the integer type. If you wish to use decimal values, you have to divide the result.

`getValue ()` - Returns the current value of the slider.

### Events

`onMove ()` - Event called when slider is moved.

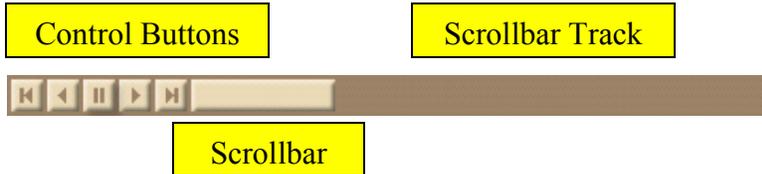


## SlideShow



Flash 8. The Slide Show component allows you to create a professional looking slideshow within your movie. Use the properties interface to organize the desired image files into a list, apply transition effects to the individual slides and set the appearance parameters for the slideshow interface.

The minimum size of the component is 125 x 75 (width by height) pixels with the navigation bar, 75 x 75 pixels without it. When turned on, the navigation bar will consume 19 pixels of the component's vertical display area, and the borders will consume an additional two pixels on all four sides of the display area.



The size of the scrollbar is 1/n of the scrollbar track, where n is the number of slides. The control buttons are, from left to right, Move to First Slide, Move to Previous Slide, Pause, Move to Next Slide, and Move to Last Slide. Clicking on any of the Move buttons will cause the slideshow to automatically pause at that slide – click on the Pause button to resume autoplay mode.

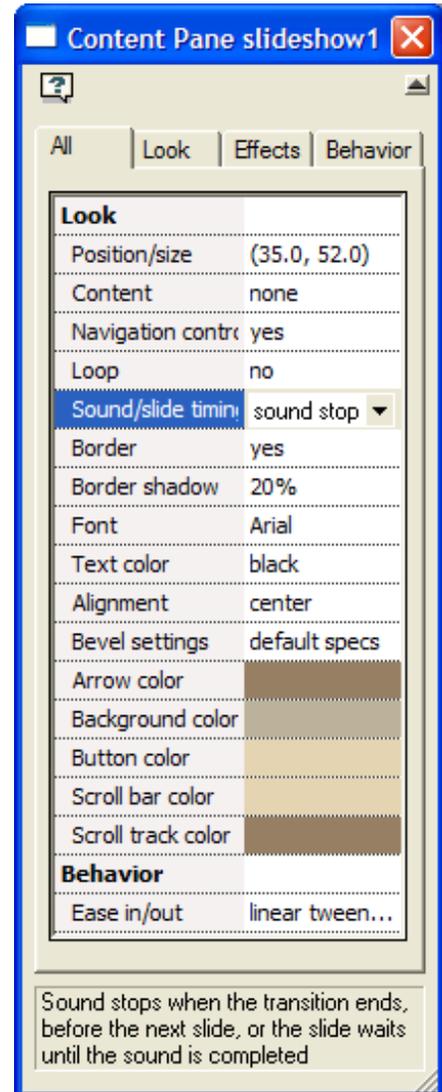
For a list of available transitions and export setting requirements, see [Slideshow Transitions](#) in Appendix B. For a quick demo of the available transitions, please see `\KoolMoves\Bin\transition_explorer.swf`. For slideshow component project examples, please see the .fun files in `\KoolMoves\Examples\Components\Flash 8\Examples\SlideShow\`.

### Slide Show API (Flash 8)

#### Functions:

`setSlides ( slides [, loopSlide [, callouts & clicks [, textformat ]])` – defines a slideshow.

`slides` is an array, each item of which is another array containing information for a slide in the slideshow (so `slides` is in effect a two dimensional array – actually three dimensional as there is a sub-sub array, described below). An item's position in the `slides` array determines that item's respective image's position in the slideshow. Each `slides` array item contains the following:



[ transition id, transition time, image / symbol library link name, image time, scale mode, slide background color, effects, sound link name, callouts & clicks indices, callouts transition ]

transition id – 32 bit hex integer (0x00000000) value that defines the transition to be used for a slide. For a list of available transitions and their respective id number, see [Slideshow Transitions](#) in Appendix B.

transition time – transition duration in floating point seconds.

image / symbol library link name – string value, either the filename of the slide image or the link name if the slide images has been placed in the symbol library. Slide link names should not contain a dot (.) because the dot is used to determine if the image source is a file or symbol link.

image time - slide duration in floating point seconds, after the transition effect has finished.

scale mode – integer value setting how the image is sized to the slideshow box:

- 0 = scale to fit
- 1 = scale and crop
- 2 = crop

slide bg color – 24 bit hex integer (0xRRGGBB) background color of a slide when the image doesn't fill the whole slide.

effects – integer value defining pixel based effects applied to an image. Except for sepia and b/w, effects can be combined by adding their values. The lined and channel build effects affect the transition and are more cpu intensive than b/w and sepia, which only affect the slide after the transition is over.

- 1 = lined (cpu intensive)
- 2 = sepia (not cpu intensive)
- 4 = b/w (not cpu intensive)
- 8 = channel build (mildly cpu intensive)

sound link name - the name in the symbol library used for the transition sound. External sound files are not supported. Link names are not allowed to contain a dot (.) because the dot is used to specify the sound loop value. For example 'mySound.3' will play the sound with link name 'mySound' three times. A sound automatically stops playing when a transition ends.

callouts & clicks indices –an array containing the array index numbers of the callouts and clickable area array to be used. [0,2] will use callout / clicks number 0 and 2 of the callouts & clicks array. Each callout or clickable area object containing the parameters for that particular callout or clickable area (see callouts and clicks below).

callouts transition – integer value used to select the enter/exit transition effect for that slide’s callouts and clickable areas. Slide callouts come in after the slide transition ends and go out just before the next slide transition starts.

- 0 = fade in/out
- 1 = from/to top
- 2 = from/to right
- 3 = from/to bottom
- 4 = from/to left

loopSlide is the zero based integer defining the target slide the slideshow will loop to when the end is reached. When set to -1, no looping will occur.

callout / click – an array of callout and clickable area definitions with the following parameters:

{ x : x-axis, y : y-axis, w : width, h : height, text : text string, link : URL hyperlink, cradius : corner radius, line : line color, fill : base fill color }

x - horizontal position of callout or clickable area, integer value.

y - vertical position of callout or clickable area, integer value.

w - width of the callout or clickable area, integer value.

h - height of the callout or clickable area, integer value.

text [optional] - text content of the callout box.

link [optional] - link to go to when callout or clickable area is clicked. This can be a slide, e-mail address or external website.

radius [optional] - radius of the corners of a callout, 0 based integer value. 0 means square corners, 6 (default) results in rounded corners.

line [optional] - line color of the callout box, 24 bit integer (0xRRGGBB)

fill [optional] - base fill color of the callout, 24 bit integer (0xRRGGBB)

textFormat – textFormat object reference. Use this to set the font and font size; see textFormat in the [Content Pane](#) component entry for an example of its use. .

gotoSlide( slide ) – string or integer value, this function moves the slideshow focus to the designated slide. slide value can be a slide number (starting at 0), 'first', 'last', 'previous' or 'next'.

setMode( mode ) – Sets slideshow to play automatically or pause for manual direction. mode can be:

- 0 = auto play

1 = manual navigation

**Properties:**

`_currentslide` - [read-only] returns the zero based number of the slide currently displayed.

`_totalslides` - [read-only] returns the total number of slides set for the slideshow.

**Events:**

`onEnd()` - Event occurs when the end of the slideshow is reached and `loopSlide` is set to -1.

`onLoop()` - Event occurs when the end of the slideshow is reached and `loopSlide` is set to a specific slide.

`onTransitStart()` - Event occurs just before a slide transition takes place.

`onTransitComplete()` - Event occurs just after a slide transition takes place.

## ***Interface Components not in the Toolbox***

There are a couple interface components that can not be created with through the Toolbox.

### **KScrollBar**

Flash 6. KScrollBar is a flash 6 component that attaches a scrollbar to a text field. It is created dynamically with the mc.CreateKScrollBar new object command. An example of scrollBar in use is located at KoolMoves\Examples\Components\Other examples\KcDemo.fun.

KScrollBar AP (Flash 6)I:

```
MovieClip.CreateKScrollBar (instanceName, depth, textfield,
horizontal);
KScrollBar.destroy ();
KScrollBar.setEnabled ();
KScrollBar.getStyleProperty (styleProperty);
KScrollBar.reAssign (object, controller, minvalue,
maxvalue, visibleSize, totalSize, horizontal);
KScrollBar.reAssignTextField (textfield, horizontal);
KScrollBar.setEnabled (enabled);
KScrollBar.setStyle (KStyleFormat);
KScrollBar.setStyleProperty (styleProperty, value);
```

### **Switch API:**

Flash 8. The Switch component is basically a toggle control; it is added at authoring time through the symbol library. The button and switch components are simple and have very little functionality: text color, alignment, bold, font name etc can all be set by using the textFormat class. An example is located at \KoolMoves\Examples\Components\Flash 8\Switch\KC8\_Example\_Switch.fun.

Switch API (Flash 8):

#### **Functions**

getSwitch() - Returns the status of the switch (true or false)

setSwitch( status ) – status = true (switched on) or false (switched off)

setLabel( text, textFormat )

text - Label text

textFormat - textFormat object specifying font properties and alignment.

example:

```
myTF = new TextFormat();
myTF.font = 'Arial';
myTF.size = 12;
myTF.align = 'left';
mySwitch.setLabel('Caption',myTF);
```

**Properties**

embedFonts - Analogous to textfield.embedFonts

**Events**

onSwitch() - Event called when switch is switched.

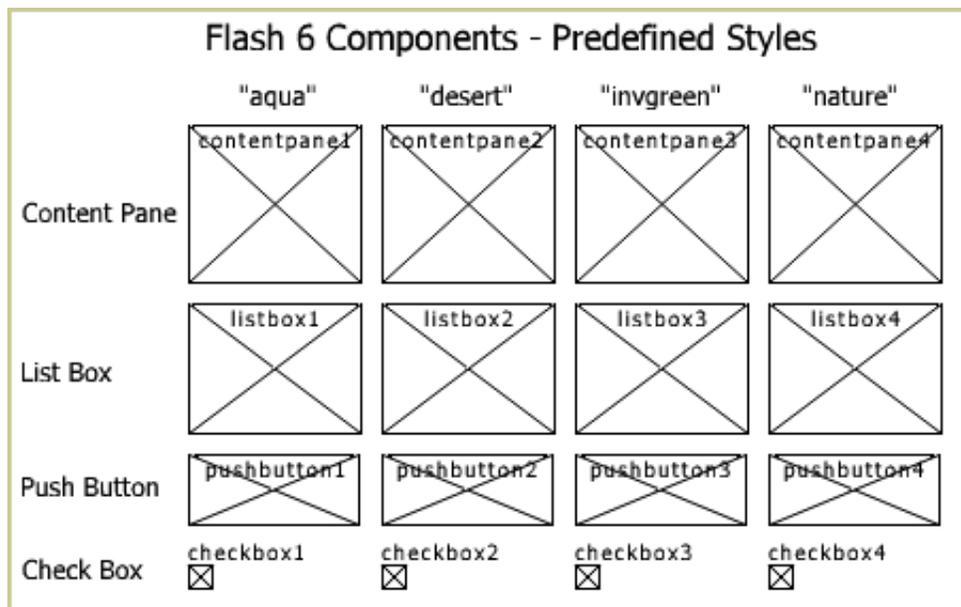
## Non-Interface Components

Not all of the components are interface components. The following classes add additional capabilities to KoolMoves without a visible presence in the movie.

### KStyleFormat

The KStyleFormat class is analogous to the TextFormat and Color classes, and is used to create object instances that can store a Flash 6 component's visual formatting (i.e., first you create a KStyleFormat object, then you set the style of the various properties, then use the object to apply your style to a Flash 6 interface component). KStyleFormat has four predefined formatting schemes ("aqua", "desert", "invgreen" and "nature").

Style objects allow you to quickly apply a consistent color scheme to related components. You can create the components in the authoring environment...



...and apply the style with ActionScript tied to an event or button action ...

```

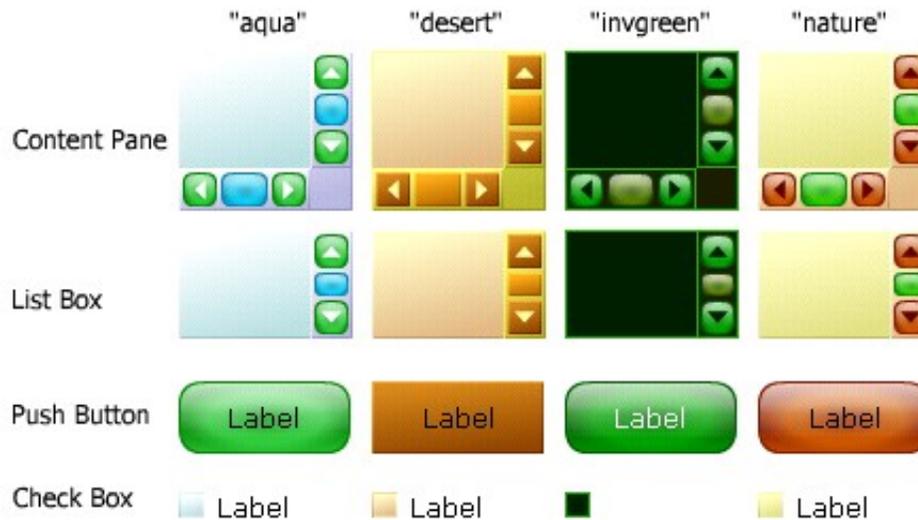
myStyle1 = new KStyleFormat("aqua");
myStyle2 = new KStyleFormat("desert");
myStyle3 = new KStyleFormat("invgreen");
myStyle4 = new KStyleFormat("nature");

contentpanel.setStyle ( myStyle1 );
listbox1.setStyle ( myStyle1 );
pushbutton1.setStyle ( myStyle1 );
checkbox1.setStyle ( myStyle1 );

```

...and see the results when you preview the animation....

## Flash 6 Components - Predefined Styles



A style scheme can be used as a starting point, with individual properties on a specific component modified as needed. For instance, if your movie is using the “invgreen” scheme, you can change the check box label textColor property to black (“checkbox3.textColor = 0x000000;”). And don’t forget, you can always create your own style scheme from scratch.

## KStyleFormat API (Flash 6):

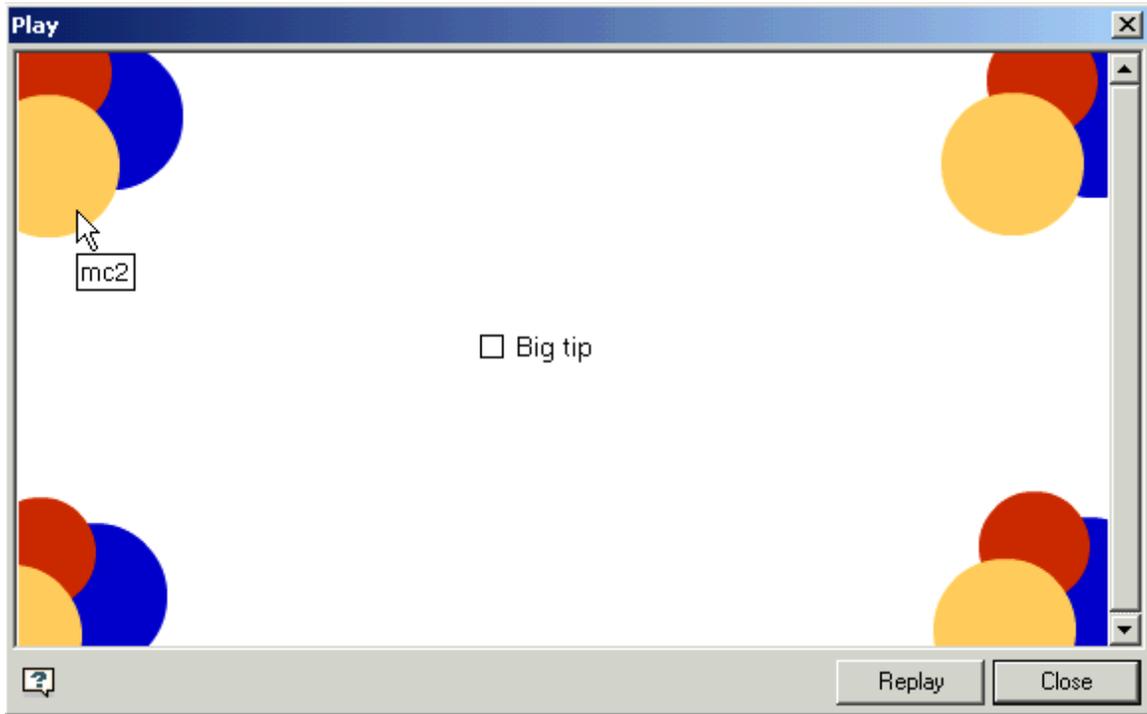
```

myStyle = new KStyleFormat();
myStyle.arrow = 0xRRGGBB;
myStyle.background = 0xRRGGBB;
myStyle.border = 0xRRGGBB;
myStyle.check = 0xRRGGBB;
myStyle.face = 0xRRGGBB;
myStyle.gradient = true/false;
myStyle.liquid = true/false;
myStyle.radioDot = 0xRRGGBB;
myStyle.scrollBar = 0xRRGGBB;
myStyle.scrollTrack = 0xRRGGBB;
myStyle.selection = 0xRRGGBB;
myStyle.shaded = true/false;
myStyle.textColor = 0xRRGGBB;
myStyle.textFont = FontFace;
myStyle.textSelected = 0xRRGGBB;
myStyle.textSize = FontSize;

```

## KToolTip

Instead of creating an object instance, KToolTip (Flash 6) adds a new capability to movie clips. After activating KToolTip, popup Screen Tips can be added to any movie clip. In the browser preview of \KoolMoves\Examples\Components\Tool Tip example\tiptest.fun below, pausing the mouse pointer over a movie clip causes a popup message to appear.



tiptest.fun is a single frame movie with a check box component and movie clips containing colored circles. After activating this feature, the code consists of assigning screen tip text to each movie clip and defining an event that changes the font size of the tips each time the checkbox is clicked on. The check box component itself is a movie clip, and has its own screen tip.

```
KToolTip.activate();

mc1._tooltip = "mc1";
mc1.mc1._tooltip = "mc1.1";
mc2._tooltip = "mc2";
mc3._tooltip = "mc3";
mc4._tooltip = "mc4";
mc4.mc1._tooltip = "mc4.1";
mc5._tooltip = "mc5";
mc6._tooltip = "mc6";
mc6.mc1._tooltip = "mc6.1";
mc7._tooltip = "mc7";
mc8._tooltip = "mc8";
mc8.mc1._tooltip = "mc8.1";
```

```
mc9._tooltip = "mc9";
checkbox1._tooltip = "Check to enlarge the tooltip";
checkbox1.onChange = function(){
    KToolTip.setStyleProperty("textSize", (this.getValue() ?
20 : 12));
}
```

### KToolTip API (Flash 6):

```
KToolTip.activate();
KToolTip.setStyle ( KStyleFormat );
KToolTip.setStyleProperty ( styleProperty, value );
```

### Usage:

```
KToolTip.activate();
myMC1._tooltip = "Tooltip for movieclip 1";
myMC2._tooltip = "Tooltip for movieclip 2";
```

## KRecordSet

This class is used to create an object to make a sql query and hold the results. After a query is passed to a 'bridge' file (covered below) the result is stored as an xml file. When the query successfully completes, the onQueried event function is executed and returns the information as an array of records. In this example it fills a listbox with information retrieved from the database.

```
myData = new KRecordSet();
myData.setDataBridge('xmlquery.php');
myData.query("select * from koolmoves order by name");
myData.onQueried = function() {
    for (var i = 0 ; i < this.records.length ; i++ ) {
        listbox1.addItem( this.records[i].name ,
            this.records[i] );
    }
}
```

By default, the bridge file is a php file named xmlquery.php, but theoretically asp files should also work. The bridge file contains the name of the database being accessed as well as all essential password and format information. An open connection to a mysql server where database queries can be directed is required.

By default, only select queries are allowed. Additional information about changing the query structure is included in the sample bridge file, i.e., it should be possible to use insert queries to add data. However, since there's no way to protect a swf file from being decompiled, anyone who studies the internet temp files could identify (and thus attack) the database. Because of this inability to protect the code, it is not wise to use Flash to access databases with sensitive information.

This is the table used in this example:

```
CREATE TABLE koolmoves (
    id int(10) unsigned NOT NULL auto_increment,
    name tinytext NOT NULL,
    kmversion tinytext NOT NULL,
    filesize tinytext NOT NULL,
    description mediumtext NOT NULL,
    urldemo tinytext NOT NULL,
    urldownload tinytext NOT NULL,
    PRIMARY KEY (id)
) TYPE=MyISAM;
```

The sample files (Koolmoves.fun, etc.) are located in \KoolMoves\Examples\Components\Flash 6\Database example\.

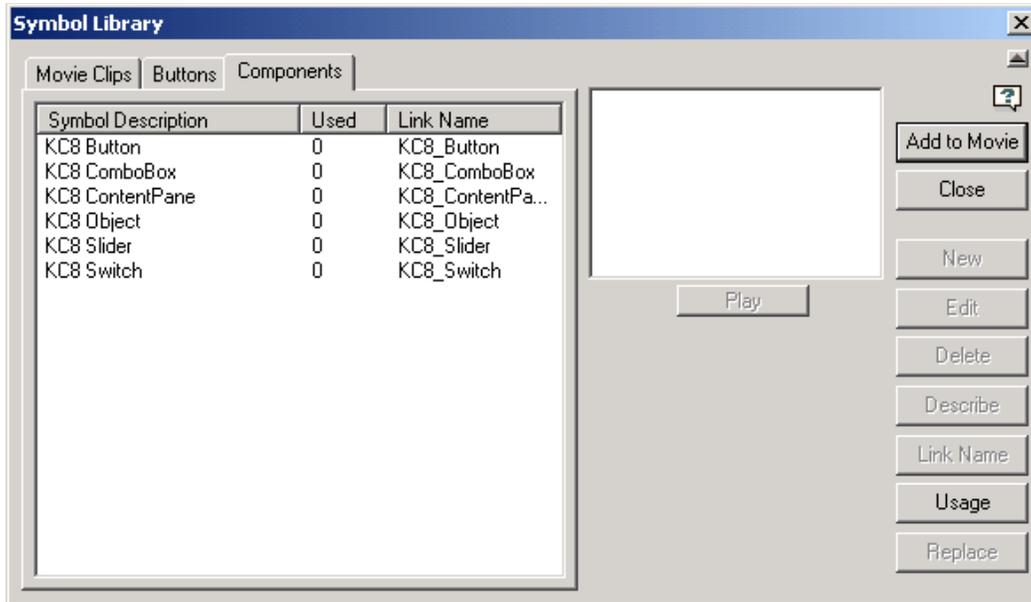
KRecordSet API (Flash 6):

```
myData = new KRecordSet();
```

```
myData.setDataBridge (filename);  
myData.query (selectquery);  
myData.onQueried = OnQueriedFunction;
```

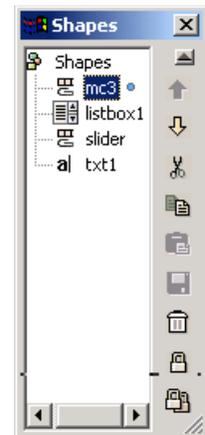
## Using the Symbol Library to add Components

All of the Flash 8 interface components can be added at authoring time through the Symbol Library. The Used column tracks how many times a particular Flash 8 component has been added to the movie at authoring time. The count includes components added through both the Symbol Library and the Toolbox. ListBoxes and ComboBoxes use the same template (they differ only in a parameter setting), so the number of both are combined in the KC8 ComboBox count.



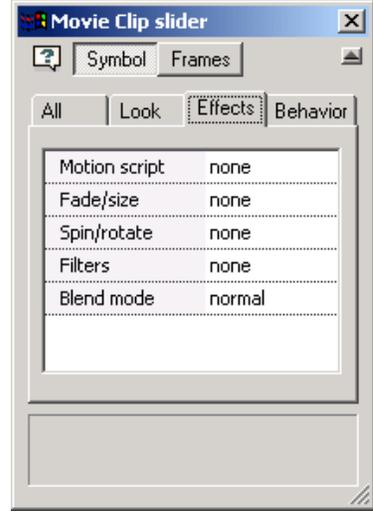
As mentioned earlier, component objects are members of a movieclip subclass. When a component is “attached” via the Symbol Library the base object containing the component is named in accordance with KoolMoves’ default movieclip nomenclature, ie, mcN, N being an integer reflecting the order of the movieclip’s creation. In the Shapes data view to the right, mc3 contains a comboBox, listBox1 was added through the Toolbox, and slider is another component object that was manually renamed to be more descriptive.

The main difference between components attached via the GUI and the Symbol Library is that the former are initialized before being placed in the movie, while the latter is initialized after with the `mc.attachMovie()` function.





It was mentioned earlier that effects could not be applied directly to components. However, effects *can* be applied to movieclips (and thus indirectly to their contents). Thus, movie clip slider has effects available while List Box listbox1 has no effects available.



## Chapter 15 – Symbol and Link Library

Before we start discussing the Symbol and Link Libraries, let us have a brief discussion about the ActionScript commands which are referred to in this chapter.

`mc.attachMovie()` is used to dynamically create an object instance based on a source object in the project's Symbol Library. The symbol library assigns a Link Name to each source object, and `mc.attachMovie()` uses the link to create a copy of the object on the “mc.” parent object (in most cases “\_root” which means straight on the Stage).

After defining Link Names in the Link Library to external sound files and bitmap images,

- `soundObject.attachSound()` can be used to load sound files, and
- `flash.display.BitmapData.loadBitmap()` can load bitmap images.

### Symbol Library

The symbol library provides an easy mechanism for reusing Movie Clips, Buttons and Components throughout the movie. Use the Symbol library to organize, track, edit and export library members. Edits to a symbol are reflected in all its instances. Additionally, movie clips can have a link applied, either through the library Link Name button or the ActionScript `attachMovie()` method.

The symbol library is limited to movie clips, buttons and components. Changing a shape to a Symbol will not add it to the symbol library; instead, convert it to a movie clip and proceed from there. If you want to work with external images, define the image's file name in the Link Library and add it to the movie with the `attachMovie()` method.

### Adding an element to the Symbol Library

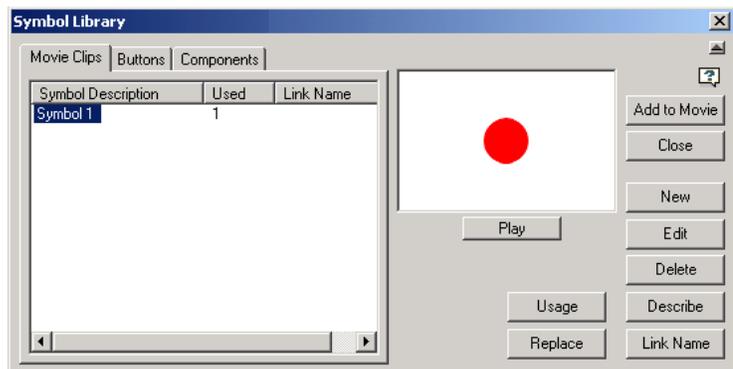


Select a movie clip or button and use `Shapes > Add to Symbol Library` or `Ctrl-F11` to add the element to the Symbol Library. Then change the description from the default value of Symbol N to something useful.

### The Symbol Library Interface

Use `View > Symbol Library` or press `F11` to open the Symbol Library.

Within the Symbol Library is a list of all symbols saved to the Symbol Library, showing their description, the number of times they have been manually placed in the movie, and the Link Name (optional) applied to the symbol.



Click on Add to Movie to place the selected symbol on the active frame. After closing the library you can manually position the symbol as desired.

Link Names allow the dynamic use of symbols and the application of behaviors to a symbol (look up attachMovie, duplicateMovie, Links, Object.registerClass, and flash.display.BitmapData.loadBitmap in an ActionScript manual). As mentioned in the Buttons chapter, buttons can not be accessed dynamically, so Link Names can not be assigned to them.

The New button allows you to create a movie clip or button from scratch and save it to the Symbol Library. These objects will only exist in the project's symbol library until they are added to the movie. While both button and movie clip symbols created from scratch can be added to a movie with the Add to Movie button, only movie clip symbols (or components) are added to a movie with the attachMovie function.

The Edit button opens the selected button or movie clip symbol for editing.

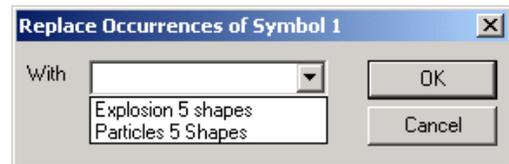
The Delete button removes the selected button or movie clip symbol from the Symbol Library.

The Describe and Link Name buttons open dialog boxes allowing you to change values in these fields for the selected button or movie clip symbol.



Click on the Usage button to display the names and locations of all instances generated (in authoring time) for a given symbol. Dynamically created copies of symbols are not tracked in the Symbol Library.

With the Replace function a movie clip or button symbol placed on the stage can be replaced globally with another movie clip or button, respectively. Select an alternative from the drop down list of available movie clips or buttons in the library. The replace function will not replace "complete" copies of the symbol that were created manually (i.e., using Ctrl-C or Edit > Copy Shapes and then pasted with Ctrl-V or Edit > Paste and the "complete copy" option selected). This means that button copies created with shared states will also be replaced when the symbol replace function is used.



## Importing Symbols

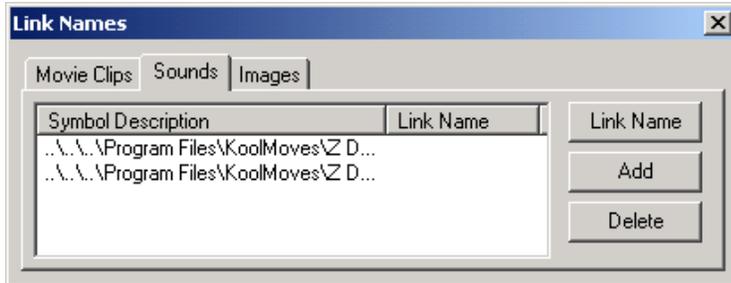
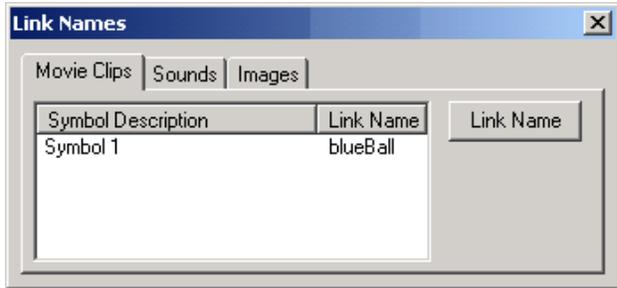
To load the symbol library from a fun file into the current project, preview the fun file with either the File > Insert KoolMoves Movie menu command or the Tools Toolbar Import Clip Art library function. The source file does not actually have to be inserted; the act of previewing it copies the symbol library into the current project's symbol library. Repeated clicking on the preview button will create additional copies of the symbols in the library.

At this time you can't pick and choose which symbols to load; the entire set in the source file comes over. Be sure to give your library symbols unique names.

## Link Library

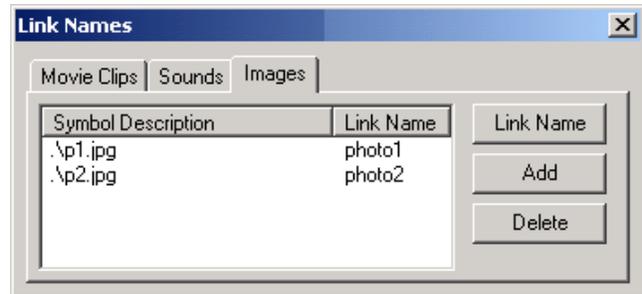
 In the Action Script Editor click on the Links button to open the Link Names window.

The Movie Clips tab displays available movie clip links created in the main gui with the Symbol Library interface. These links are used with the `mc.attachMovie()` action script command.



Under the Sounds tab, links to external sound files can be created. These links can be used with the `sound.attachSound()` action script command and the KoolMoves [Slideshow](#) component.

Under the Images tab, links to external image files can be created. These links can be used with the `flash.display.BitmapData.loadBitmap()` action script command and the KoolMoves [Slideshow](#) component. Supported formats are BMP, GIF, JPG and PNG.





## **Appendices**

**Appendix A – Known Issues and Error Messages**

**Appendix B – KoolMoves Reference Tables**

**Appendix C – Programming / Coding Advice et. al.**

**Appendix D – Useful 3rd Party Software**

**Appendix E – Toolbar Library Listings**

**Appendix F – Frequently Asked Questions (FAQ)**



## Appendix A – Known Issues and Error Messages

The performance of the Flash™ player is influenced by many factors -- the number of shapes in each frame, the number of points in each frame, the degree to which transparency is used, security settings, etc. The performance of the KoolMoves internal player is influenced by the same factors. In general, KoolMoves player is not as efficient as the Flash™ player, particularly with regard to rendering shapes with transparency and bitmap fills. The performance on a web site is a different issue and is influenced by typical internet limitations of modem speed, ISP bottlenecks, etc.

If other large applications are running while KoolMoves is running, KoolMoves may crash. This is due to a lack of system or graphic resources. Other causes of crashes are: bad video card drivers and free RAM memory managers.

Review the Movie Preferences and Export Settings sections for more information on parameters that affect movie performance.

### ***Exceeded SWF Buffer Size***

The most likely cause of this error message is that you are using a wav file that exceeds 1 MB or an image that compresses to a size greater than 1 MB.

Your best options are to either convert the wav file to MP3 which does not have a limit or to adjust the compression setting for the wav file in File > Export Settings. Using 100 JPEG quality in File > Export Settings is generally a very bad idea. If you need a 100 quality setting, you should be using gifs rather than jpegs.

### ***SWF Too Big***

The Flash player is not built for playing large files. Playing large swf movies can easily freeze a computer. So many different parameters can affect performance that Adobe (and earlier Macromedia) does not have a recommended swf size for Windows systems, though they posit that there is an optimal value for each system; for the Flash 4 stand alone player in Mac systems, Macromedia recommended that swfs not exceed 4 MB. This and other factors affecting performance are explained in

[http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_14437](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_14437).

The Flash player does have some concrete limitations that can not be exceeded: 16,000 frames, 16,000 loaded movies, 16,000 layers, and 16,000 symbol instances. “Combining multiple limits in a single Flash file would create greater risk...”, i.e., the Flash Player is optimized for normal circumstances, and testing the limits “...can cause memory and other operating system issues.”

Your best options are to either reduce the size of the movie (in particular, MP3 files using products such as GoldWave) or to daisy chain swf movies together using the Load Movie action.

With KoolMoves 5.0 and later, you will be warned when your swf file equals or exceeds 5MB. Earlier versions of KoolMoves have the warning set at 3MB.

## ***Animation Not Displaying or Warning Messages Display***

If the animation does not display in the preview window, there are several possible reasons:

- 1) You don't have the Flash plugin. Using Internet Explorer browser, go to [www.adobe.com](http://www.adobe.com) download page and download the latest Flash plugin.
- 2) You have javascript or activeX disabled. You can control this in Internet Explorer options settings.
- 3) There are security settings issues; see Internet Explorer Security Issues below.

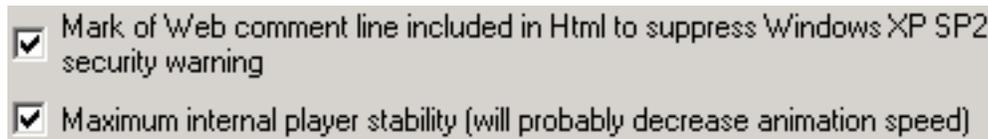
## ***Internet Explorer Security Issues***

With the introduction of Windows XP SP2, there have been changes to Windows which affect Flash animation. To avoid a security warning when a Flash animation plays from a local file off your hard disk, a special comment line can be added to the html file. By default, KoolMoves adds the following line (called "Mark of Web"):

```
<!-- saved from url=(0013)about:internet -->
```

This line causes the file to display using the Internet security level rather than the Local security level, which tends to be more restrictive for XP SP2, in particular in preventing active file content (in this case Flash movies) from displaying.

All animation preview windows in KoolMoves use local html files so this issue affects KoolMoves functionality. Other operating systems and service packs or security configurations may have a problem with this line. In any case, you can control whether the line is added or not added in the File > Preferences > Play tab.



You can also control the Internet Explorer security warning in Windows XP SP2, by opening Internet Explorer and clicking on Tools/Internet Options/Advanced and searching for the Security group, the last one. Enable "Allow active content to run in files on My Computer".

A reference to this subject is

[http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_19542](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_19542). A more general reference is [microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.msp](http://microsoft.com/technet/prodtechnol/winxppro/maintain/sp2brows.msp).

## Appendix B – KoolMoves Reference Tables

This appendix contains useful references for the KoolMoves interface and associated APIs.

### Menu Structure

Menu entries followed by ellipses ("...") open dialog windows. Shortcut keystrokes are shown where available.

<b>File</b>		<b>Copy Shapes</b>	<b>Ctrl+C</b>
New	Ctrl+N	<b>Delete Shapes</b>	<b>Del</b>
Open...	Ctrl+O	<b>Paste Shapes</b>	<b>Ctrl+V</b>
Save	Ctrl+S	<b>Paste Shapes to Beginning</b>	
Save As...		<b>Paste Shapes to End</b>	
Export Movie		<b>Paste to File...</b>	
Add Flash Movie to Web Page (Wizard)...			
Add Flash Movie to Web Page (Expert)...			
As Swf (Flash Movie)...			
As SWF (Flash Movie) and HTML Page...			
As SVG (Scalable Vector Graphics)...			
As Frame Captures...			
Diagnose Server Problems...			
Export Settings...			
Import...			
Insert from Animation Libraries			
Web Interfaces...			
Text Effects Templates...			
Clip Art...			
Animated Clip Art...			
Buttons...			
Insert KoolMoves Movie...			
Trace Image			
Display Image			
Clear Image			
Configure...			
Preferences...			
1 Most Recent Files			
2 Most Recent Files			
3 Most Recent Files			
4 Most Recent Files			
Exit			
<b>Edit</b>			
Undo	Ctrl+Z		
Redo	Ctrl+Y		
Undo History...			
Redo History...			
Cut Shapes	Ctrl+X		
		<b>Views</b>	
		Shape Properties...	Ctrl+F3
		Score/Timeline...	Ctrl+Shift+F3
		Movie Overview...	Alt+F3
		Actions and Sounds Overview...	Ctrl+E
		Symbol Library...	F11
		List of Shapes...	F9
		List of Key Frames...	Ctrl+K
		List of Scenes...	Shift+F2
		Navigation Overview...	
		Tools	
		Status Bar	
		<b>Shapes</b>	
		Select/Move	
		Selection Scope	
		This Frame	
		All Frames	
		This Frame to First Frame	
		This Frame to Last Frame	
		Select Same Color	
		Invert Selection	Ctrl+I
		Hide Selection	Ctrl+I-I
		Last Selection	Ctrl+L
		Add to Symbol Library...	Ctrl+F11
		Draw	
		Rectangle	
		Square	
		Rounded Rectangle	
		Ellipse	
		Circle with Curves	
		Circle with Line Segments	

## KoolMoves User Manual

Star		All Frames	
Polygon		This Frame to First Frame	
Freehand Shape		This Frame to Last Frame	
Freehand Line		Hide Selection	
Point by Point		Add	
Point by Point, Open Border		Delete	Del
Line		Append to Line	
Add Text...		Get Position...	
Add Dynamic Text		Set Position...	
Add Interface Component		Selection Area	
Slide Show		Color Selected Points	
Content Pane		Color Unselected Points	
List Box			
Combo Box		Effects	
Push Button		Motion Script...	
Check Box		Fade/Size...	
Slider		Spin/Rotate...	
Add Button		Filters...	
Convert to Button		Blend Mode...	
Add Movie Clip		Drop Shadow...	
Convert to Movie Clip			
Align		Transforms	
Left	Ctrl+Shift+L	Break Apart Text	
Right	Ctrl+Shift+R	Break into Text Characters	
Top	Ctrl+Shift+T	Convert to a Shape with Points	
Bottom	Ctrl+Shift+B	Rotate/Shift Gradient Fill	
Horizontal Center	Ctrl+Shift+H	Scale Gradient Fill	
Vertical Center	Ctrl+Shift+V	Rotate Image Fill	
Change Depth		Shift Image Fill	
Bring Forward	Ctrl+Up	Scale Image Fill	
Send Backward	Ctrl+Down	Numeric...	Ctrl+Alt+N
Bring to Front	Ctrl+Shift+Up	Center	Ctrl+Alt+C
Send to Back	Ctrl+Shift+Down	Perspective.	Ctrl+Alt+P
Bring Forward Selection...	Alt+Up	Rotate	Ctrl+Alt+R
Send Backward Selection...	Alt+Down	Scale	Ctrl+Alt+S
Group	Ctrl+G	Slant	Ctrl+Alt+T
Add to Group	Ctrl+Alt+G	Squish	Ctrl+Alt+Q
Ungroup	Ctrl+Shift+G	Flip Horizontally	Ctrl+Alt+H
Select All	Ctrl+A	Flip Vertically	Ctrl+Alt+V
Unselect All	Ctrl+Shift+A	Rotate Left 90°	
Advanced		Rotate Right 90°	
Morph Source		Scale/Translate Movie...	
		Scale/Translate Selected Shapes...	
Points		Frames	
Select/Move		Increase Movie Length	
This Frame			

## KoolMoves User Manual

Add Blank Frames...		Zoom Out on Click	F3
Duplicate Last Frame...		No Zoom	F4
Add Copy of Frame		Pan	
Add Copy of Fame to End	Ctrl+End	No Pan	
Add Blank Frame	Ctrl+B	Grid	
Add Blank Frame to End		No Grid	
Delete Frame		6 Pixel Grid	
Next Frame	PgDown	10 Pixel Grid	
Previous Frame	PgUp	12 Pixel Grid	
Default Tweens...		18 Pixel Grid	
Edit Actions/Sounds...	F8	24 Pixel Grid	
Edit Movie Clip Frames	Ctrl+F	Snap to Grid	
		Show All Masking	Ctrl+M
		View Associated Joints	
		Antialized Rendering	
		Wire Frame Rendering	
		Onion Skinning...	
		View Parent Movie	
<b>Movie</b>			
Movie Width/Height...			
Movie Speed...			
Background Color...			
Background Image...			
Preloader...			
Movie Dependencies...	Ctrl+D		
SWF File Statistics...			
		<b>Help</b>	
		Help Topics	F1
		Interactive Tutorial...	
		Simple Task Assistant...	Ctrl+Q
		FAQ...	
		Tip of the Day...	
		Tutorials...	
		Examples...	
		Purchase...	
		Register...	
		Update Software...	
		KoolMoves Website...	
		Support Forum...	
		Email Us...	
		About KoolMoves...	
<b>Play</b>			
Play in Web Browser	Ctrl+Space Bar		
Play Portion of Movie...			
Play in Stand-Alone Player	Ctrl+Alt+Enter		
Set Play Directory...			
Single Step			
Step Forward	>		
Step Backward	<		
<b>Options</b>			
Reposition Movie Frame...			
Zoom Dragged Rectangle			
Zoom In on Click	F2		

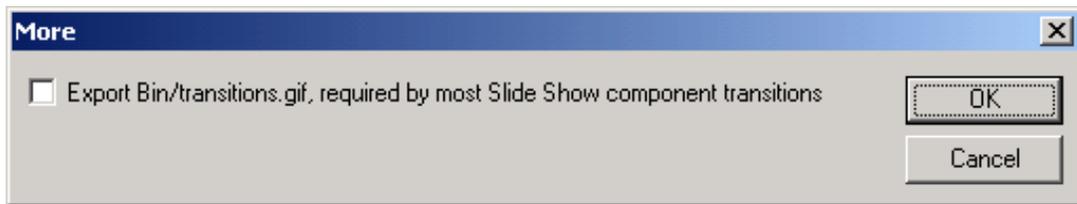
## Shortcut Keys

<	Single step backwards	Ctrl+Alt+S	Scale
>	Single step forwards	Ctrl+Alt+T	Slant
PgUp	Previous Frame	Ctrl+Shift+T	Align tops
PgDn	Next Frame	Ctrl+V	Paste shapes
Ctrl+A	Select all shapes	Ctrl+Alt+V	Flip vertically
Ctrl+Shift+A	Unselect all shapes	Ctrl+Shift+V	Align on vertical
Ctrl+B	Add blank frame	centers	
Ctrl+Shift+B	Align bottoms	Ctrl+Down	Send backward
Ctrl+C	Copy shapes	Alt+Down	Send backward popup
Ctrl+Alt+C	Center	Ctrl+Shift+Down	Send to back
Ctrl+D	Dependencies	Ctrl+End	Add frame to end
Ctrl+E	Sounds and actions	Esc	Stop internal player
Ctrl+G	Group shapes	F1	Help
Ctrl+Alt+G	Add to group	F10	List key frames
Ctrl+Shift+G	Ungroup shapes	F2	Zoom in
Ctrl+H	Hide/unhide shapes	F3	Zoom out
Ctrl+Atl+H	Flip horizontally	F4	No zoom
Ctrl+Shift+H	Align on horizontal	F5	Add point
centers		F6	Delete point
Ctrl+I	Invert selection	F7	Move point
Ctrl+K	List key frames	F9	List shapes
Ctrl+L	Last selection	Ctrl+Enter	Play in internal player
Ctrl+Shift+L	Align left	Ctrl+Alt+Enter	Play in stand-alone
Ctrl+M	Show masking	player	
Ctrl+N	New file	Ctrl+Space	Play in browser
Ctrl+Alt+N	Numeric transform	Ctrl+Up	Bring forward
Ctrl+O	Open file	Alt+Up	Bring forward popup
Ctrl+Alt+P	Perspective	Ctrl+Shift+Up	Bring to front
Ctrl+Alt+Q	Squish	Del	Delete shapes or points
Ctrl+Alt+R	Rotate	Ctrl+X	Cut shapes
Ctrl+Shift+R	Align right	Ctrl+Y	Redo
Ctrl+S	Save file	Ctrl+Z	Undo

## Slide Show Transition ID Table

The table below contains the Transition ID Numbers and their respective Transition Names. Please see the [Slide Show](#) API entry in Chapter 14 to see how these transitions are dynamically applied to slides.

Slideshows using transition names tagged with a trailing asterisk (" \*") use bitmap images stored in a gif file (transitions.gif) and need to have this file embedded in the swf to apply their effect. With swf compression enabled, slide shows that only use untagged effects add about 12.3KB to the swf (not including slide images), while slideshows using tagged effects (and thus embedding transitions.gif) add about 25KB. Click on the Flash 8 Advanced button in the Export Settings dialog and then click on the "Export Bin/transitions.gif, required by most Slide Show component transitions" check box, then click on OK to have these patterns included in the swf when your project is exported.



The values referenced above reflect the slide show component as introduced with KoolMoves 5.3, and may change more patterns are added. For slideshows with a lot of embedded images the tagged effects have relatively minor impact, but for a simple rotating banner with external images that needs to load very fast, consider limiting your project to untagged transition effects.

There is a difference in cpu load between different effects, but more important than whether or not the transitions.gif file is used, is the complexity of the rendering by the flash player, i. e., the fancier the effect the greater the cpu load. Ultimately, the factors with the biggest impact on transition effect processing requirements are the image sizes and the movie frame rate.

0x00000000	no effect	0x0113fcc0	checkerboard, out *
0x2b805cc0	bars, horizontal *	0x0013fc20	circle in [block style] *
0x2b805ec0	bars, vertical *		
0x4301dec0	blinds, large, diagonal, left-down *	0x0013fd20	circle out [block style] *
0x4301ddc0	blinds, large, diagonal, left-up *	0x0013fcc0	circles, in *
0x4301dcc0	blinds, large, diagonal, right-down *	0x0013fdc0	circles, out *
0x4301dfc0	blinds, large, diagonal, right-up *	0x0013fdc4	circles, rotating in *
0x23801cc0	blinds, large, horizontal *	0x4113fd40	claws, closing *
0x23801fc0	blinds, large, vertical *	0x4113fc40	claws, opening *
0x2b804cc0	blinds, small, horizontal *	0x00000009	cover down
0x2b804fc0	blinds, small, vertical *	0x00000006	cover left
0x0113fdc0	checkerboard, in *	0x0000000b	cover left-down
		0x0000000c	cover left-up
		0x00000007	cover right
		0x0000000a	cover right-down
		0x0000000d	cover right-up
		0x00000008	cover up
		0x8113fc40	curves down *
		0x8113fe80	curves left *

## KoolMoves User Manual

0x8113ff80	curves right *	0xfbffff97	random bars, vertical, cover right
0x8113fd40	curves up *	0xfbffff00	random bars, vertical, large
0x4113fd20	diamond in [block style] *	0x0153dfc0	rectangles, in *
0x4113fc20	diamond out [block style] *	0x0113fd00	rectangles, large, in *
0x4113fdc0	diamonds out *	0x1151df00	rectangles, large, in corners *
0x4113fd80	diamonds, narrow, in *	0x0153df00	rectangles, large, in sides *
0x4113fc80	diamonds, narrow, out *	0x0153dd00	rectangles, large, in top/bottom *
0xfbefcc00	dissolve [block style]	0x0113fe00	rectangles, large, out *
0xfbdfcc00	dissolve, little waves	0x1151de00	rectangles, large, out corners *
0xfbefcc00	dissolve, pixel	0x0153de00	rectangles, large, out sides *
0x00000010	fade smoothly	0x0153dc00	rectangles, large, out top/bottom *
0x00000014	fade, cover clockwise	0x0153dec0	rectangles, out *
0x00000015	fade, cover counter clockwise	0x04000003	scale in
0x00000019	fade, cover down	0x04000001	scale in, horizontal
0x00000016	fade, cover left	0x04000002	scale in, vertical
0x0000001b	fade, cover left-down	0x00000003	scale out
0x0000001c	fade, cover left-up	0x00000001	scale out, horizontal
0x00000017	fade, cover right	0x00000002	scale out, vertical
0x0000001a	fade, cover right-down	0x83f7c040	scanlines, crossing horizontal *
0x0000001d	fade, cover right-up	0x83f7c280	scanlines, crossing vertical *
0x00000018	fade, cover up	0x83f3c280	scanlines, down and back *
0xfbdfcd00	fade, little waves	0xc3f3c040	scanlines, left and back *
0xfbefcd00	fade, pixel	0x83f3c040	scanlines, right and back *
0x04000015	fade, uncover clockwise	0x83f7c284	scanlines, rotating *
0x04000014	fade, uncover counter clockwise	0xc3f3c280	scanlines, up and back *
0x4013fd80	flip fold down *	0x4111dd80	slice down *
0x4013ff40	flip fold left *	0x4111df40	slice left *
0x4013fe40	flip fold right *	0x4111de40	slice right *
0x4013fc80	flip fold up *	0x4111dc80	slice up *
0x6381dcc0	meander *	0x4381dc00	snake in *
0x6381dc00	meander [block style] *	0x4381dd00	snake out *
0x0013fc00	oval in [block style] *	0x000fc280	split horizontal in *
0x0013fd00	oval out [block style] *	0x000fc380	split horizontal out *
0xc013fd00	pinwheel, clockwise *	0x000fc040	split vertical in *
0xc013fe00	pinwheel, counter clockwise *	0x000fc140	split vertical out *
0xc013fdc0	pinwheels, clockwise *	0x4111df00	strips left-down *
0xc013fec0	pinwheels, counter clockwise *	0x4111dc00	strips left-up *
0x0400000f	push down	0x4111dd00	strips right-down *
0x0000000e	push left	0x4111de00	strips right-up *
0x0400000e	push right	0x0013dd00	sunrise [block style] *
0x0000000f	push up	0x0013dc00	sunset [block style] *
0xfbffffd40	random bars, horizontal	0x4111ffc0	triangles down *
0xfbffff59	random bars, horizontal, cover down	0x6111fdc0	triangles left *
0xfbffff58	random bars, horizontal, cover up	0x4111fdc0	triangles right *
0xfbffffd00	random bars, horizontal, large		
0xfbfffff80	random bars, vertical		
0xfbffff96	random bars, vertical, cover left		

## KoolMoves User Manual

0x6111ffc0	triangles up *	0x4013fd00	wedge down [block style] *
0x04000008	uncover down	0x4013fc00	wedge up [block style] *
0x04000007	uncover left	0x8013fc00	wheel clockwise, 1 spoke [block style] *
0x0400000d	uncover left-down	0x8013fd00	wheel counter clockwise, 1 spoke [block style] *
0x0400000a	uncover left-up	0x0007c280	wipe down *
0x04000006	uncover right	0x0007c140	wipe left *
0x0400000c	uncover right-down	0x0007c040	wipe right *
0x0400000b	uncover right-up	0x0007c380	wipe up *
0x04000009	uncover up	0x4011fe00	wiper, bottom [block style] *
0x4193dc00	unzip down *	0x6011fe00	wiper, top [block style] *
0x4111fc00	unzip left *	0x4113dd00	zip down *
0x4113de00	unzip right *	0x4113df00	zip left *
0x4113dc00	unzip up *	0x4111fd00	zip right *
0x8113fcc0	waves down *	0x4193dd00	zip up *
0xa111fcc0	waves down 2 *		
0x8113fec0	waves left *		
0xa111fec0	waves left 2 *		
0x8113ffc0	waves right *		
0x8111ffc0	waves right 2 *		
0x8113fdc0	waves up *		
0x8111fdc0	waves up 2 *		



## Appendix C – Programming / Coding Advice et.al.

### **ActionScripting**

KoolMoves provides up to Flash 8 ActionScripting capability (but not AS2). A large fraction of the possible Flash 5, 6, 7 and 8 operators and commands are supported at this time. It is very important to note that Flash 7 and 8 are case sensitive (e.g. gotoandplay will not work when exported as Flash 7 or Flash 8).

Some Flash 4 syntax is not supported. The only valid target syntax is the dot syntax. For example, Flash 4 syntax `_level0:clip` and `_level0/clip` should be represented as `_level0.clip`.

There are a number of ActionScript examples in the Examples folder. [actionscript.toolbox.com](http://actionscript.toolbox.com), [www.actionscripts.org](http://www.actionscripts.org), and [http://www.adobe.com/support/flash/action\\_scripts.html](http://www.adobe.com/support/flash/action_scripts.html) are good sources of information. Definitive Guide to ActionScript by Colin Moock is one of the best books on the subject. Another approach is to learn JavaScript because Flash ActionScript is basically JavaScript. This is a good guide to JavaScript: [www.oreillyn.net/pub/a/javascript/2001/%2012/07/action.html](http://www.oreillyn.net/pub/a/javascript/2001/%2012/07/action.html)

Note that if you use gotoAndPlay or gotoAndStop actions and if you use a KoolMoves' Preloader, 2 frames are automatically added by KoolMoves at the beginning of the movie.

Also note that in gotoAndPlay, the first frame of the movie is 1 not 0. It is less error prone to name frames and use frame names in goto actions rather than frame numbers.

Stop movie action should not be placed on the first frame. In fact there is error checking to prevent this. A stop action on the first frame is often ignored by the Flash player.

With Flash 6, dynamic text has both an instance and a variable name in the Flash authoring tool. Object attributes like 'text' and '\_x' are associated with the instance name which in KoolMoves is the name of the object, e.g., txt1. The variable name is used for backward compatibility for assignment statements such as `= "hello"`. The KoolMoves interface supports an instance name and a variable name. The variable name can be set in the object Properties window. By default, it is the instance name plus "var". For example, if the name of a dynamic text object is txt1, when exported as Flash 6/7/8 the variable name is txt1var and the instance name is txt1. For a Flash 4 and 5 exported dynamic text object named txt1, the variable name is still txt1.

These statements are valid for Flash 6/7/8 export:

```
txt1var = "hello";  
txt1.text = "hello";  
txt1._x = 200;
```

These statements are valid for Flash 4/5 export:

```
txt1 = "hello";  
txt1._x = 200;
```

## **Accessing Databases**

To access a database using ASP, see [www.haneng.com/lessons\\_21.asp](http://www.haneng.com/lessons_21.asp). To access a database using PHP, see [www.phpbuilder.com/columns/siddarth20000228.php3](http://www.phpbuilder.com/columns/siddarth20000228.php3) and [www.databasejournal.com/features/msaccess/article.php/1405201](http://www.databasejournal.com/features/msaccess/article.php/1405201). Another reference is [www.mysql.com/products/mysql/index.html](http://www.mysql.com/products/mysql/index.html).

## **Adding Flash Movies to PowerPoint**

[www.flashgeek.com](http://www.flashgeek.com) is devoted to the subject of adding Flash presentations to PowerPoint.

You can get a free tool from Microsoft at [www.globfx.com/products/swfpoint/](http://www.globfx.com/products/swfpoint/) that adds a Flash movie to a PowerPoint presentation.

Also see [www.itg.uiuc.edu/help/flash\\_to\\_ppt/](http://www.itg.uiuc.edu/help/flash_to_ppt/).

Alternatively, you can add it manually. The first thing you will need to do is have the control toolbox visible if you don't already. Select View > Toolbar > Control toolbox.

Next you select the More Controls button (The button looks like a hammer and a wrench) you should get a large list of controls - you want the Shockwave Flash Object control. (If it is not on the list, you do not have the Flash OCX installed and you need to install it from Adobe's website.)

Then you draw where you want your movie to play in the PowerPoint Presentation. The Movie will look like a big square with an X through it until you format it.

Right click on the square and then change the properties. Double click the Custom property to assign the correct URL of your movie to the PowerPoint presentation (remember URL's can be local- I suggest that you create a shortcut of your SWF and then copy and paste the URL (Path) from the shortcuts properties view).

Until you view the slideshow the SWF will display as a square with an X through it. This will change as soon as you view the slide show.

This works on both PowerPoint 97 and 2000.

## **Tips on Adding Flash Movies to Web Pages**

The .html file contains a large section of JavaScript that branches to a nonflash.gif if the browser does not have the Flash player or cannot read JavaScript. The nonflash.gif is not created by KoolMoves. You need to do that. Create your own image, in gif or jpeg format, for display when the Flash movie cannot play and change the name from nonflash.gif to whatever name you choose and change the name in the JavaScript file wherever it occurs. The size of the image is indicated next to the file name.

It is not guaranteed that an exported Flash™ movie can be imported into Adobe Flash™ for purposes of further editing or that a movie exported from KoolMoves, imported into Flash™,

and then exported from Flash™ will have the same behavior as the original movie exported from KoolMoves.

The Flash player can only handle SWF file sizes less than 2 1/2 - 3 MB. When a SWF becomes around 3 MB in size, it starts doing weird things. Many times it will crash the computer. This is explained here [http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_14437](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_14437). If you want to create a movie exceeding this size, consider using a lower quality jpeg setting, making the sound file smaller, or breaking the movie into pieces and using load movie to load the separate pieces.

The HTML contains simple instructions (called parameters), and executes the SWF file in the area of your web document where you insert the script. Here is an example of the Flash HTML script that you would need to add to your page:

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" codebase=
"http://active.macromedia.com/flash2/cabs/swflash.cab#version=4,0,0,0" ID=super WIDTH=560
HEIGHT=60>
<PARAM NAME=movie VALUE="super.swf">
<PARAM NAME=loop VALUE=true>
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src="super.swf" loop=true
quality=high bgcolor=#FFFFFF WIDTH=560 HEIGHT=60 TYPE="application/x-shockwave-
flash"
PLUGINSOURCE="http://www.macromedia.com/shockwave/download/
index.cgi?P1_Prod_Version=ShockwaveFlash">
</OBJECT>
```

Notice that there are both Object and Embed tags. The Object is for Internet Explorer and the Embed for Netscape. The parameters are referenced in both areas; and it is these parameters which allow you to do some customizing of your movie. [http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn\\_12701](http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_12701) provides further explanation of these tags and others which you can use to customize the display of your movie.

The movie parameter references the SWF from your directory. You'll need to upload the SWF file to the same directory as your web page. If you place the SWF file in a different directory, then you will need to show the path to that directory in this parameter -- something like flash/super.swf -- assuming flash is the name of that particular directory or folder.

With the loop parameter you can have the movie play once, or play continuously, by setting it to false or true.

The bgcolor parameter can be set to whatever hexadecimal (#FFFFFF format) color you need for your web design.

The width and height are not parameters in the same sense as the others -- but you can change these as well. Since the SWF file is scalable, you can make necessary changes and the movie will smoothly resize to the new dimensions.

Either fixed pixel-based or flexible percentage-based dimensioning can be used. Fixed dimensioning (for example, width=300 and height=200) sets the size to the exact pixel dimensions, just like you normally would with an image. However, percentage-based dimensioning -- for example, width=50% and height=50% -- makes the SWF automatically resize, proportionately, to any browser window size without any deformations.

## **Backend Code – Sources and Samples**

Check out these web sites for information on back ends.

<http://www.bignosebird.com/lscgi.shtml>

<http://worldwidemart.com/scripts/>

<http://www.cgi-resources.com/>

[http://cgi.resourceindex.com/Programs\\_and\\_Scripts/](http://cgi.resourceindex.com/Programs_and_Scripts/)

The front end is easy. All you need to do is set your URL with the Get or Post values to the proper URL and that's it. For CGI the URL will more than likely end in .CGI. When you use get or post it sends all the variables that have been set.

ASP code for load variables:

```
code:-----<%  
' initial setup (recommended)  
Option Explicit ' Forces declaration of variables  
Response.Buffer = true ' Don't output page until script finished  
Response.Expires = -1441 ' Attempt to overcoming browser cache probs  
  
' declare some variables  
dim strFirstName, strLastName  
  
' get the info  
strFirstName = "" & request.form("firstname")  
strLastName = "" & request.form("lastname")  
  
' process as desired  
%> -----
```

Where "firstname" and "lastname" are what the text fields are called in KoolMoves.

No need for first three lines, but they help. No need to DIM variables if Option Explicit not used.

Standard stuff, in other words KoolMoves creates a .swf that sends data correctly, as far as ASP is concerned anyway.

This info could then be used in html like this:

```
code:-----Hello <%=strFirstName%>
etc -----
```

More code in VB Script:

```
code:-----<%@
LANGUAGE="VBSCRIPT" %>
<% response.expires = 0 %>
<%
```

```
sMail = sMail & "You Can Put Anything You want here- It is attached to the Body" & vbcrLf &
vbcrLf
```

```
sMail = sMail & vtab & "AnyQuestion you want answered - " & request("Any Variable you
want") & vbcrLf
```

```
Set Mailer = Server.CreateObject("SMTPsvg.Mailer")
Mailer.RemoteHost = "your mail server"
Mailer.Subject = "anything you want "
```

```
Submitter = request("Var you set for the senders name")
Mailer.FromName = "any name you want"
Mailer.FromAddress = "any address you want"
```

```
Mailer.BodyText = sMail
```

```
Mailer.AddRecipient "name of who you want to send to", "E-Mail of person you want to send
to"
%>
```

Code in Perl

```
code:----- #!/usr/bin/perl
```

```
require "subparseform.lib";
&Parse_Form;
```

```
$Title = $formdata{'Title'};
$Contact = $formdata{'Contact'};
$About = $formdata{'About'};
$News = $formdata{'News'};
$Products = $formdata{'Products'};
$link = $formdata{'Link'};
```

```
@New =  
("Title=$Title&Contact=$Contact&About=$About&Products=$Products&News=$News&Link=  
$Link");
```

```
open (LOG, ">/home/formexample/TextFile.txt") || &ErrorMessage; print LOG "@New"; close  
(LOG);
```

```
print "Content-type: text/html\n\n"; print "Status=Success - Your Comments have been updated.  
Please return to the main area to see the results";
```

```
sub ErrorMessage { print "Content-type: text/html\n\n"; print "Status=Connection Failed Please  
Check the path to the text File"; exit; }-----
```

That Backend looks for the variables Title, Contact, About, News, Products, and Links and then writes them to a Text File called TextFile and generates a HTML Page telling you that the values were submitted. The Script works and is customizable. By Altering the names of the Vars in the First part of the script to mach the names you gave them in the SWF you created , remember to change them in the @New line.

## **HTML Coding**

### **Tags in Dynamic Textfields**

If a textfield has been to support HTML formatted content, either in the Properties window or dynamically with textField.html, the text can contain a subset of HTML formatting tags: <B>, <I>, <U>, <LI>, <FONT>, <P>, <BR>, <A>, <IMG>, <SPAN>, <TAB> and <TEXTFORMAT>.

### **Adding a Black Border**

Use the following HTML code to add a black border to your movie:

```
<div style="border: 1px solid black;">  
<!-- movie code here -->  
</div>
```

### **Using KoolMoves to make a HTML Signature in Outlook**

See this tutorial [www.koolmoves.com/johnie/outlook/KoolMovesandOutlook.html](http://www.koolmoves.com/johnie/outlook/KoolMovesandOutlook.html).

## Appendix D – Useful 3rd Party Software

### **Audio Editors**

This is a limited listing of freeware and shareware audio editors:

- DigiDesign, a very powerful freeware audio editor, [www.digidesign.com/flashnav/index.html](http://www.digidesign.com/flashnav/index.html)
- ALF, a WAV to MP3 encoder, [www.philnet.u-net.com](http://www.philnet.u-net.com)
- Sound format converter [www.dbpoweramp.com](http://www.dbpoweramp.com)
- ASIA, a free wave editor, [www.geocities.com/SiliconValley/Platform/9182/](http://www.geocities.com/SiliconValley/Platform/9182/)
- Audacity, a freeware audio editor, [audacity.sourceforge.net/](http://audacity.sourceforge.net/)
- Windows media player and Windows sound recorder can be used to create WAV files and convert many formats to WAV.
- GoldWave, shareware \$40, [www.goldwave.com](http://www.goldwave.com)
- Audio Mulch, shareware \$50, [www.audiomulch.com/index.htm](http://www.audiomulch.com/index.htm)

### **Converting SWF to AVI**

There are two methods.

Method 1: use SWF2Video, <http://www.flashants.com>, or SWF2AVI (freeware), <http://www.pizzinini.net/?category=8&item=37>. SWF2AVI converts any SWF into an AVI file. The AVI can then be exported as an Image Sequence from many Video Editing tools such as QuickTime. QuickTime is able to convert SWFs directly to image sequences.

Method 2: export as an image sequence and then make the AVI with your favorite AVI creation software (there are plenty of free ones) or convert to AVI using MS Movie Maker which comes with Windows XP.

To convert AVI to SWF, use CamStudio <http://www.rendersoftware.com/products/camstudio/>.

### **Flash Toolset**

This is an inexpensive product ([www.flashtoolset.com/FTS/features.html](http://www.flashtoolset.com/FTS/features.html)) which has many useful features:

- Automailer
- AS Editor
- Create Standalone Projector
- CD burner
- File operations
- and more

### **Vector Graphics Applications**

The following applications are recommended for saving drawings in a vector format that can be imported into KoolMoves: Sun Open Office (<http://www.openoffice.org>), PhotoLine 32 (<http://www.ciebv.com>), Adobe Illustrator, Adobe Photoshop, and Adobe Streamline. Shapes made with Metafile Companion almost always import faithfully into KoolMoves.

## ***Exe / Projector Creation***

You can use a freeware tool obtainable at [buraks.com/swifty/](http://buraks.com/swifty/) to create a .exe file. SWF Studio at [www.northcode.com](http://www.northcode.com) is another useful product.

[www.flashkeeper.com](http://www.flashkeeper.com) is a piece of shareware that costs \$29.99 and it can go either from EXE to SWF or the other way.

Another one that makes projectors is [www.flash-player.us](http://www.flash-player.us). It is \$19.99 for the full version 2.0 and has a free version and a trial version.

A whole list of them are at [www.soft411.com/software/swf-to-exe.html](http://www.soft411.com/software/swf-to-exe.html).

## Appendix E – Toolbar Library / Effect Listings

### ***Text Effect Templates***

Bounce in	Lateral motion 1	Shutter 2
Crazy shadows 1	Lateral motion 2	SPINREDOFF
Crazy shadows 2	open_doorL	Streak 1
Crazy shadows 3	open_doorR	Streak 2
Crazy shadows 4	open_doorR2	Streak 3
Crazy shadows 5	open_doorRL1	Streak 4
Crossover	open_doorRL2	Up down 1
Drop from sky 1	open_up2	Up down 2
Drop from sky 2	open_up2a	Up down 3
Drop from top and bounce	open_up2b	Vertical to horizontal 1
Enter from both sides	open_up2c	Vertical to horizontal 2
Enter on diagonal	open_up2d	Vertical to horizontal 3
eyequake1	Outline 1	Whirlpool
eyequake2	Outline 2	Wild motion 1
eyequake3	Outline 3	Wild motion 2
eyequake4	Outline 4	Wild motion 3
Fade in and rotate	Outline 5	Wild motion in and out 1
Five rows flashing	Outline 6	Wild motion in and out 2
FLIPFADE	Outline 7	Wild motion in and out 3
Float up	Radiation	Wild motion in and out 4
Float up with flip	Relfect 1	Wipe 1
Float up with rotate	Reflect 2	Wipe 2
get green_b	Ripple	Write on 1
go orange_b	roll_over_V	Write on 2
Jiggle 1	roll_over_V_VRS2	Write on 3
Jiggle 2	roll_over_Vboogie	Write on 4
Jiggle 3	roll_over_VHF	Write on with colors
Jiggle 4	roll_over_VrayL2	Write on with transparency
Jiggle 5	roll_over_VrayR2	Write on with twists 1
Jiggle with accordion 1	roll_over_VRS	Write on with twists 2
Jiggle with accordion 2	roll_over_VRS_V_VLS1	Write on with twists 3
Jiggle with transparency	roll_over_VVF	Write on with twists 4
Jiggle with twists 1	ROLL2RED1	Write on with twists 5
Jiggle with twists 2	Shutter 1	

### ***Web Interfaces***

archiboek1	clouds	loading intro
archiboek2	desserts	menuslide
bluewave	draw	music
bounce	fashion bizarre	pagemaster
brique bleu	folio	probe
classicgrid	green_navi	scan

scroll  
shapes  
slide  
station

videothek  
wavemenu  
website  
website2

website3  
wrappaper

### ***Clip Art Gallery Categories (Folders)***

Abstract  
Business  
Cartoon Characters 1  
Cartoon Characters 2  
Communications  
Computers  
Food  
Furniture  
Holidays & Events 1  
Holidays and Events 2

Household  
Internet  
Music 1  
Music 2  
Nature  
Odds & Ends 1  
Odds & Ends 2  
People & Clothing  
Recreation 1  
Recreation 2

Religious  
Scenery  
School & Play  
Sports  
Tools  
Transportation 1  
Transportation 2  
United States  
Weaponry

### ***Animated Clip Art***

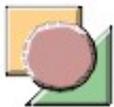
3dball  
3dglases  
butterfly  
lady  
lavalamp

light  
motionmeter  
neongrid  
rotatecube  
rotateglobe

rotatepolyball  
soundwaves  
stagecurtain

### ***Button Gallery Categories (Folders)***

#### **Abstract**



**Business & Communication & People**



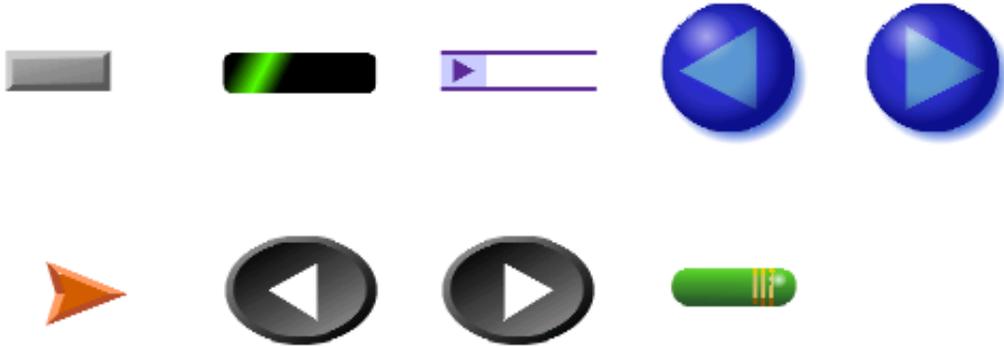
**Email & Internet**



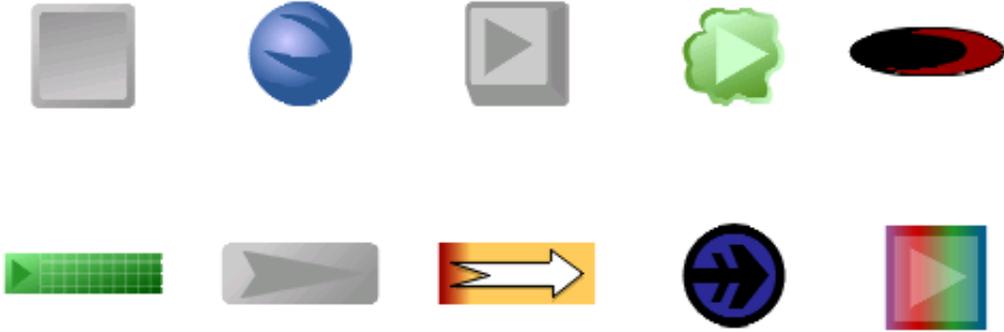
**Music**



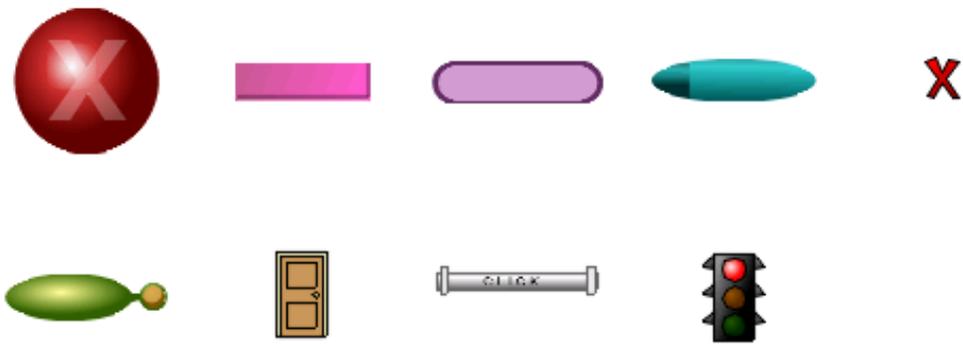
Navigation 1



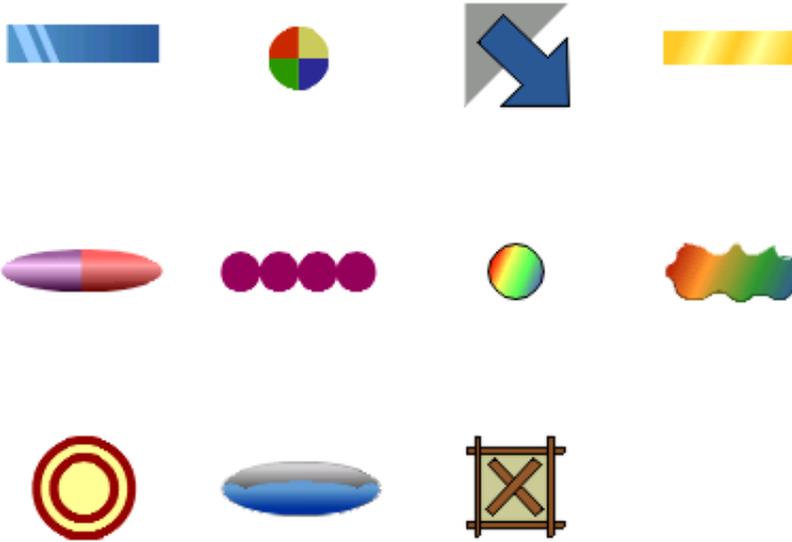
Navigation 2



Odds & Ends 1



## Odds & Ends 2



## ***Motion Scripts***

Alpha Bars  
Alpha Letters  
Bar Slide In  
Blinds  
Block Mask  
Blur  
Blur 2  
Blur and Fade  
Bounce  
Cartwheel  
Clockwise Sweep  
Cross  
Cursor Write On  
Decode  
Displacement  
Drop In  
Elastic  
Explode

Explosion  
Fall Forward  
Falling Leaves  
Flames  
Float to Center  
Flow on Curved  
Flow on Straight  
Flow on Stretched  
Frame  
From a Distance  
Growing Mask  
Jigsaw  
Jigsaw Tumble  
Light Projection  
Line Dissolve  
Matrix  
Merging Parts  
Motion Path 1

Mouse Sensitivity  
Mouse Trail  
Moving Bars  
Moving Mask  
Multi Fade  
Multi Wipe  
Opposing Motion  
Particles  
Punch Wave  
Rainbow  
Random  
Reveal One by One  
Ripple  
Ripples  
Rising Water  
Scratchy Video  
Shake  
Shapes

Simple Write On  
Slide In/Out  
Snake  
Sparkles  
Spin  
Spiral In  
Starburst  
Sweeping Arc  
Sweeping Highlight  
Ticker Tape  
Tumble In  
Twirl One by One  
Twirl 2 One by One  
Vanish  
Write On with Bar  
Zoom In  
Zoom Out

## ***3D Effects***

Arch  
Ascension  
Astroids

Back Flip  
Badge  
Boomerang

Bullet  
Clockwise  
Elastic

Disco Lights  
Dragon  
Helicopter

Interlace  
Jump Out Left  
Krazy  
Light Beam  
Orbital  
Pulse  
Punch Left  
Punch Wave

Recoiler  
Retroscope  
Rods  
Rods 2  
Roll In  
Roll Over  
Rotate 180  
Shuffle

Spiral  
Sticks  
Swarm  
Tarzan  
T-Birds  
Tornado  
Tossing  
Trail Blazer

Tumbling In  
Turning  
Twister  
Zed Rotor  
Zoom 180  
Zoom Y

### ***Filter Effects***

Adjust Color  
Bevel

Blur  
Drop Shadow

Glow  
Gradient Bevel

Gradient Glow

### ***Blend Mode Effects***

Normal  
Layer  
Multiply  
Screen

Lighten  
Darken  
Difference  
Add

Substract  
Invert  
Alpha  
Erase

Overlay  
Hardlight

## Appendix F – Frequently Asked Questions (FAQ)

### **Q. I can't find the answer to my problem here. Where can I find it?**

A. There is extensive help under Help in the program and on the support page of the KoolMoves web site. Try searching the KoolMoves forum at [www.flashkit.com](http://www.flashkit.com) using keywords. If you did not find an answer, post your question on the forum. Email [support@koolmoves.com](mailto:support@koolmoves.com) if you still need help.

### **Q. When I preview animations, the animations are missing. How can I fix this?**

A. Double-click on windowssecurity.htm in KoolMoves / Flash Help folder to read about possible causes for this problem.

### **Q. How do I add my animation to my web page?**

A. Use File > Export Movie. The first menu item, a wizard, will guide you through the process.

### **Q. Why can't I import my .fla files to KoolMoves?**

A. KoolMoves uses .fun as its project extension; fla is a proprietary Adobe format. You can import swf files into KoolMoves as editable movies.

### **Q. What are the best formats to use when importing graphic files?**

A. For ordinary picture files you can use jpg and for graphics with transparencies you can use gif and png files. 24 bit png format will give much crisper graphics than gif files although it does make the exported swf a bit larger.

### **Q. My movie will not display on my web page. Why not?**

A. Going through these steps will help to debug the problem: If you double-click on the html file on your local disk, does the movie play on the web page? Did you ftp the swf file to your server as binary? Is the file name spelled exactly the same on the server as in the html JavaScript? Is the file in the correct location (compare with the relative path in the html JavaScript)?

### **Q. My movie keeps looping. How can I get it to stop?**

A. Open Views > Sounds and Actions Overview and put a Stop command on the key frame where you want the movie to stop. Alternatively, set the looping option in File > Export Settings > Html.

### **Q. I've tried that, I want to stop my movie on key frame 1 but now it runs through the whole movie once and then stops on key frame 1.**

A. Flash will often skip commands placed on the first frame of a movie. It is best to add a copy of the first key frame and put your commands on key frame 2.

**Q. My movie takes 48 minutes to load on a T1 Cable/DSL connection and then my computer crashes. Does KoolMoves have a bug that causes this?**

A. Your movie is way too big. Try splitting your movie into small chunks that can load quickly when you need them. For example make a button that says Go to key frame 3 and Stop, then on key frame 3 in the Sounds and Actions Overview put Load Movie mymusic1.swf.

**Q. How do I make an entire swf a link on a web page?**

A. Put a shape the size of the movie frame filled with the background color in all frames, send it to the bottom and make it a button OR put a shape the size of the movie frame filled with a totally transparent color in all frames and make it a button.

**Q. What are Tweens?**

A. Tweens are non-editable movie frames that sit between the movie key frames. They are used to add movement to your movie or to add time between key frames. For example, if you draw a box at the top of your screen in key frame 3, Add Copy of Frame to End with 6 tweens to create key frame 4, and in key frame 4 move the box to the bottom of the screen. When you play back your movie, the box will smoothly slide from the top of the screen to the bottom.

**Q. I've made a flash banner in KoolMoves but I want to save it as an animated gif as well. Can I save it as a gif in KoolMoves?**

A. No, but if you go to File >Export, you can save your movie as frame captures. You can then recompile them in an animated gif compiler like Paint Shop Pro/Animator.

**Q. I'd like to study some examples of flash movies made with KoolMoves.**

A. Go to the main KoolMoves site [www.koolmoves.com](http://www.koolmoves.com), support page, where there is a list of different sites with downloadable examples.

**Q. My computer crashed and now KoolMoves needs to be reloaded. Where can I get the program?**

A. Email [support@koolmoves.com](mailto:support@koolmoves.com).

**Q. My Logitech mouse moves very slowly. How can I fix this?**

A. This is a windows setting. Uncheck checkbox "Disable acceleration in games" at Control Panel > Mouse > Motion tab or turn off the "Enable game detection" setting under Start > Settings > Control Panel > Mouse (or Logitech MouseWare program).

**Q. How do I select a shape or points when they are under another shape?**

A. Select the shape on top and then press CTRL H to hide the shape thus revealing what is underneath. Press CTRL H again to undo the hide.

**Q. How do I change the fill of a shape everywhere in the movie?**

A. Select the shape with the selection scope set to All Frames. Whatever you do to the shape is now reflected everywhere. You are prompted first before global changes are implemented.

**Q. How do I rename a shape, select a shape by name, or select a hard to get at shape?**

A. Use the Shape List tool. Using this tool, you can select individual shapes or multiple shapes with CTRL and SHIFT select.

**Q. Is there a way to combine KoolMoves, Swish, and Flash™ authored movies?**

A. Use the frame or button load movie action to load movies created by other tools or import the movies as objects.

**Q. How do I create a tiled background?**

A. Create a rectangle slightly larger than the movie frame in the first frame and send it to the back. Use the fill tool to fill the rectangle with your image. It will tile. Copy the rectangle, go to the second frame and use Edit > Paste to end of movie.

**Q. Is there a way to overcome Flash™ movie size (memory) limitation?**

A. The Flash™ player sets a limit on the size of the main swf. If you are including a large sound file, make it a separate swf and use load movie action to load it into the main movie. In this way you can effectively double the allowed size of the movie. This approach works in general for creating large movies. Moreover, breaking the movie into pieces results in faster internet loading.

**Q. What do I use to reposition the movie frame in the main editing window?**

A. Use Options > Reposition Movie Frame.

**Q. Why can't I get the Bones feature to work?**

Change your user Interface Style to Cartooning.

**Q. When I rotate something object shrinks as it rotates.**

A. The shape distortion is usually proportional to the number of degrees per Key Frame beyond a certain threshold. Two methods that work in eliminating this distortion: replace tweens with key frames to bring the number of degrees / KM below the threshold (ie, break up the rotation into smaller pieces) or use the Effects > Rotate menu command to rotate the object.

**Q. When I move points on shapes that I have drawn, the borders become detached from the shapes. How can I fix this?**

A. Uncheck checkbox "Enable OpenGL rendering" in File > Preferences, Drawing/Selection page.

Useful Flashkit Forum Threads.

You might like to review the following FAQ before posting to the KoolMoves forum (it explains a lot about the mechanics of posting to a forum):

<http://www.flashkit.com/board/showthread.php?t=599674>

Welcome to Flash Kit FAQ. Addresses the following: Posting Rules, Footer Rules, Footer Specs, Avatars and Titles, Post Count exception, Useful Forums, How to Become a Forum Moderator, and Multiple Accounts.

The following FAQ has a lot on sound file issue not related to KoolMoves:

<http://www.flashkit.com/board/showthread.php?t=649910>

Sounds and Music Forum > Common Sound Problems

Questions about, comments on, and/or suggested changes to the KoolMoves Forum Guidelines?  
Please Private Message (PM) pherbrick.

# Index

## A

Action Script... vii, 17, 29, 32, 33, 37, 44, 45, 63, 72, 73, 83, 87, 89, 91, 92, 95, 119, 121, 124, 128, 139, 147, 150, 151, 152, 153, 161, 163, 164, 166, 167, 168, 169, 171, 172, 174, 175, 176, 197, 205, 206, 207, 221

Action Script Editor..... 29, 87, 163, 164, 207  
Anti-alias..... 30

## B

Bevel..... 149, 175, 234  
Bones..... 12, 28, 30, 114, 237

Bones.....

    Double..... 117

Button..... 65, 78, 88, 95, 99, 106, 119, 120, 121, 122, 123, 126, 129, 133, 173, 175, 188, 189, 205, 206, 213, 214, 230

    Copying..... 119

    Gallery..... 129, 133, 230

    Hit Area..... 119

    Mouse Events..... 120

    Object..... 119

    States..... ii, 88, 119, 122, 123, 230

    Text..... 120

## C

Center..... 103, 158, 214, 216, 233

Change Depth..... 103, 214

Clip Art..... 129, 131, 132, 206, 213, 230

Clip Art.....

    Gallery..... 129, 131, 230

Color..... 23, 66, 67

    Background..... 23, 215

    Gradient..... 68, 69, 149, 157, 158, 177, 214, 234

    Transparency..... 23, 25, 66, 67, 69

Commands.....

    Load Movie..... 89, 90, 91, 125, 211, 236

    Morphing..... 76, 82, 152, 214

    Move..... 19, 96, 102, 104, 133, 191, 213, 214, 216

    Non-Linear Morphing..... 82

    Paste..... 106, 213, 216

    Print..... 166

    Text to Characters..... 142

    Text to Shapes..... 142

    Trace Image..... 107, 213

Components 32, 65, 161, 171, 172, 174, 178, 179, 181, 183, 186, 190, 191, 195, 199, 201, 203, 205, 214

Components.....

    Interface..... 32, 172, 178, 195, 197

    Liquid Style..... 177

    Shaded Style..... 177

Components.....

    Interface..... 32, 172, 178, 195, 197

    Liquid Style..... 177

    Shaded Style..... 177

Components.....

    Interface..... 32, 172, 178, 195, 197

    Liquid Style..... 177

    Shaded Style..... 177

Components.....

    Interface..... 32, 172, 178, 195, 197

    Liquid Style..... 177

    Shaded Style..... 177

Components.....

    Sounds and Actions..... 83, 235, 236

Drawing..... 25, 78, 82, 99, 101, 108, 110, 237

Drawing.....

    Border..... 66, 109, 173, 214, 226

    Border Thickness..... 66, 109

    Fill..... 66, 67, 109

    Line Thickness..... 110

    Reshaping..... 82, 110, 136, 155

    Tapered Lines..... 110

    Tools..... 12, 25, 101, 107, 108, 115, 136

    Tools.....

    3D74, 149, 159, 233

    Blend Mode..... 72, 77, 121, 150, 214, 234

    Drop Shadow..... 77, 149, 214, 234

    Filter..... 77, 121, 145, 149, 214, 234

    Menu..... 71, 150, 160

    Motion Script..... 71, 72, 73, 86, 129, 159, 214, 233

    Text Effect Templates..... 129, 142, 229

    Transition..... 75

Export Movie..... 55, 58, 59, 137, 213, 235

Export Settings..... 32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

    32, 33, 47, 171, 211, 213, 217, 235

Export Settings.....

Movie Clips.....17, 65, 77, 95, 99, 121, 122, 145, 147, 148,  
149, 151, 152, 165, 178, 183, 187, 188, 195, 205, 207,  
214, 215

**O**

Onion Skinning..... 30, 215

**P**

Pan Canvas.....29

Perspective..... 156, 214, 216

Print..... 128

**R**

Rotate.... 69, 76, 77, 121, 155, 157, 158, 159, 214, 216, 234,  
237

**S**

Scale..... 69, 139, 155, 157, 158, 159, 214, 216

Scenes.....15, 19, 26, 97

Screen Components.....11, 13

Select.... 15, 36, 40, 41, 44, 55, 56, 60, 68, 69, 88, 102, 104,  
106, 107, 110, 129, 132, 156, 205, 206, 213, 214, 216,  
222, 236

Selecting..... 33, 40, 101, 102, 124, 151

Selection.... 25, 28, 81, 83, 102, 103, 104, 105, 173, 213,  
214, 237

Shapes. 12, 13, 40, 63, 65, 67, 69, 78, 82, 99, 101, 102, 104,  
106, 108, 109, 110, 111, 116, 121, 141, 147, 156, 159,  
203, 205, 206, 213, 214, 227, 233, 237

Shortcuts.....216

Sounds..... 17, 26, 33, 83, 87, 88, 89, 91, 95, 123, 124, 128,  
163, 165, 167, 207, 213, 215, 216, 235, 236, 238

Symbol.. 78, 79, 85, 121, 147, 161, 165, 171, 174, 179, 203,  
205, 206, 207, 213

Symbol Library.. 85, 165, 171, 174, 203, 205, 206, 207, 213

**T**

Tell Target.....128, 136

Text.....

    Dynamic.....12, 32, 91, 139, 142, 143, 145, 214, 226

    Effect.....35, 48, 129, 141, 142, 213, 229

    Fonts.. 25, 29, 48, 67, 129, 139, 140, 142, 143, 175, 180,  
    182, 189, 193, 195, 199

    Letter Spacing..... 140

    Static..... 40, 139, 141, 142

    TextFields..... 139, 143, 226

Toolbars.....

    Menu.....viii, 14, 39, 51, 53, 63

    Tools...viii, 12, 20, 32, 63, 67, 71, 78, 99, 101, 122, 129,  
    139, 206

Tutorials..... 9, 35, 215

Tutorials.....

    Interactive..... 35, 36, 215

    Online..... 37

Tweening.....16

Tweens..... 16, 17, 95, 215, 236

**U**

Unselect..... 104, 214, 216

**W**

Web Interface..... 129, 213, 229

Wizards.....vii, 12, 20, 35, 38, 39, 40, 43, 55, 213

    Configuration..... 13, 35

    Simple Task Assistant.....35, 36, 215

    Slide Show.....44

**Z**

Zoom..... 29, 71, 75, 215, 216, 233, 234