

LIVESTAGE™
PROFESSIONAL
QUICKTIME AUTHORIZING ENVIRONMENT

Copyright © 2001 Totally Hip Software Inc. All rights reserved.

LiveStage Professional User's Manual.

This manual and the software described in it are furnished under license and may only be copied under the terms of such license. The user manual is furnished for information only and is subject to change without notice. No part of this publication may be reproduced, stored for retrieval, or transmitted in any form or by any means, electronic, photocopying, recording or otherwise without the prior written permission of Totally Hip Software Inc.

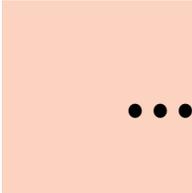
Totally Hip Software Inc. assumes no responsibility or liability for any errors, omissions or inaccuracies that may appear in this manual.

Terms of the Totally Hip Software Inc. LiveStage Professional Software License Agreement are included on the CD-ROM.

Totally Hip, the Totally Hip Logo, LiveSlideShow, LiveStage, LiveStage Professional and WebPainter are trademarks of Totally Hip Software, Inc. Apple, the Apple logo, Macintosh, the Macintosh logo, QuickTime and the QuickTime logo are trademarks of Apple Computer, Inc. Windows and the Windows logo are trademarks of Microsoft Inc., Acrobat is a trademark of Adobe Systems, Inc. All other trademarks are the property of their respective holders.

Totally Hip Software, Inc.
201-1040 Hamilton Street
Vancouver, British Columbia, Canada, V6B 2R9

Web: www.totallyhip.com
e-mail: support@totallyhip.com



Contents

Chapter 1 Introduction.....1

Welcome..... 3

About This Manual..... 3

What is LiveStage Professional? 4

Contents of the LiveStage Professional CD..... 4

Registration..... 5

About Totally Hip Software Inc. 5

Chapter 2 Getting Started7

Where To Get Information..... 9

System Requirements..... 10

Before Installing LiveStage Professional..... 10

Installing LiveStage Professional..... 11

Chapter 3 LiveStage Professional Basics13

What is LiveStage Professional used for? 15

What is QuickTime?..... 16

What is QuickTime Pro? 16

Common Terms 17

What is a Track?..... 17

What is a Media Sample?..... 18

What is a Sprite?..... 19

What is Sprite Animation?..... 20

What is QScript?..... 20

Chapter 4	The LiveStage Professional Environment	21
	Overview	23
	Project Window	23
	Stage Window	33
	Library	37
	QScript Reference	43
	Debug Console Window.	44
	Menu Overview	47
	Preferences	61
Chapter 5	Working in the Tracks Tab.	67
	Overview	69
	Creating Tracks	69
	Selecting Tracks and Media Samples.	72
	Common Track Properties	72
	Common Visual Track Properties	75
	Editing Tracks and Media Samples	78
	Track Order	79
	The Playhead	79
	Selection Properties View	80
	Media Samples	80
Chapter 6	Scriptable Tracks	83
	Overview	85
	Event Handlers	85
	Script Edit Field	87
	Script Management.	88
Chapter 7	Flash Tracks	89
	Overview	91
	Creating Flash Tracks.	91

	Flash Track Properties	92
	Flash Sample	92
	Interoperation with Stage	94
	Scripting Flash Buttons	94
Chapter 8	Sprite Tracks	95
	Overview	97
	Sprite Track Description.	97
	Creating a Sprite Track.	98
	Sprite Track Properties.	98
	Sprite Samples	100
Chapter 9	Text Tracks	115
	Overview	117
	Text Track Properties	117
	Using Text Tracks	118
	Text Tab	119
	Properties Tab.	123
	Layout Tab	125
	Chapter Tracks	128
Chapter 10	QTVR Movie Tracks	129
	Overview	131
	What is a QuickTime VR Movie?	131
	Using VR Movies	132
	VR Track Properties.	132
	VR Track Samples	133
	General Properties	134
	Scripting VR Tracks.	137
	Custom Cursors	137

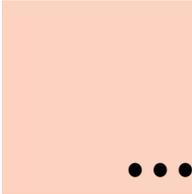


Chapter 11	Non-Scriptable Media Tracks	139
	Overview	141
	Standard Non-Scriptable Media Tracks	141
	External Tracks	141
Chapter 12	Instrument Tracks	143
	Overview	145
	Creating an Instrument Track	145
	Adding Instruments	146
	QuickTime Synthesizer Instruments	147
	Sampled Instruments	148
	Listening to Your Instruments	148
	Instrument Track Properties	149
	Using Instruments	149
	Exotica	151
Chapter 13	Effect Tracks	153
	Overview	155
	Creating an Effect Track	155
	Setting the Effect Properties	156
	Transitions	157
	Filter Effects	158
	Special Effects	160
Chapter 14	Picture Tracks	161
	Overview	163
	Creating a Picture Track	163
	Picture Track Properties	164
	Picture Sample Properties	164
Chapter 15	Color Tracks	167
	Overview	169

	Creating Color Tracks	169
	Color Sample Properties	169
	Gradient Editor	171
Chapter 16	Movie Tracks	173
	Overview	175
	Creating Movie Tracks	175
	Movie Track Properties	175
	Movie Samples	176
	Using a Movie in Your Scripts	179
Chapter 17	Skin Tracks	181
	Overview	183
	Creating Skin Tracks	183
	Skin Track Properties	184
	Skin Samples	184
Chapter 18	Streaming Tracks	187
	Overview	189
	Adding Streaming Tracks	189
	Streaming Track Properties	189
	Streaming Track Samples	190
Chapter 19	Data Generating Tracks	191
	Overview	193
Chapter 20	Tween Tracks	195
	Overview	197
	What is a Tween?	197
	Tween User Interface	198
	Tween Editor	199
	Tween Types	203



	Tutorial	209
Chapter 21	Modifier Tracks	213
	Overview	215
	Modifier Track Properties	215
	Adding Sequences	216
	Using Modifiers	217
	Modifier Types	217
Chapter 22	Introduction to QScript	219
	Overview	221
	QScript Basics	221
Chapter 23	Behaviors	239
	Overview	241
	Using Behaviors	241
	Pre-Defined Behaviors	242
	Authoring Behaviors	247
	Some Guidelines	252
Chapter 24	XML	257
	What is XML?	259
	What does XML look like?	259
	How does LiveStage utilize XML?	260
	What are the uses of the LiveStage XML Project Files?	260
	LiveStage XML Project Files	261
	The Basic Movie Script	263
	Referring to Movie Tracks in XML	264
	Glossary	267
	Index	277



Chapter 1 Introduction



Welcome	3
About This Manual	3
What is LiveStage Professional?	4
Contents of the LiveStage Professional CD	4
Registration	5
About Totally Hip Software Inc.	5

Welcome

Welcome to LiveStage Professional – QuickTime authoring environment. LiveStage Professional is the tool you need to create professional looking interactive QuickTime media for delivery via the Web, CD-ROM, or any other medium that allows QuickTime movie playback.

You'll find that LiveStage Professional is an extremely powerful and flexible tool giving unlimited creativity to the user. Unleash your creativity with this comprehensive QuickTime media creation and integration environment.

About This Manual

The LiveStage Professional User's Manual includes information about the product, the QuickTime media types it supports and details of how LiveStage supports and works with those media types. The User's Manual provides detailed information about the LiveStage Professional tools, commands and a QScript reference guide. It is designed to be used as a reference guide in your everyday work with LiveStage Professional and is also included on the CD-ROM in PDF format.

Tutorials showing the basic operation of the product and content creation methods are installed in the "Tutorials" folder along with their associated projects.

You will need Adobe Acrobat Reader 4 or later to view the PDF version of the manual and the tutorials. All illustrations in the manual are taken from the Mac OS X version of LiveStage Professional.

This manual assumes you have a working knowledge of your computer's operating system. For help or information on the operating system please see your operating system documentation.

- WebPainter 3 Software by Totally Hip Software Inc.
- QuickTime 5 Software by Apple Computer
- Adobe Acrobat Reader Software by Adobe
- CarbonLib Folder containing installer for CarbonLib 1.2.5

Registration

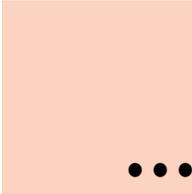
Totally Hip Software Inc. requires you to register your software so that we can provide you with technical support and send you important LiveStage Professional related information. Register your copy online by going to www.totallyhip.com/permanent/register.html.

About Totally Hip Software Inc.

Totally Hip Software Inc., founded in 1995 is a developer and marketer of interactive Web development tools. The company is located in Vancouver, British Columbia, Canada and is a public company (Trading Symbol: CDNX:THW) that is traded on the Canadian Venture Exchange (www.cndx.ca).

For more information and current news about Totally Hip Software you can check the Totally Hip home page at www.totallyhip.com or send an E-mail to info@totallyhip.com.





Chapter 2 Getting Started



Where To Get Information	9
System Requirements	10
Before Installing LiveStage Professional	10
Installing LiveStage Professional	11



Where To Get Information

We highly recommend that you read through the LiveStage Professional manual first, as it contains a lot of valuable information on using LiveStage's many powerful features. Many of the concepts and techniques used in LiveStage Professional QuickTime authoring may be relatively new to users and may take a little while to master. The LiveStage Professional CD, the Totally Hip Web Site (www.totallyhip.com) and LiveStage Developer Network (www.totallyhip.com/lsdn) provides additional sample projects that can provide further assistance for learning and deploying LiveStage Professional movies. Be sure to go through the tutorials as they will help you understand the product and its features better.

Users should also have a good knowledge and understanding of QuickTime and can go to www.apple.com/quicktime for up-to-date QuickTime resources and information.

If you are a registered user of LiveStage Professional, you may contact our tech support department at techsupport@totallyhip.com for any problems you may encounter with LiveStage Professional. They will be happy to assist you with installation support as well as answer questions about the use of LiveStage Professional. Please be sure to register LiveStage Professional before contacting technical support (see "Registration" on page 5).

To ask questions, share tips and tricks or discuss ideas, subscribe to the Totally Hip LiveStage Talk discussion list. This service is available to all users of LiveStage Professional. To sign up please go to www.totallyhip.com/permanent/support.html.

The LiveStage Developer Network (LSDN), at www.totallyhip.com/lsdn, provides information for the LiveStage Developer Community including sample script and projects, current software updates and news, tutorials, downloads to various products related to LiveStage and the archives to the LiveStage Talk list messages.

Ongoing LiveStage training seminars or courses are given by third parties in various locations. Visit the Totally Hip website and LiveStage Developer Network (LSDN) for current information.

System Requirements

Before installing LiveStage Professional, please ensure that the computer meets the minimum system requirements:

- Power Macintosh 333MHz or faster
- Mac OS 8.6 or later
- QuickTime 4.1.2 or later (full install)
- 20 MB of application RAM required
- 15 MB of hard disk space for basic installation
- Adobe Acrobat Reader 4.0 for PDF document reading

For the latest breaking news on LiveStage Professional, please see the Read Me document on the CD-ROM or visit the Totally Hip Web Site.

Before Installing LiveStage Professional

Before you install LiveStage Professional be sure that QuickTime 4.1.2 or later has been installed onto your computer. Be sure you perform a **FULL INSTALL** of the QuickTime components; this can be done through the “QuickTime Authoring” option in the QuickTime installer or by selecting all the components in the QuickTime updated.

To install QuickTime follow these steps:

- 1) Disable virus protection extensions and restart the computer.
- 2) Double-click **QuickTime Installer** to launch the installer.
- 3) Follow the installation instructions as they appear on the screen.
- 4) After completing the installation, restart the computer.

To install Adobe Acrobat Reader follow these steps:

- 1) Double-click **Reader Installer** to launch the installer.
- 2) Follow the installation instructions as they appear on the screen.



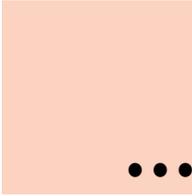
Installing LiveStage Professional

To install LiveStage Professional follow these steps:

- 1) Copy the LiveStage Professional folder from the CD to your Hard Drive.
- 2) Users not running Mac OS X should run the CarbonLib 1.2.5 Installer to install it into their system.

The first time LiveStage Professional is launched, you will be prompted to enter your name, company name (optional) and the product serial number. The serial number, included with the CD, must be entered exactly as shown, with all capital letters and no spaces.





Chapter 3

LiveStage Professional Basics

•••••

What is LiveStage Professional used for?	15	What is a Media Sample?	18
What is QuickTime?	16	What is a Sprite?	19
What is QuickTime Pro?	16	What is Sprite Animation?	20
Common Terms	17	What is QScript?	20
What is a Track?	17		

What is LiveStage Professional used for?

LiveStage Professional can be used to create engaging interactive presentations that encourage viewers to explore the designer's Web Site or anywhere the QuickTime movie is placed. Adding interactivity with LiveStage Professional is much easier and more powerful than creating JavaScript rollover effects or navigational image maps.

Let us look at a scenario. You are a Webmaster working on your site and you want to create a new navigation bar for your pages. Where do you start? Well, first we assemble the graphics needed, that's easy enough, but now we need to put those in a Web Page. HTML here we come. But wait, this doesn't give us those cool button rollover effects that all the other sites have, so time to dive into dynamic HTML, Java or JavaScript. After grinding away on your HTML, scripts and graphics you get something that works, but it doesn't work in half the browsers out there...

Does this sound familiar? Let's look at the same scenario with LiveStage Professional. You assemble the graphics in your favorite graphics program. Now you need to layout your navigation bar. No problem. Run LiveStage Professional and create a new project, drag and drop the images onto the stage and arrange them the way you want. Go into the properties for each of your buttons and tell LiveStage Professional what images to use for mouse over, mouse down, etc. With one line of script you tell each button what URL to display when it is clicked. Now you tell LiveStage Professional to create your movie and *voila*, your navigation bar is complete. Embed the self-contained QuickTime movie in your Web Page and you will have a new navigation bar.

The great part of this is if you want to change the graphics or URLs at any point in time you can just reload your LiveStage Professional project, edit the buttons to use new graphics or modify the URLs and regenerate the movie. Copy that movie to your Web Server and you can get on to more pressing tasks.

Creating those multimedia projects for distribution on CD-ROM or DVD-ROM is just as easy. Drag and drop the graphic content you want, add navigation buttons, tell LiveStage Professional what type of user interaction you want and you have interactive multimedia projects for presentations, classroom instruction, kiosks or the Internet. These movies play anywhere QuickTime plays!

If this sounds too simple then you will be pleasantly surprised with the ease of use and power that LiveStage Professional offers you. It is, however, a developer tool and not a graphics tool. Thus, for new users, their abilities and experience will determine their initial level of comfort and progress with the product.

What is QuickTime?

QuickTime is a recognized standard for delivery of rich media, digital video and streaming media, and is a leading Web Browser plug-in. QuickTime is superior in its capabilities for creation and playback of interactive content. Its movies can play in any program capable of embedding or supporting QuickTime.

QuickTime is an enabling technology comprised of components that extend the ability of the Macintosh or Windows operating systems to handle dynamic media. Totally Hip Software Inc. designed and developed LiveStage Professional to allow end users to harness the power of QuickTime technology and turn it into an interactive application development environment. The combination of LiveStage Professional and QuickTime allows you to build interactive content such as games, interactive ad banners, interfaces and training materials for CDs or the Internet.

QuickTime is much more than just video and sound. It is a true multimedia architecture that allows the integration of text, still graphics, video, animation, 3D, VR and sound. LiveStage Professional and QuickTime make it easy to bring these media types together in an interactive, compelling experience using the QuickTime browser plug-in. QuickTime is distributed free of charge by Apple Computer, Inc.

What is QuickTime Pro?

A QuickTime Pro license unlocks QuickTime Player, expanding the application from a media viewer to a powerful editing tool. It is not necessary for the viewing of your content but it may be needed for the ability to modify the content. However, LiveStage Professional does not require QuickTime Pro for playback or authoring.



QuickTime Player is an application that comes with QuickTime which allows you to open and play back all of the different QuickTime content types. The QuickTime Pro set of tools includes the fully featured QuickTime Player, the Picture Viewer tool, and the QuickTime Plug-In. QuickTime Pro may be purchased directly from Apple Computer Inc.

Common Terms

Make sure you understand the few basic terms and concepts used within this manual in order to make LiveStage Professional much easier to understand.

See the “Glossary” on page 267 for terms used within this manual.

What is a Track?

Tracks are the basic building blocks of a LiveStage Professional project and they are the containers of media. A Track contains one or more units of media called Media Samples. By incorporating multiple tracks you can create sophisticated effects like interactive video, scrolling text and synchronized sound.

Tracks are used to organize media in LiveStage Professional. You may any number of tracks within a project, but any given track can contain only one type of media sample. For instance, in a picture track you may have one or more pictures samples, but no audio samples.

There are visual tracks such as sprite, picture, video and text tracks. Visual tracks have an initial location and dimensions. Each visual track also has a layer and a drawing mode that are used by QuickTime to composite the tracks together and render onto the screen.

Audio tracks include digital audio (MP3, AIFF, Streaming, PureVoice, etc.) and MIDI songs that can be used for either ambient musical settings or user-triggered audio such as a click sound on a user interface control.

There are also tracks that act as data sources for other QuickTime elements. These tracks are neither directly seen nor heard by the end-user. Data source tracks, such as tween and modifier tracks, can be used to animate the motion of a sprite, fade-in of a picture, fade-out of audio, or directly triggering scripts within the project.

LiveStage also has compound tracks such as effect tracks that use other visual tracks as the source for on-the-fly video processing such as embossing, edge detection, or any of the NTSC transitions (for example, a cross-fade or radial wipe). QTVR is another example of a compound track in LiveStage. QTVR uses multiple tracks to create an immersive visual panoramic environment or a tangible 3D object a user can rotate, tilt and scale within your presentation.

LiveStage supports tracks which are editable, augmentable or non-editable. Editable tracks are those in which the user has the ability to change or edit the media samples. Sprite, Instrument, Effect, Tween, Modifier, Picture, Color, Text and Skin tracks are just some of the fully editable track types within LiveStage Professional. Augmentable tracks allow the users to add or modify script within the media but can not change or edit the media sample. Flash, Streaming and VR tracks are augmentable. Non-editable tracks are those which can be brought into a LiveStage project but may not be directly changed or edited. These include video, sound, QuickDraw 3D and some other QuickTime-support media types.

The most powerful track types that LiveStage supports are the scriptable tracks. These are the Flash, Sprite, Text, and QTVR tracks. Scriptable Tracks are capable controlling other tracks and objects in the movie driven by system and user events. The Scriptable Tracks are Flash, Sprite, Text and QTVR tracks.

What is a Media Sample?

Media samples are indivisible units of data for a particular media type. For example, each media sample of a picture track would contain a single image.

Some track types are able to reuse settings from previous samples to reduce the size of the movie file. For instance, in a text track if samples share the same text and only the style changes then LiveStage will export the movie with one copy of the text and the time based changes to the style. This compression technique is called sample differencing.

Media samples in some track types do more than just contain chunks of media. They can provide interactivity within the samples via LiveStage Professional scripting capabilities. For example, the sprite track contains sprite samples that utilizes the scripting power within LiveStage.

Media samples organize the loading and unloading of media. When the movie playhead is not within the duration of a sample, that sample may be unloaded to conserve computer resources such as memory.

Media samples also determine how much data must be transferred (downloaded) from the movie file to the end-user before the movie is visible.

Understanding media samples is essential when optimizing QuickTime files for Internet delivery.

What is a Sprite?

Traditionally Sprites are known as visual objects that can be animated and moved around on a computer screen. In LiveStage Professional this definition still holds true, however there are important additions to what Sprites may do in LiveStage Professional.

Think of Sprites as actors performing in a theatrical production, they can move around, change appearance and interact with other Sprites. They do offer an additional piece of functionality however that makes them extremely useful, Sprites have the ability to respond to and manipulate their environment. A Sprite can monitor the status of a stream or the loading of a child movie and provide animated feedback.

Sprites can resize other media tracks, switch panorama nodes or animate a swing in camera perspective. Sprites can play complex MIDI passages or change the playback rate of movie. Whether visible or not, Sprites are powerful tools in the creation of interactive QuickTime movies.

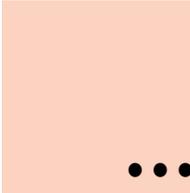
What is Sprite Animation?

Sprite animation is different from traditional video animation. With traditional video animation, each frame of the animation is a single image. The images are pre-defined, so nothing can change when they are played back. By contrast, with sprite animation, each frame of the animation is generated (composited) at the time it is displayed. Runtime compositing allows the animation to be different each time. Sprite animations can be altered from frame to frame and have infinite durations.

What is QScript?

QScript is a simple object oriented scripting language created by Totally Hip Software. QScript provides the ability for wired movies to manipulate themselves by responding to real-time events (i.e. mouse clicks, etc.) and changing the behavior of the movie.

The syntax of QScript is very similar to many of the common programming languages such as JavaScript, Basic, Lingo and C++. People with prior experience with any of these programming languages should find QScript familiar and easy to learn.



Chapter 4

The LiveStage Professional Environment

Overview	23	Filters	39
Project Window	23	Media Preview	40
Tracks Tab	24	Media Tab	41
Defines Tab	28	Local Tab	41
Info Tab	29	Scripts Tab	42
Annotations Tab	32	QScript Reference	43
Stage Window	33	Debug Console Window	44
Library	37	Menu Overview	47
Library Toolbar	38	Preferences	61

Overview

LiveStage Professional is a project based environment which makes the creation of interactive QuickTime movies simple whether you are a developer or a content creator. LiveStage can incorporate many different types of media into your movies. Most media used in LiveStage Professional is imported from external files. The media files are created with tools such as Adobe's Photoshop, Apple's Final Cut Pro, and Macromedia's Flash. Other media types, such as QuickTime special effects or MIDI instruments are created within LiveStage.

Using LiveStage Professional involves creating a project, adding media, adding script or pre-built behaviors (as required) and compiling the project into a QuickTime movie. A project can be tested within LiveStage Pro or exported and tested in different playback environments.

When you first open LiveStage Professional you will be presented with a dialog box asking if you want to open an existing project or create a new one. After a file has been opened or created, at least one window, the Project Window, will be displayed. Other windows may be displayed depending on what was visible when LiveStage Professional was previously used. These windows may be hidden or shown by using the Windows Menu, or keyboard shortcuts. These Windows include the: Stage Window, Debugging Console, Errors Window, Library Window and the QScript Reference. These Windows can also be closed by clicking on the close box in the title bar.

*Note: Keyboard Shortcut: Command Y will bring up the Library Window
Command T will bring up the QScript Reference.*

Clicking on the close box of the Project Window closes the project. If you have made changes you will be asked if you wish to save them.

Project Window

Whenever a project is opened or created in LiveStage Professional, a Project Window is displayed. The Project Window graphically represents your project and is the organizational hub for most operations you will perform in the LiveStage Professional environment.

There are four tabs in the Project Window: Tracks, Defines, Info and Annotations. All editing of the projects tracks and samples is accessed through the Tracks Tab. The Defines Tab is used for creating any constants that you use in scripts. General movie properties are specified in the Info Tab. Annotation Information, copyright Information, system requirements and other details which the user wishes to put in the movie are placed in the Annotation Tab.

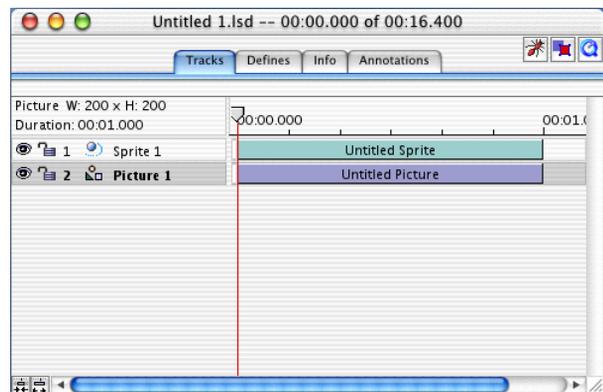
When each tab in the Project Window is clicked, the corresponding editor is displayed where you can view and edit that part of the project. The following section outlines the use of each of the four tabs in the Project Window.

Tracks Tab

The Tracks Tab presents you with a chronological view of your project's tracks and samples. The majority of the content creation and editing is initiated here and it is the primary area in which you will work.

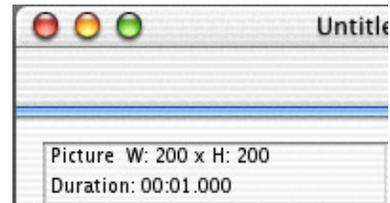
Tracks are created using the Create item in the Tracks Menu. To use previously prepared movies such as QuickTime or Macromedia Flash movies, drag them from the Library or the Finder directly into the Tracks Tab of the Project Window or use one of the items in the "Create" menu.

The timeline view presented in the Tracks Tab is used to coordinate the various media elements in your movie. Media Samples can be created through the Tracks Menu. These samples can then be dragged to new positions within the track that contains them or resized by dragging their ends.



Selection Properties Panel

The Selection Properties view displays information about the selected track or sample. If you have selected a track, the track type, height, width and duration will be displayed. When a sample is selected, the start time, end time and duration of the sample will be shown.



Timeline

The Timeline is a time scale ruler. Time increases as you move right. The Timeline also shows the Playhead.



Moving the playhead sets the current

time and simultaneously updates the Stage Window (described later). When a Track Sample is selected, the color of the Track Sample will change indicating that sample has been selected. A bar will appear across the tick marks in the Timeline indicating the time duration of the selected sample. The scale of the Timeline and sample graphics shown in the Tracks Tab may be adjusted using the zoom in and out buttons located at the bottom left of the window frame.

Track List

A list of tracks in the project is displayed in ascending index order along the left side of the window. You can adjust the width of this list by dragging the bar at the left of the Timeline.



You can reorder one or multiple tracks in this list by selecting the track(s) and dragging the selected track(s) to the desired location within the list. Dragging a track to a different position will change its index number but NOT its drawing order unless its layer number is the same as another track. The order of the tracks in the track list will only affect the drawing order of tracks that are on the same layer. In this case, the tracks are drawn in the order listed in the tracks tab, with the last track drawn appearing in front of other tracks in the same layer.

However, it is highly recommended that you control how you want the tracks to be drawn by setting the Drawing Layer in the Spatial Tab of the Track Property

Double clicking a media sample will bring up a Sample Editor specific to that media type (if the media sample is augmentable or editable by LiveStage Professional). The media sample's start and end times can be changed by altering the data within the Properties Window associated with that sample (if supported). Another way to do this is to drag the ends of the sample or drag its center. Holding down the option key while dragging a sample will snap it to the start or end times of samples in other tracks making it easy to align samples in multiple tracks. Media samples within a given track can not overlap with each other. External tracks like audio and video tracks can be stretched in time to line up with other tracks. This will alter the playback rate of the scaled media. Audio may change pitch and video may show motion artifacts. You can do this by dragging the right-most edge of the sample. You can also drag the offset bar at the left of the track to alter its offset so it will start at a time other than zero. This can be helpful if you need to offset all the samples in a track or if the track is an external track and does not allow the editing of the samples. Changing a track's offset affects the media loading order of QuickTime. Offsetting a track can be used to delay QuickTime's loading or transfer of media associated with the track.

Zoom In/Out

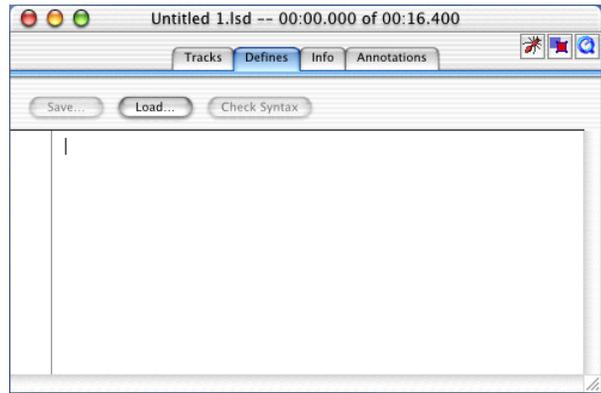
The zoom in and out buttons adjust the scale of the Timeline and sample graphics displayed in the Tracks Tab. These buttons are located in the bottom left corner of the Tracks Tab. You may also use the Zoom In or Zoom Out menu items from the View Menu.



Note: Keyboard shortcut: Command '[' and Command ']' will zoom in and out the scale of the Timeline displayed in Tracks Tab respectively.

Defines Tab

The Defines Tab is used for creating constants that will be used in any scripts you write in other parts of your project. This tab contains three buttons and a script editor field where the constants are entered.



Constants associate a name with a value in LiveStage Professional. For instance, if a project uses a URL, "http://www.myco.com/fanzine/" a constant could be created to contain the URL string so that it was located in one place convenient for checking and updating.

Constants are entered using the format: Constant Name = value. e.g.:
kNumCardsInDeck = 52. They are used in scripts by putting a "\$" in front of the constant name. This text will be replaced at compile time with the number or string.

You can also use complex names for your define if you enclose them in quotes like this: "Sprite track number 1" = 10. LiveStage will automatically add defines for all your images used in your Sprite Track. Since the names of your images can have spaces and other special characters, LiveStage will always enclose them in quotes.

So if you have an image named "Deck of cards - frame 1" you can refer to this image index like this: \$"Deck of cards - frame 1".

Your custom events (See "Scripts Tab" on page 42) are automatically entered into the defines for you so you can refer to your custom events by name. So if you have a custom event named "Initialize Deck" with ID of 1000, you can refer to it in two ways like this:

```
ExecuteEvent (1000)
```

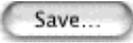
or like this:

```
ExecuteEvent ($"Initialize Deck").
```

You should always use a define instead of an actual number whenever possible. This allows you to quickly change the value of the define without having to search throughout all your scripts for places that you used that value. It will also make your code more readable and thus easier to understand.

For more details on QScript please refer to “Introduction to QScript” on page 219.

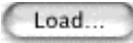
Save...



This button allows you to save the contents of the Script Editor Field to a QScript file which may be retrieved at a later time.

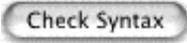
Saving QScripts in this way will help you reuse work from project to project. Often developers will place links in their application wide or local project library to a directory containing reusable scripts.

Load...



This button loads a previously saved QScript file into the Script Editor Field replacing any data in the script editor.

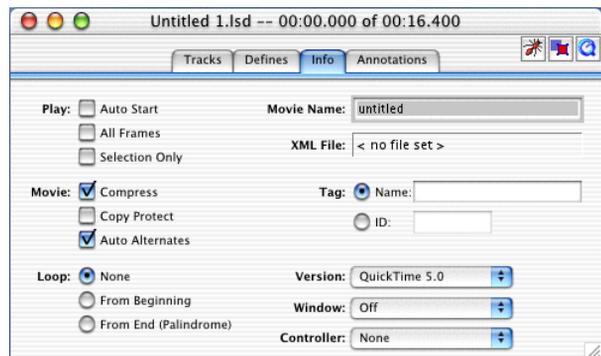
Check Syntax



Checks the syntax of the QScript in the Defines tab. If an error is found an Errors window will appear to notify you of the error. This will list the error and the line number of your script where the error was detected.

Info Tab

The Info Tab displays general properties of your project for viewing and editing. In this tab you can set up various movie playback properties. The properties displayed in this tab are placed into related groups. We will go through each group detailing their settings.



Play

The playback group contains properties that affect the playback of your movie.

AutoStart If you check this, the movie will automatically start playing when it is opened. Movies deployed on the Web will still need to use an autoplay attribute in their HTML to ensure a Web Browser will automatically play the movie when loaded.

All Frames If you check this, the movie will ensure that no frames are skipped when playing. Setting All Frames will disable all audio tracks. This may cause stuttering and other playback problems but can solve problems with Flash tracks skipping frames with scripts in them.

Selection Only If you check this, the movie will play only the portion that is selected. Currently the only way to make a selection is via a script.

This is most often used to restrict playback or create loops.

Movie

The movie section deals with all the properties associated with the actual creation of the movie.

Compress The movie resource will be compressed if this is checked. This will make a smaller movie which is particularly valuable for Web deployment.

Copy Protect Prevents users from saving and modifying this movie when checked. However, some browsers can still save the movie in the cache. While this protects the movie against modification it does not protect against the movie being copy from machine to machine.

Auto Alternates Checking this will allow QuickTime to automatically select tracks from alternate track groups. Alternate tracks are used to provide tracks for different languages or different bandwidth requirements. QuickTime selects the appropriate track to use when the movie is played.

Loop

The Loop section sets the looping properties of the movie.

None	Turns off looping.
From Beginning	Plays the movie looping either the current selection (if specified) or the entire movie. When the playhead reaches the end the playhead is moved back to the beginning to start anew.
From End (Palindrome)	Plays the movie loop either constrained by the selection or the duration of the movie. When the playhead reaches the end of the movie, playback will be set to reverse causing the movie to be played backward until the start is reached at which point the movie will once again play forward alternating in this fashion until the movie is stopped.

Movie Name

Specifies the name of the file created when you export the project to a wired movie.

XML File

The XML file specified will be included in your QuickTime movie. The XML file contains data for initializing the QTList of the movie (for more details, refer to the “QTList Properties and Action” in the QScript Reference).

The XML file can be dragged from the Library or the Finder and dropped into this location.

To remove an attached file, select it and then press the Delete key.

Tag

A movie file can be tagged with either name or a numeric ID. The tag will be embedded into the movie file and is independent of the movie file's name. Movie name or ID is used for intermovie communication. These fields should be ignored if you are not creating movies that communicate with each other.

Name Select this radio button in order to tag the movie with a name and then enter the name to be used in the field.

- ID** Select this radio button in order to tag the movie with an numeric ID and then enter the ID to be used in the field.

Version

Version determines the set of QScript actions and properties available to the project. If QuickTime 3 is selected, operations introduced with QuickTime 4 or greater will signal errors during the compilation of the project.

Window

Window selects amongst the present movie modes for playback in QuickTime Player.

Off play movie in a window.

Normal Size play normal size, but full screen with a black background.

Double Size play movie double size as a full screen presentation.

Half Size play movie half size in a full screen presentation.

Full Screen play movie scaled to user's screen size in full screen presentation.

Current Size play movie at current scale in full screen presentation.

Controller

Select the type of movie controller to use from this popup menu. For most movies either “None” or “Standard” should be used. For VR movies you should use “QuickTime VR”.

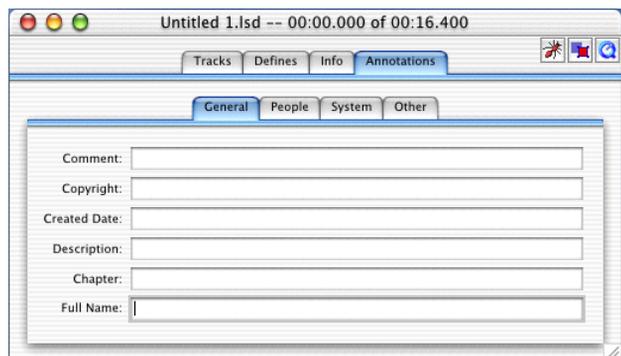
None the QuickTime controller will not be shown.

Standard Movie QuickTime’s linear media controller will be shown.

QuickTime VR QuickTime’s VR media controller will be shown.

Annotations Tab

Within the Annotations Tab there are fields for strings describing the content itself. Information can range from copyright to author/producer and keywords.



All annotations containing data will be embedded into the exported movie.

Annotations can be set from AppleScript for consistent and efficient workflow. There are four tabs in this location: General, People, System and Other. Some of this information is displayed in the QuickTime player, including Copyright, Created Date, Director and Info.

General Tab

The General Tab contains information on the full name, copyright, creation date and general information related to the actual movie.

People Tab

The People Tab contains information for the actors, writers and creators of the content in the movie.

System Tab

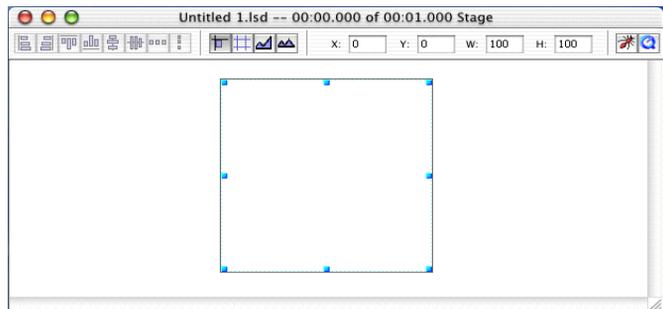
The System Tab contains requirement or recommendations for a system to ensure proper playback of the movie.

Other Tab

The Others Tab contains other miscellaneous information related to the movie.

Stage Window

The Stage Window shows a representation of all the visible elements in the movie and acts as a layout area. The Stage shows the time indicated by the playhead on the tracks tab of the



Project Window. The Stage does not display the contents of certain track types, such as QuickTime effects and movie tracks.

Visual tracks can be moved or resized within the Stage Window. Other visual items like sprites can also be moved around within the Sprites Track inside the Stage Window.

Objects Displayed in the Stage Window

Only visual tracks like Sprite Tracks, Picture Tracks, Text Tracks and the Skins Track are displayed in the Stage Window. The images representing the tracks are displayed for the entire duration of the movie even though the sample may not be active during the time that you are viewing. The methods for adding tracks to the project are explained in more detail in “Creating Tracks” on page 69.

Dragging and Dropping onto the Stage

Certain files can be dragged and dropped onto the Stage from the Library and the Finder and appropriate tracks will be created automatically. Dragging one or more images onto the Stage Window or the Tracks Tab will create a Picture Track containing the images you dropped. If you drag and drop one or more images onto a Sprite Track in the Stage Window, a Sprite will be created for each image.

You may also drag and drop folders from the Finder or the Library that contain images and an installer AppleScript. These AppleScripts will automatically create sprites using the images provided and populate the sprites with the appropriate scripts. There are several pre-made button creator AppleScripts included in the Library’s “Widgets” folder within the Media Tab. Drag one of the button folders onto a Sprite Track in the Stage Window and LiveStage Professional will automatically create a fully scripted button for you.

Using the Stage

The Stage not only shows a representation of the output, but also provides the capability to manipulate the size and position of visible tracks. The position and size of a track will remain the same throughout the entire movie.

Select the track by clicking on it in the Stage Window or selecting the track in the Tracks tab in the Project Window. The track will be shown selected with re-size handles.

Dragging the track anywhere in the Stage Window will move the track, clicking and dragging the handles will resize the track. The position and dimensions of the currently selected object will be shown at the top of the Stage Window.

Tracks in the stage can also be resized and repositioned by selecting them and entering the position and size at the top of the stage Window.

An object can be removed from the stage by selecting it and pressing the “delete” key.

If you want to quickly get access to an editor for a specific object in the Stage Window, double click on the object and the editor associated to that object will appear. If no editor is associated with that object then the media player will be displayed.

The position of a Sprite inside a Sprite Track can be changed within the movie. First select the Sprite Track. Next, click and drag the Sprite within the Track to move it around within the Sprite Track at that point in time.

Sometimes a Sprite within a Sprite Track may be larger than the Sprite Track. This makes it hard for you to select the Sprite Track when clicking on it and you will find that you will first have to move the Sprite and then the Sprite Track separately.

Another way is to hold down the Command and “Option” keys and then click on the Sprite Track. This will make sure LiveStage ignores all Sprites within the Sprite Track and select of the Sprite Track itself.

To make the Sprite move as time passes you will have to enable the preference “Auto Create New Sprite Properties”.

Groups of tracks and sprites can be selected and manipulated all at once. To select a group, hold down the shift key and click to select the Tracks you wish to move. You can then move the selected items by clicking and dragging them around the screen. You can also align selected items by using the buttons provided in the toolbar or the items in the Grid Menu in the Stage Window.

Toolbar

The Stage has a Toolbar across the top of its window. The Toolbar provides features that aid in viewing, aligning, position and resizing of the objects on the stage at a given point in time. For details on time refer to “Sprite Timeline” on page 104. The buttons are as follows:

Align Left Edges	Aligns the left edges of the selected objects.
Align Right Edges	Aligns the right edges of the selected objects.
Align Top Edges	Aligns the top edges of the selected objects.
Align Bottom Edges	Aligns the bottom edges of the selected objects.
Align Horiz. Centers	Aligns the horizontal centers of the selected objects.
Align Vert. Centers	Aligns the vertical centers of the selected objects.
Spread Horizontally	Horizontally spaces the selected objects evenly using the leftmost and rightmost sprites as edges.
Spread Vertically	Vertically spaces the selected objects evenly using the topmost and bottommost objects as edges.
Snap to Grid	Aligns objects to the grid when they are dragged around.
Show Grid	Displays / Hides the grid.
Zoom In	Enlarges the view of the objects.
Zoom Out	Shrinks the view of the objects.
X, Y	Sets the position of the selected object.
W, H	Sets the width and height of the selected object.

Library

The Library provides a convenient way to view and access media in LiveStage Professional. The Library allows you to drag and drop media into the project. It can be accessed from the Window Menu by choosing Library Window.

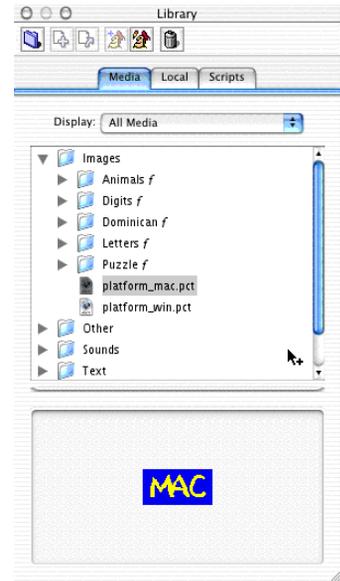
Note: Keyboard Shortcut: Command Y brings up the Library Window.

Use the Library to assist you in keeping the media for your projects organized. Each tab in the Library operates much like the Finder. You may create/remove files and folders using the Library. You can drag and drop items from the Library into different parts of LiveStage Professional. To reveal a file in the Finder:

“Control” and click any item in the Library Palette.

Each tab in the Library contain several standard folders to help you manage your content.

You can adjust the height of the list in the library and the Media Preview area by dragging the bar between the list and the Media Preview area up and down.



Library Toolbar

Media that is stored in the Library is accessed using standard drag and drop operations. To use images in the Library for a sprite you drag the image from the Library window into either the image list of the Sprite Sample Window or into a Sprite Track in the Stage Window.

If the media is a script it, can be dragged into any Script Editor and be added to the contents of that Script Editor.

The files in the Library are not stored within the document itself, only references to them are. The Library can access items on media storage devices such as a hard disk, CD-ROM, Network Server, etc. This linking between LiveStage Professional project and media files offers the most flexibility in providing access to the components of a movie.



New Folder

This creates a new folder and gives you the option of naming it. This gives you flexibility in defining custom folders to contain media for individual projects.



Add a Copy of a File

Use the following steps to add a media file to the Library:

- 1) Select the desired library folder where the new item is to be added.
- 2) Click the Add a copy of a file button. The “Choose a File” dialog will be displayed.
- 3) Select the file you wish to add as the new item in the Library.
- 4) Click Open or Choose to add the file to the Library. The new media asset will appear in the selected folder.



Add Aliases

As with Add a copy of a file, Add Alias allows files located outside the library to be accessible from within the Library. Add alias creates reference files to files located outside the Library folder and copies these reference files into the

Library folder. If the item in the Library is an alias, holding down the Control key while clicking the item will show the path to the original file. Use the same steps to add an alias of a media file to the Library as with adding the file (described in Add a Copy of a File).



Create a New Image

Tells LiveStage Professional to launch WebPainter so that a new image can be created. WebPainter is a graphics and animation tool from Totally Hip Software Inc. that is included with LiveStage.



Edit Current Image

Launches WebPainter and open the selected image file for editing.



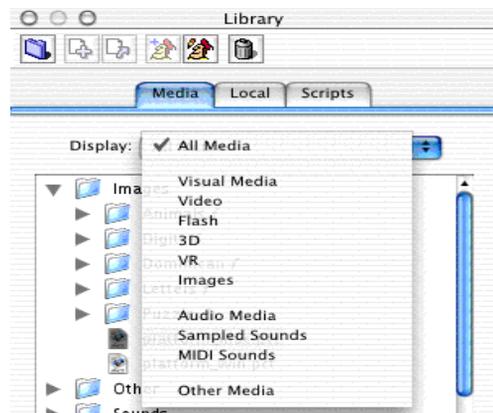
Delete Item

Removes the selected item from the Library. A standard dialog will be displayed asking you to confirm the deletion. If the item you are deleting is an alias, only the alias will be deleted. The original file will remain untouched.

Filters

Filters simplify the process of looking for content when you have a large number of files in your Library. The same set of filters is provided in both the Media and Local tabs. For example, you can filter out everything except for Visual Media or view only QuickTime VR media.

Filters are accessed through the Display popup menu located across the top of each tab in the Library.



This popup menu provides you with a variety of filters that allow you to display specific types of content within that particular tab. The contents of this popup menu will change depending on the tab you have selected.

Filtering does not affect the actual contents of your Library, only what is visible in the current tab. To prevent filtering select All Media from the Display popup menu.

Media Preview

The Media Preview is the display area located at the bottom of the Library Window.

The actual Media Preview will differ depending on the media being previewed. Audio media will display a movie controller and can be played to hear the audio. Still images will provide the image, while video or Flash movies will preview the movie combined with a controller for scrubbing through the file. QTVR content will preview the QTVR movie with a QTVR controller.



The file location of the media file is displayed just above the preview. If the Media Preview area is unable to display the media, the following text will appear: “Invalid media type selected or no selection”.

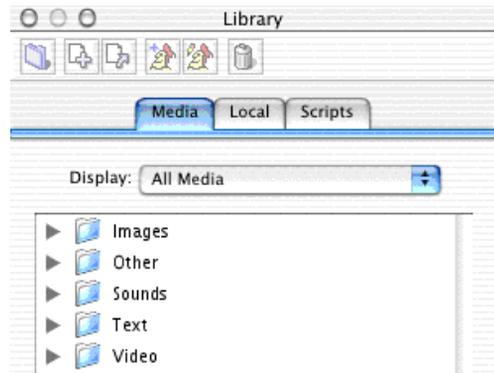
The preview area displays scripts and any image format that is supported by QuickTime.

With image types that have layers (Photoshop and GIF files), you can view each layer individually by expanding the image in the Library Window using the expansion triangle and clicking on the particular layer.

Double clicking in the Media Preview area for some of the media types will display that media type at full size in its own window.

Media Tab

All content or aliases to content displayed under Media are stored in the Library folder of the LiveStage Professional application. Any items added in this tab are available whenever you are running LiveStage Professional. All content displayed here is stored in a Library folder in the same folder as the LiveStage Professional application.



The media installed with LiveStage Professional are grouped into several folders. These include Images, Sounds, Text, Video and Widgets.

LiveStage Professional includes many images that are installed by default when you install the product. You will see all of this media in the Media Tab.

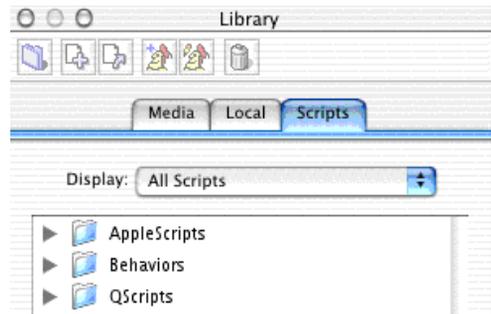
Local Tab

The Local Tab of the Library works the same as the Media Tab except it lists files and folders local to the current project. A Library folder is automatically created in the same folder as your project when you first save your project. You can store files that are only used in that project in this library folder. The files in this folder will be listed in the Local Tab. If there is no library folder created in the same location as your project file or if the library folder is empty, the Local Tab will display nothing.

LiveStage can keep track of external files better if they are stored in the Local Library folder by setting the preferences to search for media in your Local Library folder first. For details of how to set the preference refer to “Media Tab” on page 41

Scripts Tab

The Scripts Tab in the Library palette displays the list of custom scripts available to be used in the project. The custom scripts are commonly used routines. There are several pre-defined folders where scripts can be stored. The Script folders are AppleScripts, Behaviors and QScripts. It is possible to create sub-folders within the three main folders so those custom script files can be placed in an organized manner.



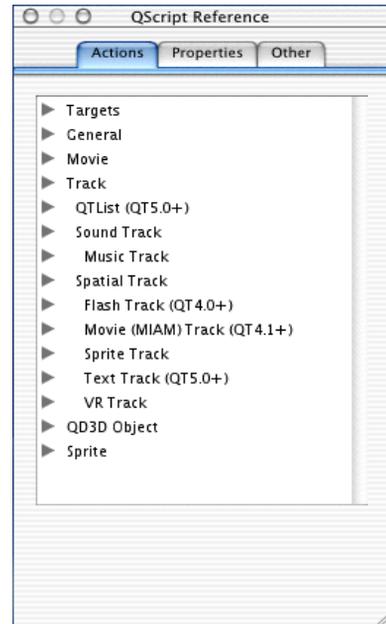
The Media Preview can displays the text for the selected script. To copy a script from the library into an edit field drag the icon representing the script from the library palette onto the script editing field. Sprite, Flash, QTVR, Text tracks, and the Define Tab can be augmented this way.

QScript Reference

The QScript Reference provides you with an online reference for the QScript language elements that are built into LiveStage Professional. This palette contains all of the QScript keywords including a summary of what each does and which parameters (if any) are used. It is accessed from the Window Menu by choosing QScript Reference (Command T).

Note: Keyboard Shortcut: Command T brings up the QScript Reference.

The QScript keywords are split into three main categories: the Actions Tab, Properties Tab and Other Tab. Each tab contains a set of folders which help with the categorization of items in that section. By expanding each folder you will see a list of the QScript keywords. If you select one of those items, details about that keyword will appear at the bottom of the QScript Reference Window. You can drag and drop these keywords into any Script Editor field as a shortcut to typing in the QScript keyword.



The Actions Tab contains QScript statements that perform actions. These include actions like the SetVisible and SpriteNamed keywords. The Actions Tab combines a target specification list. Target specifiers are used to specify what object a QScript action or property is applied to. The Actions Tab also contain a listing of all available QScript actions. Actions differ from properties in that actions do not return values.

The Properties Tab lists the available QScript properties that you can access within your movie. In here you will find items like the NumSprites and IsVisible keywords. The properties are grouped according to track type. These include the Movie Properties, Track Properties and Sprite Track Properties.

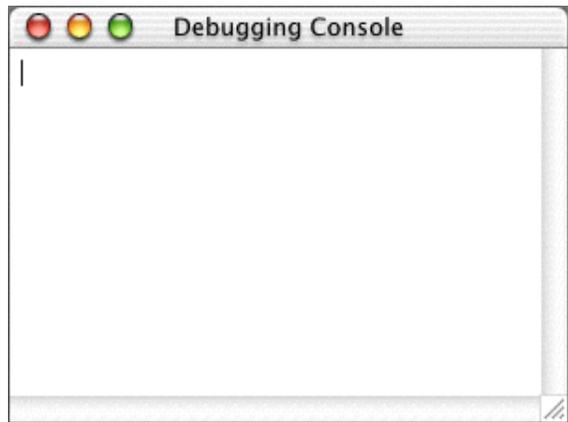
The Other Tab is a collection of loosely related topics regarding the QScript scripting environment. This section includes keywords for defining variables, operators, control statements and various constants.

Debug Console Window

The Debug Console Window displays debugging information while your movie is running in LiveStage Professional. This output can assist you in tracking down problems in your scripts by either viewing the script statements as they are being executed or through your own debugging messages using the DebugStr QScript action.

To view the Debug Console Window select the Debugging Console menu item in the Window menu or click on the  Debug icon in any window.

The Debug Console Window can display two types of debugging information. The first type is where the Debug Console Window displays the QScript statements that are being executed. To enable this functionality open the LiveStage Professional Preferences and check the “Default to Show source in debugger” item in the Compiler Tab. Now whenever you run a movie using the “Run Wired Movie” menu item (Command R) all of the QScript statements will be displayed in the Debug Console Window.



Note: Keyboard Shortcut: Command R will run a wired movie.

You can also use the #debug preprocessor directive to show or hide a portion of a script. With the “Default to Show source in debugger” setting in the Compiler Tab of the Preference window set, the “#debug off” can then be used to turn off the display of scripts in the Debugger Console. The following example shows how this command works.

```

#debug off
GlobalVars inc
SpriteVars tempstring

#debug on
// increment the counter
inc = inc + 1

#debug off
// create the string and then append
// the counter value to it
SetString(tempstring, "counter value: ")
AppendString(tempstring, inc, tempstring)

```

Since we have set the “Default to Show source in debugger” setting on for the compiler, all our scripts will automatically display the source in the Debug Console. The `#debug off` command turns off the source code display from that point on and the `#debug on` command turns on the source code display. The following piece of script will be displayed in the Debug Console Window.

```

// increment the counter
inc = inc + 1

```

Alternately, you can clear the “Default to Show Source in Debugger” setting in the Compiler Preferences and use “`#debug on`” and “`#debug off`” to enclose the lines of script you want displayed in the Debug Console. Another debugging method involves placing debug messages in your scripts that indicates which parts of your script had executed. Do this by using the QScript `DebugStr` action with either a string or numeric value. Any information passed by the `DebugStr` command will be displayed in the Debug Console Window, as in the following example.

```

DebugStr("Sprite was clicked on")

```

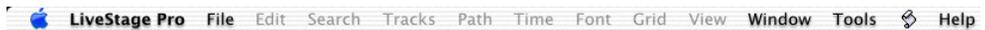


```
counter value: 2
counter value: 3
counter value: 4
counter value: 5
.
.
```

For more information on the DebugStr QScript action please refer to the QScript Reference.

Menu Overview

The following menu choices are available in the LiveStage menu bar: File, Edit, Search, Tracks, Path, Time, Font, Grid, View, Window, Tools, Apple Scripts and Help. Menu selection options may change depending on which window or palette is active.



In most windows there is a small tool bar which provides a shortcut to a few different menu items. The tool bar includes shortcuts to the “Show Debugging Console” menu item, “Show Stage Window” menu item and the “Run Wired Movie” menu item.

File Menu

The File Menu contains menu items that apply to files and documents. From here you can open, save and print your documents as well as do other file related operations.

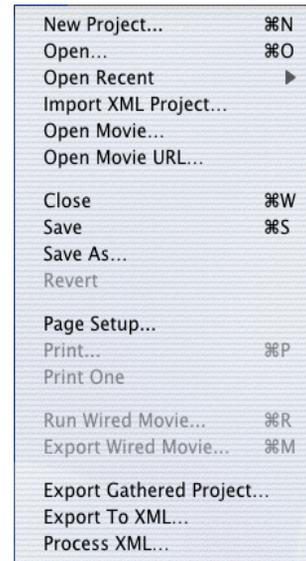
New Project (Command N) Creates a new Project Window with the Info Tab selected for editing the movie settings for the new project.

Open... (Command O) Brings up the Open dialog box for selecting a previously created project to open. In the Open dialog box, select the project file, and click the open button, or double-click the project name. This menu item will open LiveStage projects, QuickTime Movies and Flash Movie.

When you open a QuickTime movie or Flash movie file, LiveStage will create a new project that contains the tracks from that file.

Open Recent Display a popup menu listing the most recent project files opened by LiveStage. Selecting a file in the list will open that Project. If the document you are trying to open is already open in LiveStage, its Project Window will be displayed.

Import XML Project... Imports an XML project file. LiveStage Professional has the ability to import/export its project files as XML formatted documents. This capability allows LiveStage Professional to work with external applications. For example, a LiveStage Project can be exported as an XML project file. This XML project can then serve as a project template and can be modified by an external application. The resultant XML project file can then



be imported into LiveStage Professional and the customized movie can be generated. Make sure the media that is used is in the proper locations since the data in the XML file contains just references to the media.

Open Movie...

Displays a dialog allowing you to select a QuickTime movie to open. This allows you to use the debugging console with movies you have already created.

Open Movie URL...

Displays a dialog allowing you to enter a URL to select a movie to open. This allows you to open QuickTime movies from a Web Site and to use the debugging console with them.

Close (Command W)

Closes the current window. If the Project Window is being closed and changes were made to the project, a prompt will appear asking if the project should be saved.

Save (Command S)

Saves changes to the current project. This will write over the previously saved version. When a project is saved for the first time, LiveStage produces a Save As dialog box, where a name and location can be assigned to the project file.

A Library folder will be created in the same folder as the newly saved project if one does not already exist. This is where files used by the project should be placed.

Files should be saved often to preserve the most recent work in case of power loss or system crash. There is no auto save feature.

Save As...

Saves the current project with a prompt for another name. This new project file becomes the current project file you are working on.

Revert

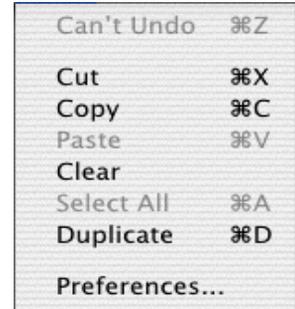
Reverts the current project to the last saved version. This is useful if some undesirable changes have been made to the current version of the project.

Quit (Command Q)

Quits the application (**Under OS X, this menu item is in the LiveStage Pro menu**). If any open projects need to be saved, the program will prompt the user and allow changes to be saved.

Edit Menu

The Edit menu commands vary depending on which LiveStage Window is currently active. If a command can be undone, the Undo and Redo menu items will be available. If the command cannot be undone, the options will be grayed out. The Edit menu also provides standard editing commands such as Cut, Copy, Paste, Clear, Select All and Duplicate which can be applied to the script editing pane as well as objects in the Stage Window, in the Tracks Tab of the Project Window and Samples sprites Tab within LiveStage Professional. The Edit Menu also allows the user to change the Preferences for LiveStage Professional.



Undo (Command Z)

Undoes the previous action or command. It is not available for all items.

Redo (Shift Command Z)

Repeat the previous undone action or command. It is not available for all items.

Cut (Command X)

Cut an object or some text from the Script edit field to the clipboard.

Copy (Command C)

Copies an object or text from the Script edit field to the clipboard.

Paste (Command V)

Paste an object, or text in the Script edit field from the clipboard.

Clear (Del)

Removes the selected object or text.

Select All (Command A)

Select all of the objects, or all of the text in the QScript pane.

Duplicate (Command D)

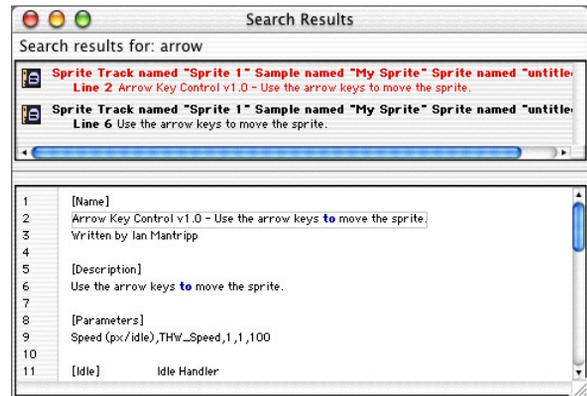
Duplicate or make a copy of the selected object.

Preferences...

To modify general preferences for LiveStage, go to the Edit menu and select Preferences under OS 9.xx (**Under OS X, this menu item is in the LiveStage**

Click on “Find” to just do a find and click on “Replace All” to replace all occurrences of the string. “Replace All” is NOT undoable so use with caution.

When doing a global or batch search, all search results will be presented in the search results window.



The upper part of the window presents each location of the search string. The Track, Sample and Sprite is shown as well as the line number and some text from the script. Clicking on one of the search results will show the script containing the found string and will select the line of text that contains that string. You can then edit the script in this window. You can also double click on the search result to open the script in its own editor window. This will open the sample editor or sprite editor as needed.

Enter Selection (Command E) This command will enter the currently selected text as the search string. You can use this along with “Find Again” (Command G) to quickly find the next occurrence of the string.

Find Again (Command G) This will search from the current insertion point for the next occurrence of the search string.

Find Selection (Command H) This command will enter the currently selected text as the search string and will search from the current insertion point for the next occurrence of the search string.

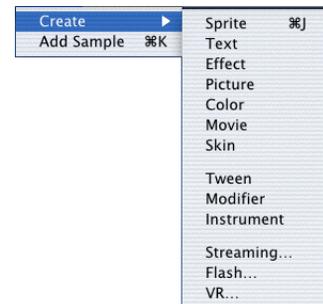
Replace (Command =) This will replace the selection with the replacement string.

Replace & Find Next (Command L) This will replace the selection with the replacement string and will then search again for the search string.

Replace All This will replace all occurrences of the search string with the replacement string. “Replace All” is NOT undoable so use with caution.

Track Menu

The Track Menu provides commands to create new tracks in the project as well as add media samples to the selected track. This menu works in conjunction with the Tracks tab in the Project Window and is only enabled when that tab is active.

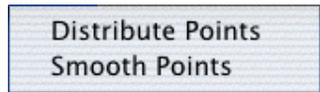


Create Select the track type to create from this menu. The types of tracks that can be created via the menu are: Sprite, Text, Effect, Picture, Color, Movie, Skin, Tween, Modifier, Instrument, Streaming Flash and VR.

Add Sample (Cmd K) Creates a new sample in the selected track. The new sample will be placed after the last sample in the track.

Path Menu

The Path menu is active when you are editing Path Style Tweens and have more than one path point selected. For more details on Tweens refer to “Tween Tracks” on page 195.



Distribute Points

Distributes the selected points evenly within the selected portion of the path.

Smooth Points

Smooth Points will reduce angles at the points of a selected group of points.

Time Menu

The Time Menu provides a method to move to any point in time within the movie.



Go to Start

Goes to the start time of the project, or the time of 00:00.000 (minutes/seconds/fractions of a second).

Go to End

Goes to the end time of the project. This is the time at the end of the last sample.

Go to Previous Frame

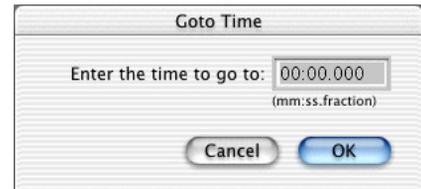
Goes to the previous frame in the current movie.

Go to Next Frame

Goes to the next frame in the current project.

Go to Time...

Brings up the “Go to Time” dialog to allow a specific time in the format of 00:00.000 (minutes/seconds/fractions of a second) to be entered from the dialog shown below. The fractions of a second component of the time is based on a range from 0 to the project’s time scale which is fixed at 600 in LiveStage. Thus a time value of 00:00.300 would represent half a second.

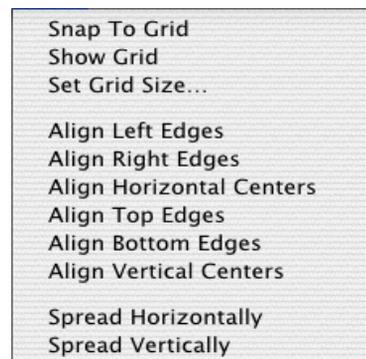


Font Menu

The Font Menu provides a method to change the Font, Style, Size and Color of the fonts used in the Text Media Sample. The menu is disabled unless you are entering text into a Text Media Sample. Note that when specifying a font and type style information for a Text Media Sample, you can use multiple settings within one media sample.

Grid Menu

The Grid Menu handles all alignment and display of objects in the Stage Window and the Tween Editor. The settings for the Grid are used in both. This menu is only active when the Stage Window or the Tween Editor is active.



Snap to Grid

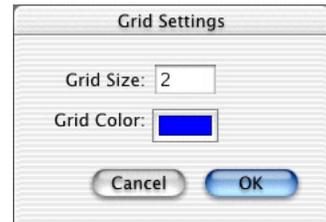
Align objects to the grid when they are moved.

Show Grid

Displays / Hides the grid.

Set Grid Size...

Brings up the Grid Settings dialog to allow setting the grid size and color. Enter in the grid size in pixels (the number of pixels between grid lines) and choose a color for the grid lines.



Align Left Edges

Aligns the left edges of the selected objects.

Align Right Edges

Aligns the right edges of the selected objects.

Align Horizontal Centers

Aligns the horizontal centers of the selected objects.

Align Top Edges

Aligns the top edges of the selected objects.

Align Bottom Edges

Aligns the bottom edges of the selected objects.

Align Vertical Centers

Aligns the vertical centers of the selected objects.

Spread Horizontally

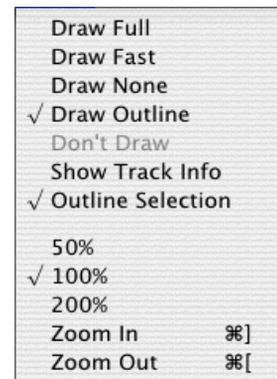
Horizontally spaces the selected objects evenly.

Spread Vertically

Vertically spaces the selected objects evenly.

View Menu

The View Menu contains the modes used to display the objects inside the Tracks Tab of the Project Window as well as the Stage Window. It also sets the zoom settings which are used for the Stage. This menu is only active when the Stage Window or the Project Window is active. Some of the menu items are only available for the Stage and will be marked as such in the following detailed description for that menu item. The menu selection will vary depending on whether the Stage Window or the Project Window has focus.



Draw Full (Stage Only)

Draws the objects as accurately as they would appear in a QuickTime movie. Draw Full includes drawing the objects utilizing any specified drawing

modes, transparency settings, anti aliasing, etc. When this is set, the Draw Fast setting is cleared. This mode should only be used on faster machines.

Draw Fast (Stage Only) Draws the objects as fast as possible, forfeiting accuracy for speed. When you are using this setting the Stage does not honor any drawing modes, transparency settings or image bit depths. When this is set, the Draw Full setting is cleared. This is the recommended setting for most machines.

Draw None (Stage Only) This is the quickest draw setting since the only thing that is drawn is a blank image. To make this useful you should turn on the Draw Outline mode which will draw the frames of all the objects in the stage.

Draw Outline (Stage Only) Toggles between drawing the outline for everything or not. If this is set, an outline is drawn around each object on the stage or the object in the Tween Editor. This can be used to see the full size of an object that may be rendered transparently. It is also useful when using the Draw None option.

Don't Draw (Stage Only) This will disable drawing within the preview area within the Tween Editor only.

Show Track Info (Stage Only) This will show the name of the track and the sprite in the stage.

Outline Selection (Stage Only) This will outline the selected object with a thin blue line. When un-checked a very faint outline is drawn. This allows for easier alignment of the edges of Tracks and Sprites.

50%, 100%, 200% Sets a zoom level for the Stage.

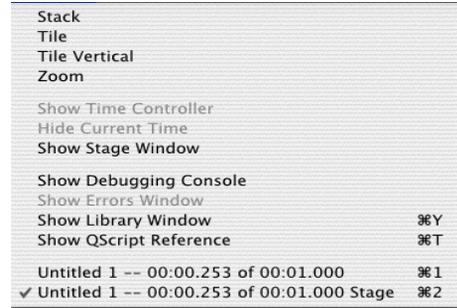
Zoom In / Zoom Out Zoom in or out the Stage or Project Window to the next available level.

Note: Keyboard Shortcut: Command] will Zoom In and Command [will Zoom Out.

settings can be incrementally set via the Zoom In (Command `]`) or Zoom Out (Command `[`) menu items.

Window Menu

The Stage and palettes within LiveStage can be shown and hidden from within the Window menu. From this menu, there are four layout options available; Stack, Tile, Tile Vertical and Zoom.



Stack

Stacks all open windows from the top left corner of the screen towards the lower right of the screen.

Tile

Arranges the open windows so that they are all visible and accessible horizontally on the screen.

Tile Vertical

Arranges the open windows so that they are all visible and accessible vertically on the screen.

Zoom

Enlarges the specified window to the size of the screen area.

The bottom menu items display the currently open windows by name. These can be chosen by selecting them in the menu or using their corresponding shortcut commands. These Window items show or hide their corresponding window, palette or display.

Time Controller
Current Time
Stage Window
Debugging Console
Errors Window
Library Window
QScript Reference

Sprite to Behavior

Will take a sprite's scripts and save them as a Behavior file. You can then edit the resultant file to customize the behavior.

Help Menu

Visit our Web Site for online support, registration, up to date QScript References and any other information.

Visit the Totally Hip Web Site This will launch the default Web Browser and go to Totally Hip's Web Site.

Get LiveStage Online Support This will launch the default Web Browser and go to the LiveStage Support page.

Register LiveStage Online This will launch the default Web Browser and go to the LiveStage registration page. It is important to register LiveStage in order to receive update/upgrade information and other information from Totally Hip.

View QScript Reference Online This will launch the default Web Browser and go to the QScript Reference online area of the Totally Hip Web Site. The most up-to-date QScript information is here.

Preferences

This section details the items found in the Preferences dialog which can be opened by selecting "Preferences" in the Edit Menu. The preferences are split into four categories: Misc, Media, Compiler and Editor. Each tab contains preferences for specific areas within LiveStage Professional. The "Defaults" button at the bottom of the Window resets all the items in the Preferences back to its original settings.

Misc Tab

The Misc Tab controls the general usability settings within LiveStage Professional.

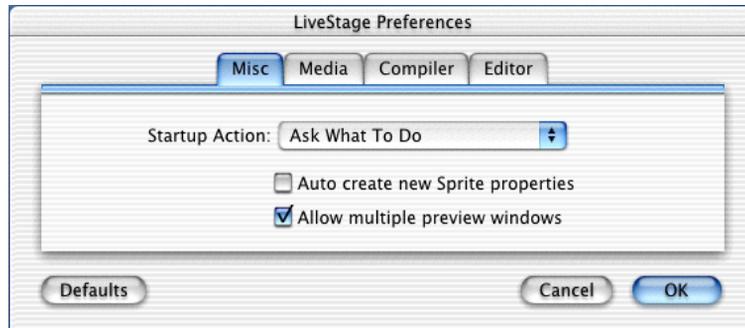


Fig 4-1: Misc Tab in Preference Window.

Startup Action

This allows you to customize the way LiveStage starts up. The options are:

- | | |
|-------------------------------|---|
| Ask What To Do | This setting will ask if you would like to open an existing project or create a new project. This is the default setting. |
| Create New Document | This setting will result in LiveStage automatically opening a new untitled Project Window. |
| Open Existing Document | This setting will open a standard file open dialog allowing you to choose which project to open. |
| Reopen Last Document | This startup action will automatically open the last saved LiveStage document. |
| Do Nothing | This setting will tell LiveStage to run but it will not open or create any document. |

Auto create new Sprite properties

The “Auto create new Sprite properties” setting controls what will happen when you move a Sprite in the Stage window. When checked, a new Sprite property chip will be created (if needed) at the current time whenever you move a Sprite

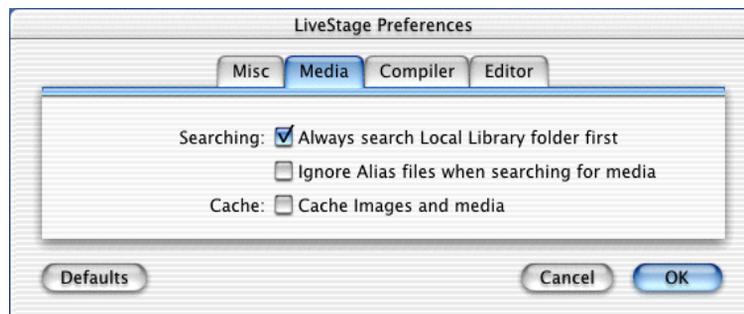
in the Stage window (See “Sprite Tracks” on page 95 for more details). When un-checked, the currently active property chip for the Sprite will be updated with the new position of the Sprite.

Allow multiple preview windows

The “Allow multiple preview windows” setting allows the user to have multiple Wired Movie Preview windows running. When un-checked, you can only have one preview window up at a time. When you run your wired movie, any existing preview window will first be closed.

Media Tab

The Media Tab controls how the media is accessed or found.



Searching

Always search Local Library folder first tells LiveStage to look for all media elements in the Local Library Folder before looking elsewhere.

Media files that are used within a LiveStage project are stored as references and thus may be located virtually anywhere. This includes media being stored on network servers, removable media, etc. This way all the media may be copied into a Local Folder and references can be set to the media within the Local Folder. This will make it easier for you to place a project on another computer

Ignore Alias files when searching for media will cause an alias to a media file or folder to be ignored. This can be helpful if you have a project copied from another computer or aliases to a network disk that is not available.

Cache

Cache Images and Media settings tells LiveStage Professional to utilize available memory to store images that are being used in memory. This setting will make LiveStage operate faster but will require more memory.

Compiler Tab

The Compiler Tab controls the compiler and debugger settings.



Compiling

Check “Play sound on success” or “Play sound on failure” to play distinctive sounds to indicate whether or not a script has compiled successfully. The default has both options checked.

Running

Default to Show source in debugger if checked, will tell LiveStage to display all executed QScript commands in the Debug Console Window by default when you are running a movie. The display of script source can also be controlled via the use of the “#debug on/off” preprocessor directive.

Clear Console before running will clear the Debug Console before running the movie.

Open exported movie in Movie Player will open a movie in the QuickTime Movie Player after it has been exported to a wired movie.

Editor Tab

The Editor Tab gives settings that affect all QScript Editor fields.

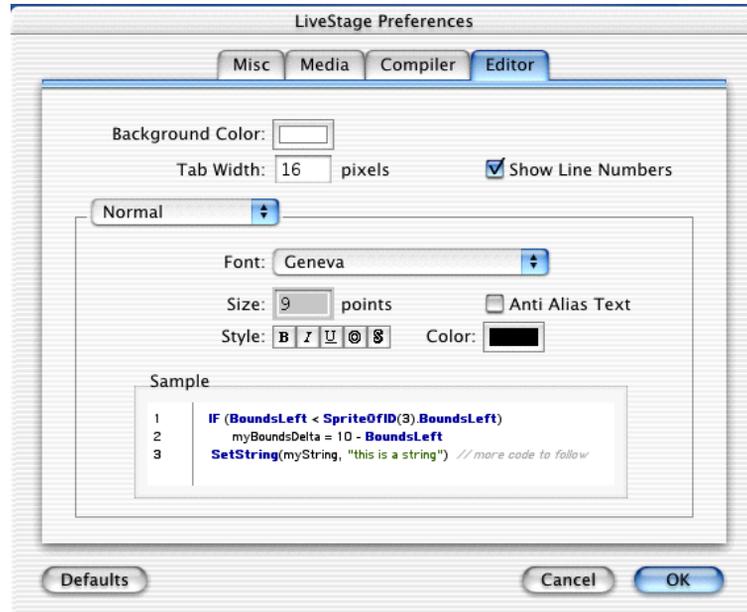


Fig 4-2: Editor Tab in Preference Window.

Background Color a color chip indicating the background color for all QScript edit panes. Click on the color chip to change the background color.

Tab Width indicates the size of a tab within a QScript edit field.

Show Line Numbers this indicates whether to display the line numbers along the left margin of the Script Editor.

Syntax Style

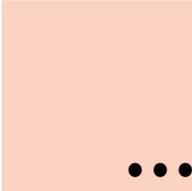
This indicates the formatting styles being used in the QScript editor's syntax highlighting. From the popup menu select the Syntax Style to be modified: Normal, Reserved, Constants, Comments, Strings or Highlighted. After selecting the syntax style, customize it by changing the Font, Size, Style and Color items in the Syntax Style group. A preview of the new Syntax Style will appear in the Sample section.

Here is a description of each syntax style and when it is applied:

Normal	This is the default text setting for any text which is not covered by the other settings.
Reserved	This indicates the text setting for any keywords that are used within the QScript language.
Constants	This indicates the text setting for any constants that are used within the QScript.
Comments	This indicates the text settings used for all comments in your QScripts.
Strings	This indicates the text setting for any strings in your QScripts.
Highlighted	This indicates the text setting used on a QScript line when it is highlighted.

Following is an outline of each syntax style setting:

Font	The font menu lists all available fonts that can be used in the syntax highlighting.
Size	Indicates the point size to use.
Anti Alias Text	When checked all text in the Script Editor is anti aliased.
Style	The type style to be used on the font. You may use any combination of the styles provided, including bold, italic, underline, outline and shadow.
Color	Click on the color chip and a color picker will appear. Select the desired text color from the palette. By default the color is set to black.



Chapter 5 Working in the Tracks Tab

Overview	69	Editing Tracks and Media Samples	78
Creating Tracks	69	Track Order	79
Selecting Tracks and Media Samples	72	The Playhead	79
Common Track Properties	72	Selection Properties View	80
Common Visual Track Properties	75	Media Samples	80

Overview

This chapter provides details on working with the Tracks Tab, the primary tab within the Project Window. We will go into how you work with the Tracks Tab to create, delete and modify properties of a track. In later chapters we will outline the different track types and how they are used.

To view the Tracks Tab, click on the Tracks Tab in the Project Window.

The Tracks Tab provides a timeline based view of your project. Most operations that you will want to perform on content within your document will be done through this tab.

LiveStage presents the contents of your project in the form of tracks. A track contains media samples, which are the containers of the media data in a track.

Your LiveStage movie has no fixed duration, thus you can place media samples at any point along the timeline. Your movie's duration is determined by when the last sample in the project finishes. Your movie also has no fixed dimension; its dimensions are just big enough to allow all visible tracks to be seen.

Creating Tracks

When you create a new project, the Tracks Tab is empty, so the first thing you need to do is create a track in which media can be placed. Opening the Tracks Menu will present you with a Create option. Selecting Create presents you with a list of different track types that you can create in LiveStage.

These are:

Flash Track

A scriptable visual track in which a Flash Movie (swf) file created in Macromedia Flash is stored. This track allows you to script Flash movies and integrate them into your own project. For more information please refer to chapter 7 on page 89.

Sprite Track	A scriptable visual track in which sprite samples are stored. Sprites are graphics that may overlay other visual tracks and may have interactivity and animation. For more information please refer to chapter 8 on page 95.
Text Track	A scriptable visual track in which text samples are stored. Text Tracks contain text samples that may be displayed within the movie or used as a Chapter or HREF (text that references a Web Page) track. For more information please refer to chapter 9 on page 115.
VR Track	A scriptable visual track for storing interactive 3D movies. For more information please refer to chapter 10 on page 129.
Instrument Track	An audio track in which instrument samples are stored. Each sample stores information on different instruments you may use in scripts within your sprite track. For more information please refer to chapter 12 on page 143.
Effect Track	A visual track in which effect samples are stored. Effect tracks are used to render transitions or filters. For more information please refer to chapter 13 on page 153
Picture Track	A visual track in which picture samples are stored. Picture tracks contain a series of images that can be displayed over time as in a slide show. For more information please refer to chapter 14 on page 161.
Color Track	A visual track in which color samples are stored. Color Tracks contain a vector graphic image that can have gradient colors. For more information please refer to chapter 15 on page 167.
Movie Track	A visual track in which movie samples are stored. Movie Tracks contain other QuickTime movies that can be independently controlled. For more information please refer to chapter 16 on page 173.

Skin Track	A visual track in which a skin sample is stored to control the shape and size of the frame surrounding your movie to create a unique environment. For more information please refer to chapter 17 on page 181.
Streaming Track	A visual track which links to a streaming movie stored on a server. For more information please refer to chapter 18 on page 187.
Tween Track	A non-visual track in which tween samples are stored. Tweens perform interpolations between values of various data types that may then be used as a source of data in another track. For more information please refer to chapter 20 on page 195.
Modifier Track	A non-visual track that produces a predefined sequence of values over time, as specified by the user. Modifier Tracks are used as a source of data for properties in other tracks or sprites. For more information please refer to chapter 21 on page 213.
External Tracks	A visual or audio track which can be incorporated into a LiveStage project but cannot be edited. Examples of External Tracks are Video Track and Audio Track. For more information please refer to “External Tracks” on page 141.

The last three tracks mentioned above are supported by LiveStage but may not be created in LiveStage. These are known as external movie files. Streaming Track, Flash Track and VR Track are augmentable tracks. Video and Sound tracks are created by dragging and dropping a QuickTime movie into the Tracks Tab of the Project Window. The Video and Sound Tracks are not editable.

After creating a new track you will see that track appear in the track list and it will contain one default media sample. The default sample is used as a starting point for you to add content to that track. By default it contains no data initially.

Selecting Tracks and Media Samples

To select a track, click on the Track Header (see “Track Header” on page 26) in the Tracks Tab. To select a Media Sample (see “Media Sample” on page 26) click on the sample in the Tracks Tab. Note that the last item you clicked will be the currently selected item.

When you click on a Media Sample, both the track and the media sample are selected. The Timeline will display a horizontal bar with the same start time and duration as the selected Media Sample. The Information Panel at the top left corner of the Tracks Tab will show either track information or sample information based on the current selection. If a track is selected, the Track dimensions and its duration will be shown; if a sample has been selected, the Start Time and Duration will be shown.

Common Track Properties

Track properties describe various details of a track, such as the track’s name, how it is used in the movie and preload information.

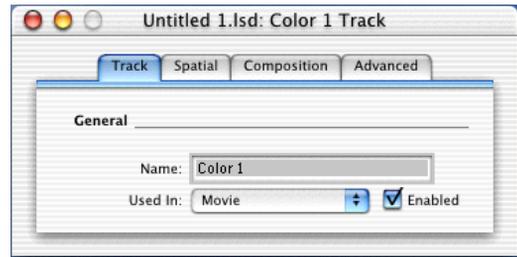
In order to view and/or modify track properties, double click on the track’s name in the Tracks Tab. Any track properties window already open for the track will be brought to the front.

Each track type will display a slightly different version of this window since the various track types have different properties, but all of the tracks will contain the Track and Advanced Tab. Those two tabs contain all of the common track properties. Visual tracks will display the Spatial and Composition Tabs. The Instrument Track and other audible tracks will have an Audio Tab.

Following is a summary of the common properties and what they mean. Details of those properties unique to different track types will be discussed in the chapters on those tracks.

Track Tab

The Track Tab is common for all the tracks and it contains the name and information on how the track is used. Some Tracks may contain additional information in this tab. For more information please refer to the chapter that describes the specific track type.



Name

The name of the track. This is displayed in the track list and may be used to refer to the track in QScript. An appropriate default name is given to each track as it is created. You can change this to a name that is more meaningful.

Used In Movie

Indicates how the track is to be used in the movie. The options here indicate if the track is to be used in the preview, the poster, the movie or some combination of these.

The preview is a very short version of the movie, usually less than 5 seconds in duration, which gives the user an idea of the movie's contents.

A movie's poster is a still frame, which represents the contents of the movie.

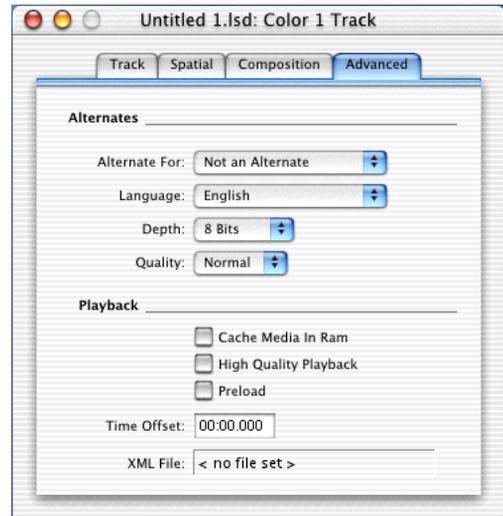
The movie setting indicates that the track will be available while the movie is playing. This is the usual setting.

Enabled

This check box indicates that the track will be enabled when the movie is initially launched. If you are enabling this track at some future point using a QScript or you wish to have the track initially disabled then you will want to uncheck this field. By default a track is enabled.

Advanced Tab

The Advanced Tab contains “Alternates” and “Playback” related information for the track. The information presented in the tab is common to all tracks types.



Alternates

Alternate For

Specifies whether this track is used as an alternate track for the track selected in the popup menu. Alternates are used to support different language types or different levels of quality from an audible or visual track.

Language

Choose what language this track is in. When an alternate track is chosen, the one matching the user’s language will be enabled.

Depth

Choose the bit depth of this track. The track that most closely matches the bit depth of the user’s monitor will be enabled.

Quality

Select the quality of this alternate. A track of the highest quality that matches the other alternate criteria will be chosen.

Playback

Cache Media In Ram

This check box will provide QuickTime with a hint that it should try to cache the media for this track. This can improve performance when the movie plays, but will require more memory for playback.

High Quality Playback

This check box indicates that the track should be displayed at the highest possible quality, without regard to real time performance considerations. Enabling this option may cause your movie to play slower on lower performance computers.

Preload

This check box indicates whether to preload the track. When you preload a track, that track is loaded into memory. This provides smoother playback when that part of the track is played. In most cases you would leave the preload unchecked because it eats up RAM.

Time Offset

A time value that indicates an offset for all of the samples within the track. By setting this value you effectively offset all samples in the track by a fixed value. This is useful for external tracks where you can not move or edit the track samples.

XML file

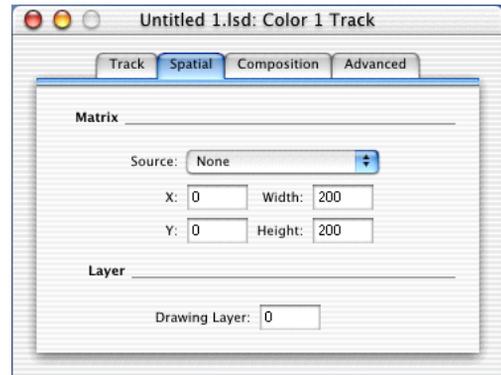
The XML file attached will be included in your track. The file is used to initialize a track's QTList. The XML file can be dragged from the Library or the Finder and dropped into this location. To remove an attached file, select it and then press the Delete key.

Common Visual Track Properties

Visual Tracks have an extended set of common properties, which are located in the Spatial and Composition Tabs. Some of the track types for which these apply to are: Sprite, Color, Text, Movie, Effects and Picture tracks.

Spatial Tab

The Spatial Tab contains information for the visual track's Matrix and Layer properties. It is highly recommended that you control how you want the tracks to be drawn by setting the Drawing Layer in the Layer properties for all visual tracks.



Matrix

Source

This popup menu lists all of the Tween and Modifier tracks that can be used to automatically set the drawing matrix for this track. If you select “none” then you can set the matrix of the track yourself using the properties that follow. Tweens and modifiers can rotate and distort the visual image of the track.

X, Y, Width, Height

These values specify the visual location and size of the track. Specifying the location only makes sense if there is more than one track in your project.

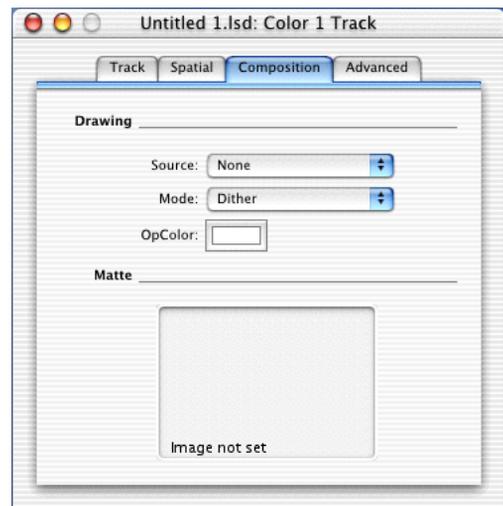
Layer

Drawing Layer

The drawing layer indicates the visual stacking order (z-order) that the tracks will be drawn in. Lower numbers are closer to the front. Values can range between -99999 and $999,999$.

Composition Tab

The Composition Tab contains information about the Drawing and Matte effects used on visual tracks.



Drawing

Source

This popup menu lists all Tween and Modifier tracks that can be used to set the drawing mode for this track. Select the one you want to use or “none” if you want to set the properties manually.

Mode

This popup menu shows all of the standard drawing modes. Select the mode you want from this list. The drawing modes are described in detail in Appendix II.

OpColor

This color control allows you to select the color to be used by the drawing mode. Not all drawing modes make use of this color.

Matte

A gray scale image can be dragged into this Matte field from the Library or the Finder to be used as a mask for the track. The image will use the alpha channel to mask out and display only areas where the Matte image overlaps the original image. A Matte image can be 8 bits deep and will specify various degrees of transparency. Currently with QuickTime 5.0.1 you can only use the Matte image in a video. All other tracks must use a 1 bit mask.

Double clicking in the Matte field will display that image at full size in its own window.

Editing Tracks and Media Samples

There are a few simple editing operations you can perform on Tracks and Media Samples that can make the creation of your projects quicker and easier.

Duplicating

If you want to duplicate a Track or Media Sample in LiveStage, use the Duplicate menu item located in the Edit menu. Duplicate will create a copy of the currently selected Track or Media Sample and place it in the Tracks Tab. In the case of duplicating a Track, the new Track will appear at the end of the Tracks list. Duplicated Media Samples will appear in the first available location in the track. The duplicated Media Sample will have the same properties as the original.

Duplicate performs the same operation as doing a copy and paste operation.

Cut / Copy / Paste

The Cut, Copy and Paste operations work in much the same way in LiveStage as they do in other applications. Within the context of the Tracks Tab both tracks and samples can be operated on in this manner.

Use Cut on a track or sample to remove the selected item from the tracks tab and store it in the clipboard for later use. Copy operates in much the same way except it stores a copy into the clipboard and the original is not removed.

Pasting a track into the Tracks Tab will add that track to the end of the track list. The new track will have all the same properties as the original. To paste a sample into a track you must first select a track that can contain the type of media that sample contains. For example, you can copy a sprite sample and paste it into any sprite track but you cannot paste that sample into an instrument track.

Removing Tracks and Media Samples

A selected media sample or track can be removed by using the Clear item in the Edit menu or by pressing the delete key.

Track Order

When you are viewing tracks in the Tracks Tab you may have noticed a series of numbers that are displayed beside the track's name in the track header. These numbers represent the index of that track. Generally this information remains unused, however if you are accessing one or more tracks through QScript, you can use commands that refer to the track index.

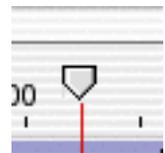
The track index can be used to identify that track, but note that the track indices may change. You can change the ordering (and thus the indexing) of your tracks by dragging them to a new position in the list.

If you do rearrange the tracks be sure to update any references to the track's index number in all QScripts that you have created.

As you drag a track around in the track list, a highlight bar will be displayed indicating where that track will be placed when you release the mouse button. Rearranging the tracks only affects their visual representation if more than one track has the same layer number. When more than one track has the same layer number, then those tracks are drawn in the order they appear in the tracks tab with higher numbered tracks appearing in front of lower numbered tracks.

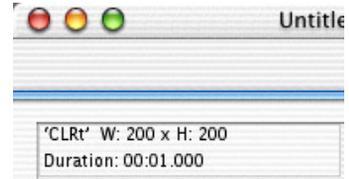
The Playhead

The playhead indicates the time in your project that is displayed in the Stage Window. By dragging the playhead, the view in the Stage Window will be updated to represent the visual elements in place at that point in time. This also sets the current time in the document which can affect property editing in Sprites.



Selection Properties View

The Selection Properties View provides you with a quick reference area for track and media sample properties. This view displays information on the item that you currently have selected.



In the case of tracks, the Selection Properties View will show you the track type and its duration. For visual tracks the width and height of the track will also be displayed. When you have a media sample selected the view will show you the starting time and duration of that sample.

Media Samples

Media samples represent a block of media data within a track. A track may contain one or more samples and these in turn will have different starting points and durations within the track. The data that a sample contains will depend on the type of track in which it is contained. For example, a sample in an instrument track will contain a list of instruments and a sample in a picture track will contain a single picture.

Adding

LiveStage will create an initial sample for you when you create a new track. To create an additional sample, select the track you wish to add new samples to and then choose "Add Sample" from the Tracks menu for each new sample you want to add. This will place a new empty and untitled sample in your track at the end of the last sample.

Note that External Tracks, such as Flash and VR, allow very little editing to their contents. One of their limitations is the inability to add new media samples to the track.

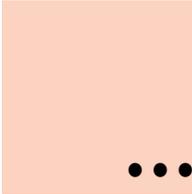
Moving

In some cases you may wish to move samples around, either to change where they start and end or to re-sequence a group of samples in a track. The simple way to move samples is to click and drag the sample.

To make aligning samples easier you can hold down the Option key while dragging a sample. This will snap the sample to the edge of any other samples in other tracks.

Changing the start time and duration properties is an alternative method to move samples. You can also offset the entire track by altering the track offset value in the track properties or dragging the bar at the left edge of the track sample area.





Chapter 6 Scriptable Tracks



Overview	85
Event Handlers	85
Script Edit Field	87
Script Management	88



Overview

Several track types within LiveStage Professional are scriptable. This means that these tracks (Flash, Sprite, Text and QTVR tracks) are capable of responding to events triggered by the user (mouse and keyboard events) and by the computer system (idle and frame loaded events). Scriptable Tracks act as controllers within your project. This means that they can monitor and respond to events and send out commands (QScript Actions and Properties) to control itself and other tracks within the QuickTime movie.

The Scriptable Tracks in LiveStage Professional are the Flash, Sprite, Text and QTVR Tracks. The following four chapters provide the details for each of these scriptable tracks.

Event Handlers

LiveStage Professional are “event driven”. This means that as things happen (i.e. the user click the mouse, time passing, etc.), events are “triggered”. A Scriptable Track is capable of responding to these events. The way a Scriptable Track responds to events is via Event Handlers.

An Event Handler is a script written to respond to a specific event. As events are triggered, their corresponding Event Handler in the Scriptable Track (if it exists) is executed. This allows the Scriptable Track to take control at the time the event is triggered. For example, a sprite in a Sprite Track can respond to mouse clicks. The Mouse Click Event is triggered when the mouse button is clicked on a sprite. If a script is available in the sprite (receiving the mouse click), it is executed and thus can take control of the movie at that time. This allows you to perform simple operations like changing the image of a sprite (animation), or more complex operations like moving the sprite around in its track (i.e. drag and drop) or start the movie playing.

Each Scriptable Track type in LiveStage Professional uses a different editor interface due to the different types of media each track type handles. However, each of these editors have a similar scripting interface. Each Scriptable Track’s Script Editor provide an Event Handler list where you can select an Event

Handler to script. To enter a script for an Event Handler, select the event in the Event Handler list then click in the Script Edit field and enter the script to handle this event.

There are many Standard Event that a Scriptable Track is capable of handling. Below we list these Event and describe what triggers them. These events can be divided into two groups. These are the Track Events and the Object Events.

Track Events are triggered by events that are not related to the objects contained with the Scriptable Track. For instance, in a Text Track, the Frame Loaded event is a Track Event as it is not related to a Text Hotspots but to the Text Track as a whole.

Object Events, on the other hand is directed specifically at a Scriptable Track Object (i.e. Sprites, Text HotSpots, Flash Buttons, and QTVR Hotspots).

Frame Loaded The Frame Loaded Event is a Track Event and is triggered when the sample is loaded from the movie. There is only one Frame Loaded Event for each Track Sample. When you write scripts for this event, be sure to refer to the Scriptable Track using the track targeting syntax (described in Introduction to QScript).

List Received The List Received Event is a Track Event and is triggered by the receipt of a QTList in response to a ExchangeList or a QueryListServer QScript Actions. This event is only available for Flash, Sprite and Text tracks.

Idle The Idle Event is triggered whenever the movie gets idle time. The frequency of this event is affected by the value you specify in the Scriptable Track's Idle Frequency setting. This event is only available for Sprite, Text and QTVR tracks.

Key Pressed The Key Pressed Event is a Track Event and is triggered when the key is pressed. This event is only available for Flash, Spite and Text tracks.

Mouse Moved The Mouse Moved Event is a Track Event and is triggered when the user moves the mouse within a Scriptable Track. This event is only available for Sprite and Text tracks.

Mouse Click	The Mouse Click Event is an Object Event and is triggered when the user presses the mouse button down and releases the mouse button while over an object in the Scriptable Track.
Mouse Enter	The Mouse Enter Event is an Object Event and is triggered when the user moves the mouse over an object in the Scriptable Track.
Mouse Exit	The Mouse Exit Event is an Object Event and is triggered when the user moves the mouse from inside an object in a Scriptable Track to the outside.
Mouse Down	The Mouse Down Event is an Object Event and is triggered when the user presses the mouse button down while over an object in the Scriptable Track.
Mouse Up	The Mouse Up Event is an Object Event and is triggered when the user releases the mouse button after it was first pressed in an object in the Scriptable Track.

***Note:** Flash buttons within Flash tracks trigger a different set of events. This is due to the differences between the Flash and QuickTime scripting metaphors. Only the Frame Loaded and List Received events listed above are supported for Flash tracks. Please see the chapter on Flash Tracks for details on the Flash Button events supported by LiveStage Professional.*

Script Edit Field

The Script Edit field becomes active when you have selected an Event Handler from the Event Handler List.

If the Event Handler contains a script, it will be displayed in the Script Edit field. In cases where there is no script attached to the selected Event Handler, the Script Edit field will remain empty. Once you have an Event Handler selected you may enter QScript commands into the Script Edit field. You may notice that as QScript commands and keywords that are entered into the Script Edit field will become colored or emphasized in some way. This is due to the syntax highlighting. For more information on the syntax highlighting refer to Preferences “Editor Tab” on page 65.

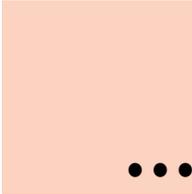
Script Management

There are a couple of script management features you may want to be aware of while working with Scriptable Tracks. Above the top of the Script Editor field you will see three buttons labeled “Save...”, “Load...” and “Check Syntax”. These buttons allow you to store and retrieve scripts that you may use on a regular basis.

If you have a script that you would like to save for use at some later point, you can click on the “Save...” button. This will prompt you with a file save dialog, you may then specify where to store the script file.

To retrieve a script that you have previously saved or otherwise acquired, use the “Load...” button and use the file requestor to locate and open that file. The new script will appear in the current script editor window.

Clicking on the “Check Syntax” button will perform verification of the script in the script editor to ensure that it is syntactically valid.



Chapter 7 Flash Tracks



Overview	91	Flash Sample	92
Creating Flash Tracks	91	Interoperation with Stage	94
Flash Track Properties	92	Scripting Flash Buttons	94



Overview

LiveStage allows you to import a movie file created in Macromedia Flash and placed into a single track called a Flash Track. Flash Tracks are External movies since they are not created within LiveStage Professional but can be utilized by LiveStage to create a movie.

Creating Flash Tracks

You can create a Flash Track by selecting “Flash Track” from “Create” submenu within the “Tracks” menu. An Open dialog will appear and you can then select a SWF file or a QuickTime movie contain a SWF file or a Flash Track and import it into your project.

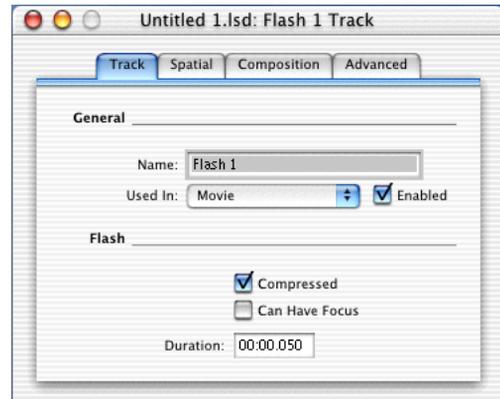
Alternatively, you can drag and drop a SWF file or a QuickTime movie containing a Flash Track from any folder into the Tracks Tab. This will also create a Flash Track in your project. If you dragged a QuickTime movie containing tracks other than the Flash Track you want, you can then delete any of the extra tracks.

Both the Track Properties and the Track Sample for the Flash Track can be edited. However no new samples can be added to the Flash Track. Flash Tracks are augmentable tracks since their content cannot be edited by LiveStage. You can however augment Flash Tracks by adding scripts to Flash button handlers.

Flash Track Properties

Double clicking on the Flash Track will bring up the Flash Track Info Window. The Flash Track properties include all of the standard properties for a visual track. For more information on these properties please refer to Chapter 5, “Working in the Tracks Tab” on page 67. The unique settings specific to the Flash Track are the “Compress” checkbox, the “Can Have Focus” checkbox and the Duration. The “Compress”

checkbox allows you to compress the flash sample. This can make the flash track even smaller but can sometimes affect playback performance. The “Can Have Focus” checkbox will enable or disable the ability of the Flash Track to receive keyboard focus. The Duration setting allows you to set the duration for the Flash Track.

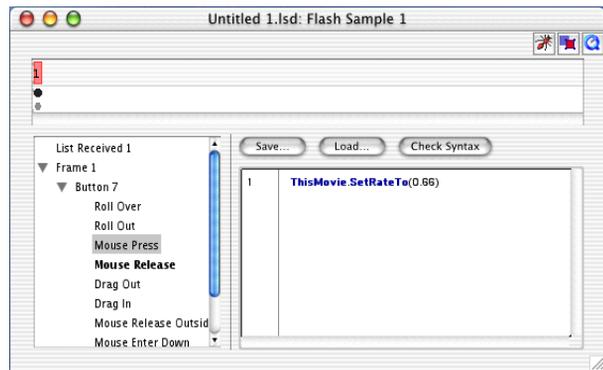


Flash Sample

The Flash Sample window provides the capability to add scripts at a specified frame or button event. This allows you to manipulate the Flash Track as well as other tracks or scripts. Scripts in the Flash Tracks perform exactly the same functions as the scripts written for the other scriptable tracks except that there is no support for the GlobalVars, LocalVars and SpriteVars keywords. You can only use the MovieVars keyword.

The sample in a Flash Track represents a group of frames from the Flash Track. The contents of the Flash Sample are broken down into frames so that you can see the scriptable Flash elements (buttons) easily.

By double clicking on the sample, you will be presented with the Flash Sample window. At the top of the window is the Flash Sample Timeline. Below the Flash Sample Timeline near the left side is the Flash Object/Event Handler List. Beside the Object/Event Handler List



is the Script Editor window. Along the top of the Script Editor Window is the Save Script, Load Script and Check Syntax buttons.

The Flash Sample Timeline displays all the frames in the Flash file. The rectangular red highlight indicates the currently selected frame. The gray dot indicates that there are buttons in that frame and the black dot indicates that there are Flash scripts in that frame. You must select a frame containing Flash buttons in order to enter your scripts by clicking on the top half of the Flash Sample Timeline.

At the top of the Object/Event Handler List is the List Received handler. The Object/Event Handler List also contains the currently selected Frame. If the frame contains buttons there will be an expansion icon beside them, allowing you to expand the view. The buttons themselves can also be expanded to display the Event Handlers for that button.

This is followed by a list of Flash Variables in the Flash Track. Flash buttons are a little different from other scriptable objects in LiveStage Professional. If you expand a button's view you will see that there are different handlers that may be scripted. Like sprite handlers, just click on the handler you want to script, then enter the script into the script edit window. The script handler and associated button and frame in the Object/Event Handler List will be bolded if there is scripted entered for that script handler.

The Flash Object/Event Handler list can be hidden by holding down the Control key and clicking on the separator bar between the list and the Script Editor. A separator bar will appear to the left of the Script Editor when the Flash Event Handler list is hidden. Clicking on that separator will bring the list back.

For details on what the different Flash button events are and how they work, please refer to the Macromedia Flash documentation.

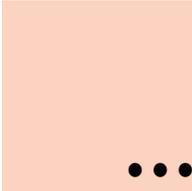
Interoperation with Stage

You can bring up the Flash Sample window by double clicking on the Flash Track in the Stage. The window will appear with the current Frame in the Flash Sample Timeline selected. When moving the cursor over the Flash Track in the Stage you will notice the cursor will change every time it passes over a button. This makes it easy for you to identify what is a button and what is just a plain graphic. By double clicking on the button, the Flash Sample window will appear and it will also open up the Event Handlers for the selected button as well as the current Frame. This makes it easier to add the script to the correct buttons since the original names for the buttons cannot be identified when imported into LiveStage.

Scripting Flash Buttons

You can enter QScript commands into the Script Editor field corresponding to a Flash frame or Button event handler. Thus allows you to control the Flash and/or other movie elements. You may script a frame by selecting the frame and then enter a script into the script edit window. This works like the Frame Loaded handler in a sprite object. If there is Flash script in that Event Handler you can override the script with your own. If you just delete the script, it will not remove it. The only way to remove the script is to replace the script with a space.

Please refer to “Introduction to QScript” on page 219 for details of QScript concepts. Also refer to the Flash Track Properties and Actions sections in the QScript Reference documentation for details of the Flash Track Action and Properties.



Chapter 8 Sprite Tracks

Overview	97	Sprites Tab	103
Sprite Track Description	97	Properties Tab	105
Creating a Sprite Track	98	Button Tab	108
Sprite Track Properties	98	Scripts Tab	110
Sprite Samples	100	Behaviors Tab	112
Images Tab	100		



Overview

In this chapter we will discuss Sprite Tracks and Sprites and how they can be used in your projects. Sprite Tracks are scriptable tracks. Sprite animation produced by LiveStage is substantially different than traditional video animation. In traditional video animation, a frame is described by specifying each pixel, in other words, a static image. In contrast to this, sprite animation describes frames by specifying which sprites appear, where, and when. At any point in time a sprite will display a single image that is selected from the pool of images stored in the Sprite Sample.

Sprite Track Description

Sprite media is one of QuickTime's most powerful media types. A Sprite Track is a visual track type and therefore has a placement, size and layer. In addition to these visual properties a Sprite Track has one or more Sprites which themselves can be layered, positioned, scaled, rotated, hidden, or animated.

Each Sprite in a QuickTime movie refers to an image stored in its Sprite Sample. This image is what is used to visually represent the sprite in the movie. Within a Sprite Track there may be many Sprite Samples each of which may contain many sprites and images. The image that a sprite refers to is just another property of the sprite and thus may be changed during the course of the movie.

You can consider a Sprite Track almost as a virtual world inhabited by Sprites. Within the track, Sprites are able to move and change their visual properties. A Sprite may move outside the visible area of the track, but it will not be visible.

Each Sprite's visual properties can change either as a function of the movie's time or scripting. The image associated with a sprite may be a complex map or a cartoonish space-ship. Since a sprite must be assigned an image, it is common practice that sprites that are never displayed during playback of the movie use a 1 pixel by 1 pixel GIF image to minimize resource usage.

Sprites can be scripted. A visible Sprite may respond to user input to emulate a button. An invisible Sprite may act as a monitor examining results and playback conditions of the movie.

A powerful feature of QuickTime is the capability called “Image Override”. Image Overrides allow Sprite images to come from the sprite sample itself, another track - such as a video or Flash track - or a Web Server.

Creating a Sprite Track

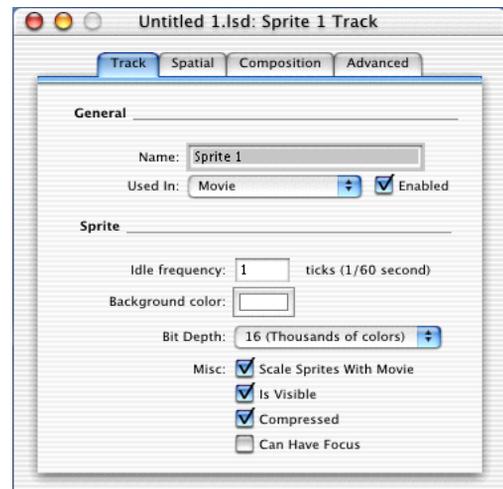
In order to start working with Sprites you will need to have at least one Sprite Track (see Chapter 5, “Working in the Tracks Tab” on page 67 to learn how to create a Sprite Track). Once you have created a Sprite Track there will be one Sprite Sample (created by default) in which you can place images and create Sprites.

Sprite Track Properties

Sprite Tracks contain the standard set of properties for a visual track (see “Common Track Properties” on page 72 and “Common Visual Track Properties” on page 75). However, there are a few additional properties contained in the sprite information window which are unique to Sprite Tracks. Double click on the Sprite Track name in the Tracks Tab of the Project Window to view and edit these properties.

In the Sprite Track property window select the Track Tab. There you will see a Sprite section that is unique to Sprite Tracks.

The Sprite section contains visual characteristics that are unique to Sprite Tracks as well as settings that affect how the Sprite Track will play back in your movie. It contains the properties that are described below:



Idle Frequency

Sprites receive idle time through the Idle Event Handler. This property indicates how frequently that idle event handler should be triggered. The value indicates how many ticks (1/60 second) to wait before calling the idle event handler again. You should set this to the highest value that allows your movie to play back smoothly in order to reduce the processor demands of the system that will eventually play your movie. It should be noted that setting the idle frequency does not guarantee that the Idle Event Handler will be called at that frequency. This value is used as a guide and thus if QuickTime is busy processing other requests then the actual idle frequency may be much less than specified. Setting the idle frequency to -1 turns off idle event calls while a value of 0 generates idle events as fast as QuickTime can.

Background Color

The color used as the background for the Sprite Track. If you are going to make your Sprite Track transparent, then choose a color that is not used in any of the sprites to ensure that portions of your sprites do not inadvertently become transparent.

Bit Depth

Indicates what bit depth the Sprite Track will be in. This can be set to either 8 (256 colors), 16 (thousands of colors), or 32 (millions of colors). All content in the Sprite Track will be displayed in the bit depth selected. You should choose the lowest bit depth that provides acceptable image quality in order to reduce the memory and processor demands of the system that will eventually present your movie.

Scale Sprites With Movie

This check box adjusts how vector images are drawn when the movie is scaled. If this setting is enabled a vector image will draw smoothly as it is scaled. If you disable this setting a vector image will have rough edges and some artifacts when it is scaled.

For example, you may use the following statement:

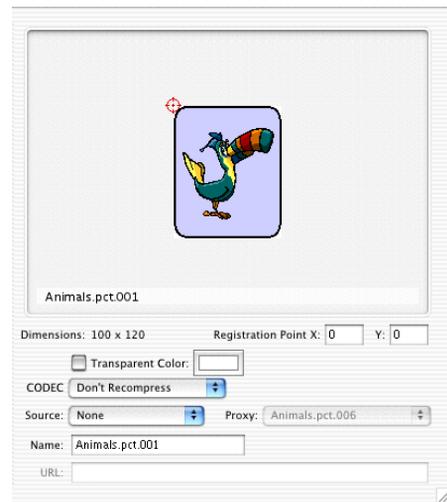
```
SpriteNamed("MySprite").SetImageIndexTo(2)
```

This statement tells the sprite named MySprite to use the second image in the Sprite Sample. You can also use the image's name in your scripts in order to refer to an image no matter how it is organized in this list. You do this by prefixing the name with \$ like this:

```
SpriteNamed("MySprite").SetImageIndexTo("$Image1.gif")
```

Image Properties

Selecting an image from the image list will display a list of properties for that image along the right hand side of the Image Tab. You will see a preview of the image, along with various properties for it. To change the graphic used for an image, drag and drop an image from the Library into the image preview area. This will replace the original image data.



Registration Point

Registration points are used to set the location in the image that is used to position that image spatially. Typically an image's registration point is located at the top left corner, defined as 0, 0. By changing the image's registration point, you create an offset for the image which will affect all drawing of that image.

As an example, if you had a sprite showing an image whose registration point was at 0, 0, and then moved it to 100, 100 the image's top left corner would be at 100, 100. If, however, you then changed the image's registration point to be 10, 10, the image's top left corner would now be located at 90, 90.

If you would like to use content from a Web Server, be it static images or streaming content, you can choose to specify the image's source using the "Get from URL" property. When your movie is played, QuickTime will retrieve the image data from the specified location. When using the "Get from URL" option you need to specify an image in the Proxy property.

Proxy

The Proxy property tells the image to use a static image stored in the Sprite Sample while waiting for the data to become available from the URL. The images listed here are the images stored in the Sprite Sample.

Name

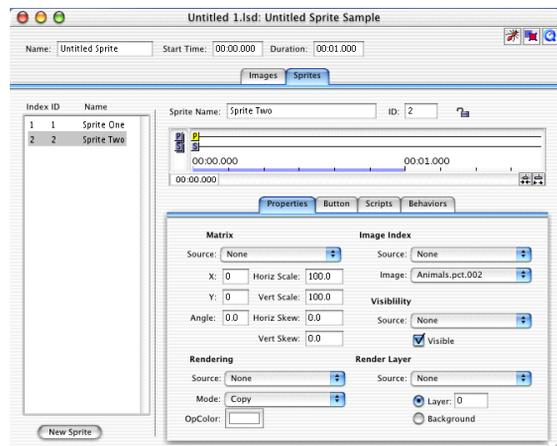
Indicates the name for the image in LiveStage. The name can be set to anything you like since it is only used for referring to the image in the property windows and scripts. The default will be the name of the original file.

URL

This is where you specify the URL to be used when the "Get from URL" setting is selected in the Source Popup Menu.

Sprites Tab

The Sprites Tab encapsulates all creation and manipulation of Sprites within a Sprite Sample. There are three sections to the Sprites Tab, the first is the Sprite List, followed by the Timeline and a series of four tabs: Properties, Button, Scripts and Behaviors that contain a variety of property settings for each sprite.



shown in the Project Window to the time for that Chip and will display the associated properties for that time. Command clicking in the Sprite timeline area will change the time shown in the Project Window to the time clicked on but will not set that as the current time for the sprite properties. These two actions allow you to see changes to other objects in the stage and to set source popups for Tween Tracks.

You can drag out new Chips from the pool of chips at the left. You can use these to change the scripts or properties of a sprite over time. These work very much like creating tab-stops in word processor packages like Microsoft Word™.

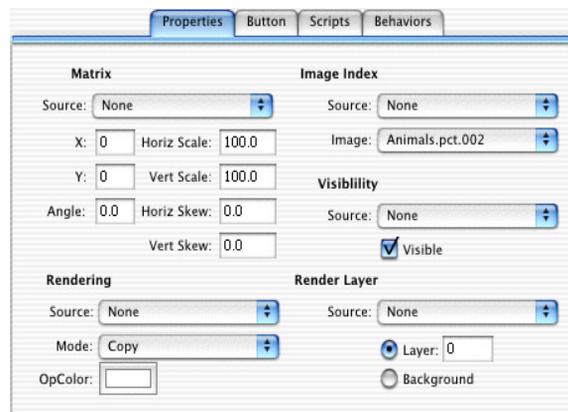
Chips can also be operated on using the Cut, Copy, Paste and Duplicate items in the Edit menu. Select a Chip, choose Copy and then Paste (or choose Duplicate). A copy of the Chip will appear just offset from the Chip originally selected. To delete a Chip, simply select it and press Delete.

Note: The first property Chip cannot be deleted.

To change the data represented by one of these Chips, click on it. This will display all of the properties for that Chip in the respective tabs in the Sprites Tab and will also change the time shown in the Project Window to the time of the Chip. If you need to view other parts of the Sprite Timeline use the scroll bar that is located across the bottom of the Sprite Timeline. Note that the time of the selected chip is displayed just below the timeline. You can also zoom in and out by clicking on the zoom icons.

Properties Tab

The Sprite Properties Tab displays the properties for the Property chip at the time selected in the Sprite Timeline. In the Sprite Properties Tab there are a number of properties that are organized into related groups. We will go through these groups one by one describing what those properties do.



The drawing mode specifies how the sprite will draw its content. The standard mode for a sprite is Copy, which just takes the source media and draws it onto the screen but will dither the source image if the destination bit depth is less than the source.

For more information on using of the Drawing Mode go to the “Appendix II - Drawing Mode References” PDF documentation.

OpColor Some of these modes will utilize a color value to modify the drawing operation. This field specifies the color to use.

Image Index

The Image Index is where you specify the image data that is to be used by the sprite.

Source specifies the Tween or Modifier Track to use to change the image index.

Tween Tracks used to control the Image Index property must be of the “Index / Layer / Flag” tween type. For more information on using Tween Tracks refer to the “Tween Tracks” chapter.

Image specifies the image to use in this Sprite Sample. Any of the images in the list can be chosen as the initial image for the sprite. The image used by a sprite can be changed under QScript control or by changing its image source in a Property Chip at any point in the sample.

Visibility

The Visibility property indicates if the sprite is visible. The visibility of the sprite can also be changed through QScript. Sprites may be hidden or shown by modifying appropriate values in their Property Chips. A hidden sprite still receives mouse-related events. It is just not visible to the user.

- Source** specifies the Tween or Modifier Track which will provide values for the Visible property over time.
- Visible** If checked the sprite will be visible at the time when the property chip is set otherwise it is hidden.

Render Layer

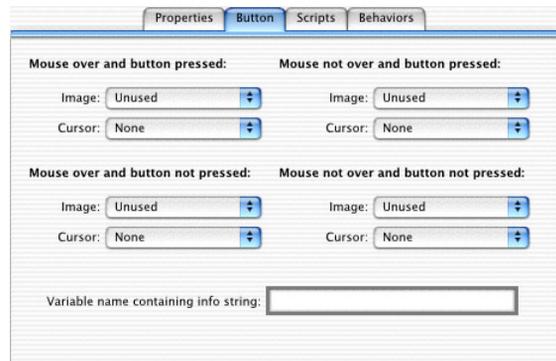
Specifies the sprites drawing order within the Sprite Track.

- Source** specifies a Tween or Modifier Track as a Source for the Layer property. This can be quite handy for moving the sprite in front of or behind other sprites within the same Sprite Track as time goes by.
- Layer** specifies the numeric value representing the sprites layer. Sprites with lower layer numbers appear in front of those sprites with higher layer numbers. The values specified for the layers may be positive or negative and range from -999 to 9999.
- Background** Clicking on the Background radio button indicates that the sprite should be placed behind all other sprites. This allows QuickTime to optimize drawing for sprites that do not change or move and are behind all other sprites.

Button Tab

The Button tab contains settings that allow you to easily make the sprite behave like a button. A button can be in one of four states, and the properties of the sprite for each of these states can be specified here.

The properties indicates the various mouse and button states for the Sprite. These include mouse over, mouse not over, button pressed and button not pressed.



Each individual grouping includes two properties that modify the behavior of the sprite for the given condition. An outline of these properties and how they affect the sprite follows.

Image

Designates which image will be displayed when the condition is met. The image is either “None” or one of the images available in the sprite’s sample. Choosing “None” means that the image will not be changed by this condition.

Cursor

Designates which cursor to display when the condition is met. These cursors are the standard cursors available in QuickTime. QuickTime 3 will not display any cursors. Choosing “None” means that the cursor will not be changed by this condition. Following is a list of the available cursors:

- None
- Open Hand
- Closed Hand
- Pointing Hand
- Right Arrow
- Left Arrow
- Down Arrow
- Up Arrow

At the bottom of this tab there is a property which is global to all the button and mouse states.

Variable name containing info string

This property is only used when the movie is being played back in a Web Browser. The string that you type in here specifies the name of a Global variable you have defined in a QScript. The variable contains a string to be displayed within the Web Browser. Typically this string will be displayed either in the browser’s status bar or as a popup window. So if you have a variable in your QScript declared like this: “GlobalVars myStatusStringVar” then you would enter “myStatusStringVar” (without the quotes) into this field. Then somewhere in your script you could set the variable like this: “SetString(myStatusStringVar, “This is a sample status string”)” and see this string displayed in the status area of your browser.

events are always displayed in bold and contain the Event Handler ID and the Event name. To remove a Custom Event Handler from the Event Handler List, you must select the event in the event list and press the delete key.

Custom Events

If the above standard Event Handlers do not provide you with what you need there are also custom Event Handlers. Custom Event Handlers are created using the “New Event Handler” button. A custom Event Handler operates the same way the standard Event Handlers do except that you decide when they will be triggered. To trigger a custom Event Handler you use the `ExecuteEvent` QScript command with the ID or name of the custom Event Handler.

Each sprite can have its own set of custom Event Handlers. They can have the same ID as custom Event Handlers in other sprites but operate independently. The `ExecuteEvent` QScript command targets a specific sprite so each sprite can respond differently to the same custom Event Handler. If you would like to trigger an event within the current sprite, a call to the `ExecuteEvent` command will do the trick. For example:

```
ExecuteEvent(2000)
```

This example will trigger a custom event with the ID of 2000 in the current sprite. If you want to trigger the same event in a different sprite, use one of the QScript commands that reference a sprite. Here is an example:

```
SpriteNamed("Our  
Test").ExecuteEvent(2000)
```

Here we have indicated that we want the event with ID 2000 to be triggered in the sprite named “Our Test”. This idea could be carried even further. Let’s suppose we have the same two sprites but they are in separate Sprite Tracks. In this case we need to identify the Sprite Track and the Sprite. This example would be written as follows:

```
TrackNamed("Second  
Track").SpriteNamed("Our  
Test").ExecuteEvent(2000)
```

This example refers to a track called “Second Track”. You need to put in whatever name you have used for your track. You could also use the `TrackOfID` or `TrackOfIndex` keywords. For more information on these and other QScript commands refer to the “Appendix I - QScript Reference” PDF documentation.

this icon will not be displayed. Each of these icons will show the associated information for the selected Behavior when it is clicked. To hide the information click on that icon again.

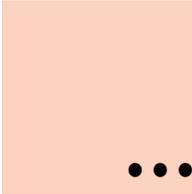
Parameters

Some Behaviors may require information in order to perform their task. Behaviors get this information in the form of parameters that are entered in the Behaviors Tab. To view the parameters for a Behavior click on the parameter's icon  This will expand the Behavior view to show any parameters for the current Behavior.

You can see from the picture above that each parameter is listed under the sprite with a name followed by an edit field to enter the value for that parameter. Although parameters typically have default values, you will usually want to change the parameters to get the desired Behavior. You may change a parameter's value by entering a value in the parameters edit field. Although any value may be entered, parameters typically have a range of values that provide reasonable behavior. Most parameters list a reasonable value range in the parameter name as shown in the picture above.

Removing a Behavior

Removing a Behavior is simple. Select the Behavior in the Behaviors Tab and press the delete key.



Chapter 9 Text Tracks



Overview	117	Hyper Text (Hotspots)	120
Text Track Properties	117	Scripting Text Tracks	122
Using Text Tracks	118	Properties Tab	123
Text Tab	119	Layout Tab	125
 Displaying Raw Text	120	Chapter Tracks	128
 Fonts, Styles and Sizes	120		

Overview

Text tracks are scriptable tracks that facilitate the display of textual information. A Text sample contains the text, font specifications, background color and additional presentation parameters needed to render itself.

Text samples can also be animated via scrolling. This capability can be used to scroll in news flashes or scroll-up credits.

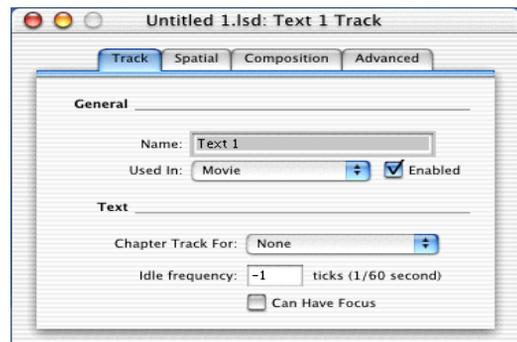
As the Text is rendered in the movie at playback time, the file size is greatly reduced when compared with a bitmap representation of the Text. However, the display of the Text is dependent on the user's machine having the specified font. In the event that the specified font is not present a standard replacement will be used. Under normal circumstances, it is best to test your project on various browsers on different platforms to avoid surprises in text size and style.

Ranges of Text can be specified as hotspots and can be scripted to perform such actions as retrieving a Web Page in a browser, playing other audio/visual media or sending of a CGI request to a Web Server.

Starting with QuickTime 5 text tracks can also be edited and can accept keyboard input from the user while the movie is playing. To respond to user input QScript is required. These new capabilities greatly expand QuickTime's usefulness especially in the areas of user input and feedback.

Text Track Properties

As with all tracks, the properties for a Text Track may be viewed by double clicking on the Track Header in the Tracks Tab. The Text Track property window displays general information about the track including its name, preload information, etc. Since this is a visual track type there are also the standard Visual Track properties which include the Drawing Mode, Matrix and Layer of the track.

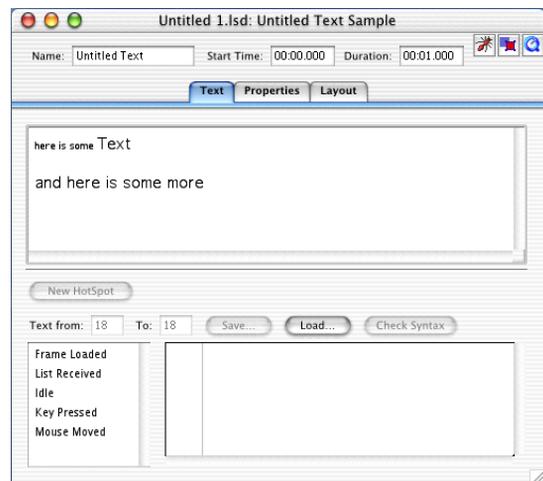


For more information on the standard track properties refer to Chapter 9, “Text Tracks” on page 115.

The properties unique to Text Tracks are the Chapter Track property, the Idle Frequency and the “Can Have Focus” checkbox. The Chapter Track property indicates that the Text Track is to act as a Chapter Track for the track specified in the popup menu. A full description of what Chapter Tracks are and how you use them may be found later in this chapter in the section titled “Chapter Tracks” on page 128. The Idle frequency indicates how often the Text Track will receive Idle events. The “Can Have Focus” check box will allow the Text Track to receive keyboard focus if it is checked.

Using Text Tracks

Like all other editable track types in LiveStage, Text Tracks are created by selecting Create in the Tracks menu and selecting the appropriate track type. Once the Text Track has been created there will be a default sample available for you to work with. Double clicking on the Text Sample presents you with the Text Sample window that contains all the settings available for the sample. In the sample window are three tabs used to organize the information in this window.



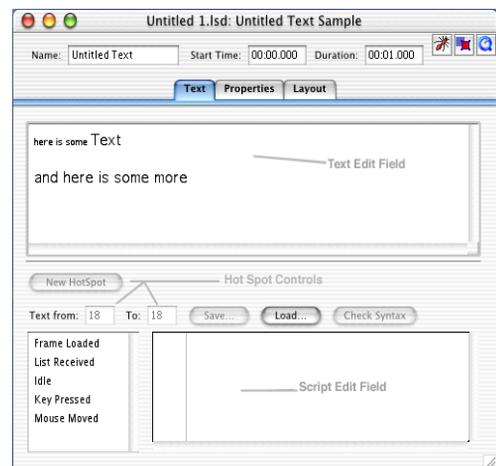
The Text tab provides the controls for text editing and hotspot creation and management. The Properties Tab contains all of the text properties which primarily affect the display characteristics of the text. The Layout Tab contains information on how to layout the text and other positional and size information related to the text.

Aside from the three tabs which handle the primary Text Sample data there are also properties displayed across the top of the Text Sample window. These fields are common to all sample windows and include the sample's Name, Start Time and Duration. A sample's name is used for display purposes in the Tracks Tab and is primarily for organization. The Start Time and Duration fields display an accurate time value in the movie for the sample. By editing these values you can precisely set up the timing information for the sample. Any changes made to these fields are immediately shown in the Tracks tab.

Text Tab

Contained in the Text tab are the Text Edit Field and the HotSpot Editor. Here you edit the text, in the Text sample the hotspots for your text and add scripts to handle events triggered by the text sample.

The text you want to be displayed in the text sample is entered into the Text Edit Field. In the Text Edit Field you will see the text much like it will appear in the movie. All font styles and hotspots are rendered. To change the font settings for any text in the Text Edit Field, select the text and then choose your settings from the Font menu. All changes are immediately reflected in the Text Edit Field.



Displaying Raw Text

In their simplest form Text Tracks may be used to display text within a movie. You may for example wish to display some informative text when the user moves over certain hotspots within the movie. A Text Track may also be used to create scrolling credits in of your movie.

To use a Text Track in its simplest form you need to create a Text Track, double click on the Text Sample and enter some text into the Text editor window. Running your movie will now display this text in your movie window. If you would like to have the text display over top of another track you can set the Drawing Mode of the track to render transparently or select the “Keyed Text” option in the Properties tab.

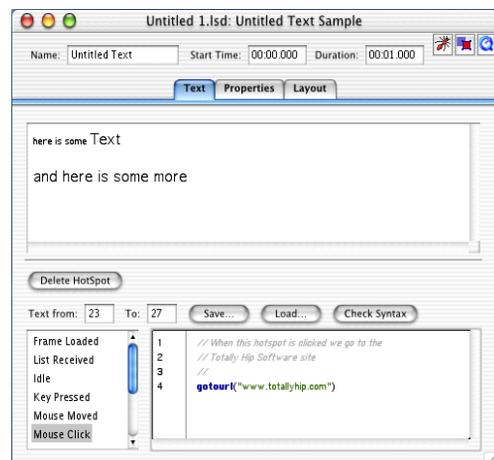
Fonts, Styles and Sizes

Typically when you use text in your movie you may want to use different fonts and styles. Text Tracks allow you to mix different fonts and styles together using a very simple interface. To change the font, style, size and color, select part of the text and choose the different settings you want to use from the Font menu in the menu bar. The styles you have specified will be displayed for you in the Text Edit area of the Text Sample window.

Hyper Text (Hotspots)

Below the text edit field is the HotSpot Editor that incorporates a variety of controls used in making hotspots within your text. Hotspots are areas of the text that can be interacted with. The Text Sample as well as each hotspot you create can be scripted using QScript to perform almost any task.

To create a hotspot you simply select some text in the Text Edit Field and click on the “New HotSpot” button. The hotspot text will now be drawn inside a surrounding box. Note that once a piece of text is part of a hotspot you



cannot select it as part of another hotspot. You can now enter QScripts into any of the Event Handlers of a hotspot in a Text Sample, Just like you would script sprites in a Sprite Track. You can script the Frame Loaded, List Received, Idle, KeyPressed and Mouse Moved Event Handlers without first creating and selecting a hotspot. These are known as Track Event Handlers.

This hotspot functionality leads to some very interesting possibilities. You may use this approach to create hot links just like you do in Web Pages, but you may also create scripts that interact with sprites, VR elements or other elements in your movie.

New HotSpot

The “New HotSpot” button becomes active when you have selected a block of text in the text edit field. Clicking on the “New HotSpot” button creates a hotspot for the piece of selected text. This button will change to “Delete HotSpot” if you have a hotspot currently selected. Clicking on the button at this time will remove the current hotspot.

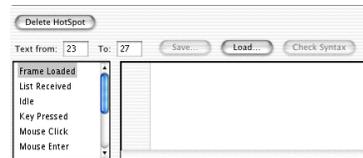
Text From... To...

The Text From and To are two fields that indicate the starting and ending character positions for the current hotspot. If no hotspot is currently selected these fields are disabled. You can adjust which characters are part of the hotspot by editing these values.

Event List

The event list displays the Event Handlers available. The Text Track has some basic Event

Handlers and when a hotspot is selected the Event Handler list will reflect that by displaying additional Event Handlers. These Event Handlers operate much like Event Handlers do in Sprites, the only difference is that you cannot have custom event handlers in hotspots. Clicking on an item in the Event List selects that handler at which point you may enter QScript statements. For more details please refer to the “Event Handlers” on page 85 in the Scriptable Tracks chapter.



Script Edit Field	The Script Edit Field becomes active when you have an Event Handler selected. In the Script Edit Field you enter QScript statements that will be executed when the event is triggered.
Save	The Save button becomes active when there is a script being displayed in the Script Edit Field. Clicking on the Save button allows you to save the script to disk for later retrieval.
Load	Whenever the Script Edit Field is active the Load button is active. Clicking on the Load button allows you to retrieve a saved QScript from disk and have it loaded into the current Event Handler.
Check Syntax	The Check Syntax button is active whenever there is a script being displayed in the Script Edit Field. Clicking on the Check Syntax button will verify that the currently displayed script is syntactically correct. If there are errors, the Error Window is displayed.

Scripting Text Tracks

If you are adding scripts for the Text Track Event Handlers for the Text Track, select the Event Handler in the Event List and then add your script in the Script Editor. If you are adding script for one of the hotspot Event Handlers, select the hotspot and then select the specific Event Handler before you add your script. If you had an event handler selected for a hotspot, selecting another hotspot will set the Script Editor to edit the same handler in the new hotspot. The Event Handler will be displayed in bold if there is script entered for that Event.

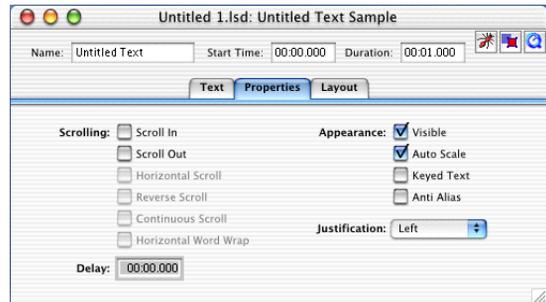
In order to script the Text Track you will need to specify a track and movie target. If no movie target is specified then the current movie is considered to be the target. If no track target is specified then the track that contains the object that is currently executing the script is considered to be the target.

To familiarize yourself with QScript refer to “Introduction to QScript” on page 219.

There are several ways to specify movie and track targets. For details on this please refer to the Target section in the QScript Reference documentation.

Properties Tab

The Properties Tab contains the visual properties for displaying the Text Sample. These consist primarily of how the text is displayed and scrolled into the viewable area.



Scrolling

The Scrolling group contains several properties that allow you to enable or disable various scrolling characteristics of the text sample. When scrolling is enabled the text will by default scroll vertically from the bottom of the text track to the top. The direction of the scrolling, timing, etc. may be modified with the properties in this group.

Scroll In

When checked this property indicates that the text will scroll into view during the time the text sample is playing. The text will scroll from the bottom of the track or text box (if set) to the top. The duration of the text sample controls how long the text will be held after scrolling all the way in.

Scroll Out

The text will scroll out of view during the time the text sample is playing. If Scroll In is also selected, then the scrolling time is divided between them with the text first scrolling in, then scrolling out.

Horizontal Scroll

This property alters the default scrolling action to make the text scroll horizontally instead of vertically. Each line of text will occupy exactly one line as it scrolls. You can change this by turning on horizontal word wrap.

Reverse Scroll

By default when using scrolling the text will scroll either from the bottom of the text track to the top or from right to left if horizontal scrolling has been

selected. Setting this property will reverse the direction so that the text will scroll from top to bottom or from left to right.

Continuous Scroll

If set, the next sample will cause this sample to scroll out. You will see both samples displayed at once.

Horizontal Word Wrap

If you are using horizontal scrolling then each line of text will scroll in one long line. Turning this option on will cause the text to be formatted to fit into the track bounds or text box (if set) and then scrolled using that formatting.

Delay

The delay field allows you to specify a time duration that affects the timing of the text scroll operations you have specified. The delay time is the duration of the that the text will be displayed in the sample (i.e. no scrolling). Any scroll in or scroll out operations for this sample then have whatever time is left over in the duration of the text sample.

For example, if your sample duration is 1 second and you set a delay of 1/2 second, there is 1/2 second left for scrolling operations. If you only have a scroll in or scroll out specified, that scroll operation would be performed in the remaining 1/2 second. Now, if you had both the scroll in and scroll out operations specified each scroll operation would take 1/4 second since there is only 1/2 second for two scroll operations.

Appearance

Visible

Unchecking this will cause the text in the Text Sample to be hidden until another sample is reached that has this property checked.

Auto Scale

Checking this will scale the text sample if the movie or track dimensions are changed. By default this property is checked.

Keyed Text

Checking this causes the background for the text to become transparent.

Anti Alias

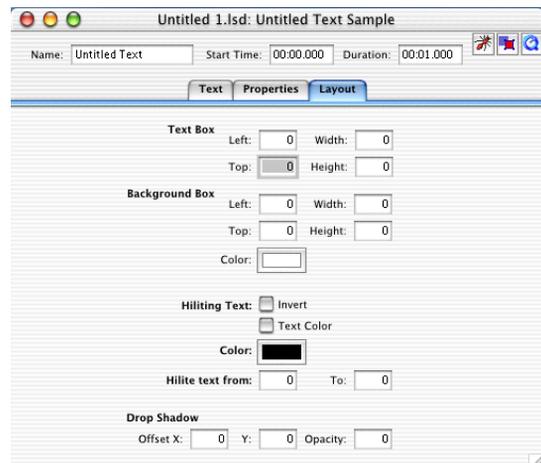
Checking this will tell QuickTime to perform anti aliasing on your text when it is displayed. For larger type faces this can make the text look cleaner and more readable. By default this option is not turned on.

Justification

The justification popup menu controls how the text is justified in either the text box or the track dimensions. The options for justification are Right, Left and Center.

Layout Tab

The Layout Tab contains all of the properties regarding layout of the Text Sample. These consist primarily of text dimensions, background dimension, background color, hilite information and drop shadow information.



Text Box

The Text Box group contains four fields that are used to designate a bounding box for the text. Any text you enter for the sample will be constrained to the area specified by the bounding box coordinates. Entering zero for all values will disable the text box.

Top, Left, Width, Height These fields allow you to enter the position and size of the text box. You should always ensure that your text box is within the boundaries of the text track and the background box or you may get unexpected results.

Background Box

The Background Box group contains four fields that are used to designate a bounding box for the background. Any text you enter for the sample will be constrained to the area specified by the background box. The background color will only display within this box.

Top, Left, Width, Height These fields allow you to enter the position and size of the background box. You should always ensure that your text box is within the boundaries of the text track and the background box or you may get unexpected results.

Color This property indicates the background color that will be used for the text sample. By default the background covers the rectangle defined by the track's position and size. If you specify a rectangle in the Background Box property then the background color will be constrained to that rectangle.

Hiliting Text

The Hiliting Text group contains several properties that are used to define a section of the text to be highlighted. Highlighting a section of text will display that text in a different color so that it stands out from the remaining text.

Invert	Setting this property will cause all highlighted text to be displayed inverse. This means that black text is displayed in white, etc.
Text Color	Setting this indicates that the hilite color will refer to the color specified in the “Color” chip.
Color	This color chip indicates what color will be used to highlight the text when you are using the highlight options.
Hilite textfrom...to...	These two fields indicate the starting and ending positions of the highlight.

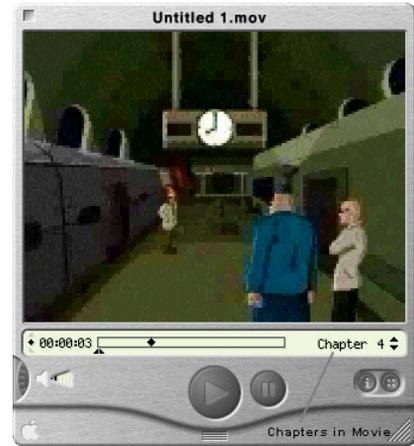
Drop Shadow

The Drop Shadow group contains properties that are used to enable and alter the drop shadow effect. A drop shadow gives the effect of a light source that is being directed towards the text and is then creating a shadow.

Offset X and Y	The x and y values specify how many pixels down and to the right to offset the drop shadow.
Opacity	The opacity specifies how dark the drop shadow is (0 - 255). Zero is off and 255 is black.

Chapter Tracks

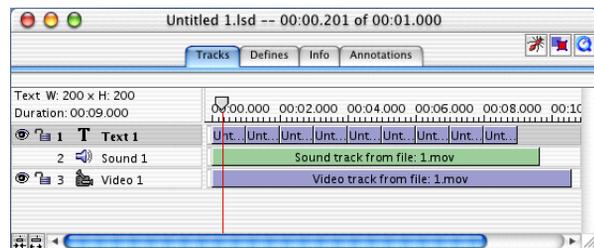
Chapter Tracks act as labels for specific times within your movie. When you create a Chapter Track, the labels you use for each chapter appear in the movie controller. By selecting different chapters you will go to different parts (times) of the movie. You can see the current chapter title displayed in the controller in the picture to the right.



Creating Chapter Tracks

You create a Chapter Track by creating a new Text Track. Each sample in the Text Track represents a single chapter and the start time of the chapter is determined by the sample's start time.

For each chapter you need to provide a title, this is what the user will see in the movie controller when they play your movie. The text you enter into a Text Sample will be used as the chapter title.



By default the text in each sample would be displayed in the movie, in the case of a Chapter Track the text should not be visible. To correct this problem you can simply disable the text track.

Finally, in the track properties window you will see a chapter track popup menu. Select the track that you want this text track to be a chapter track for.

Overview

This chapter covers QuickTime VR movies, what they are and how they are incorporated into LiveStage projects and QuickTime movies.

We will start this chapter with a description of what a QuickTime VR movie is and the kind of things you can do with them. Following this is a walk through on how you incorporate a VR Movie into your project.

What is a QuickTime VR Movie?

QuickTime VR is an extension to QuickTime that allows users to interactively explore three-dimensional virtual worlds. QuickTime VR does not require special virtual reality devices such as goggles or gloves, instead the VR movies are navigated using a keyboard or mouse. The images displayed in QuickTime VR movies can be either captured photographically or rendered on a computer using a three-dimensional (3D) graphics package.

QuickTime VR movies are made up of a single scene that contains one or more nodes. A node is a position in a virtual world at which point an object or a panorama can be viewed. A node can be an object or a panorama but not both. An Object in a VR movie provides a view of a single object. This object can be viewed from different angles using the mouse or keyboard. A panorama provides the user with a panoramic view of a particular location, much like you would get by standing in one place and turning around 360 degrees.

When a QuickTime VR movie contains more than one node, the user can move from one node to another by clicking on hotspots created in the movie. A hotspot transports you from one node in the movie to another node; this gives the effect of walking or moving through the virtual world.

The term “movie” in QuickTime typically refers to media that is time-based, however in the context of VR the term “movie” is used just for consistency. QuickTime VR movies are not time-based.

value of -1 in this field indicates that you do not require idle time. You should always use as large a value as you can for the idle time since idle time processing can slow down a movie.

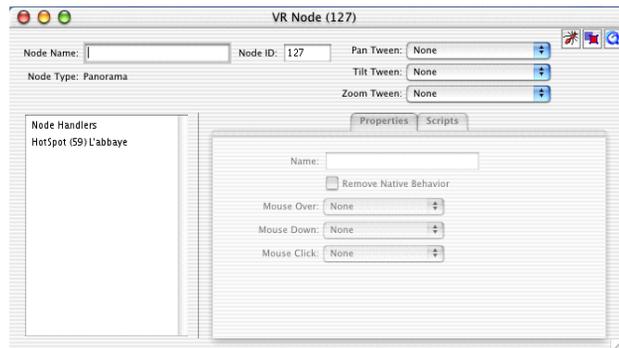
Duration

Reflects the duration of the VR Track. The default value is the duration specified by the original VR Movie. You can only change the duration of a Panorama VR movie. Object movies will not work if you change their duration.

VR Tracks also have a Matrix group just like all other visual track types in LiveStage. What makes the Matrix group special in VR Tracks is that you can specify negative values for the width and height fields. Using negative values in these fields will invert the VR Track either horizontally, vertically or both.

VR Track Samples

As mentioned previously, samples displayed in a VR Track represent the nodes contained within the movie. Just like samples in the other track types you can double click on a VR Sample to view its properties.



The VR sample window operates in a similar manner to the Sprite Sample window. This window is laid out with some general sample information across the top followed by a list of hotspots along the left-hand side and the properties associated with the current hotspot on the right hand side.

A VR Sample window contains a number of elements that allow you to set up various properties and script event handlers. Let's take a look at the VR Sample window and the components in it.

General Properties

There are properties presented across the top of the VR Sample Window which allow you to set/view standard sample information. Following is a list of fields described in detail.



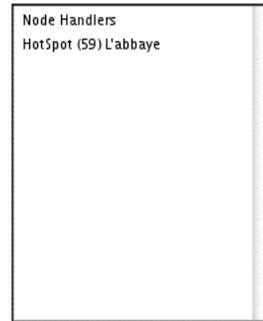
Node Name:	<input type="text"/>	Node ID:	<input type="text" value="127"/>	Pan Tween:	<input type="text" value="None"/>
Node Type:	Panorama			Tilt Tween:	<input type="text" value="None"/>
				Zoom Tween:	<input type="text" value="None"/>

Node Name	This is the name used for the Node/Sample and is displayed in the Tracks Tab.
Node ID	The ID of the node. This is used by the GotoNodeByID QScript command and is also displayed in the label of the VR sample within the Tracks Tab.
Node Type	This is a display only field that indicates what type of Node you are editing. (object or panorama)
Pan Tween	This setting allows you to set the tween source to use for the horizontal panning of the VR movie.
Tilt Tween	This setting allows you to set the tween source to use for the tilt or the vertical panning of the VR movie.
Zoom Tween	This setting allows you to set the tween source to use for the zooming in and out (Field of View) of the VR movie.

Hotspot list/Node handlers

Hotspots are areas in a QuickTime VR movie that enable the user to interact with the movie. LiveStage does not create hotspots in a QuickTime VR movie: your VR creation software must create these. Traditionally hotspots are used for navigating within a QuickTime VR movie or jumping to Web locations by linking to a URL. LiveStage, however, offers you a lot more freedom in the type of functionality you can associate with a hotspot.

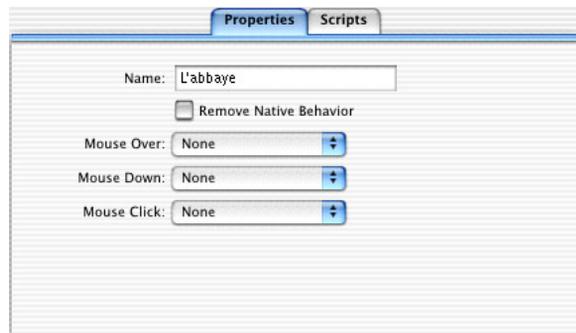
The hotspot list displays all of the hotspots contained in the VR Sample (node). There is also an additional entry at the top of the list called Node Handlers that contain information that is global to the VR Node. Selecting an entry in the hotspot list enables the Properties and Scripts Tabs so that you can set the properties and scripts associated with the selected hotspot.



Properties Tab

The Properties tab is used to set the values for various properties of a hotspot. This tab becomes active whenever there is a hotspot selected in the hotspot list.

Here is a list of the properties available to you through the Properties Tab and a description of what they do:



Name

Contains the name of the hotspot you are currently editing. This name is displayed in the QuickTime Player when the mouse is over the hotspot. The name is also displayed in the hotspot list in the VR Sample window.

Remove Native Behavior

When this is checked any actions defined for the hotspot in the VR movie are disabled. This allows you to reprogram the hotspot to perform actions other than those specified in the VR movie. If you want to augment functionality that already exists in the hotspot then you should leave this setting unchecked. If you wish to completely replace the hotspot's functionality then check this setting.

Custom Cursors

This is a group of properties that indicate what cursors are to be displayed when the mouse is in various states while over a hotspot. Each state (listed below) contains a popup menu containing all of the available custom cursors. To find out how you create custom cursors in LiveStage Professional, please read the section on “Custom Cursors” on page 137.

The available mouse states for custom cursors are:

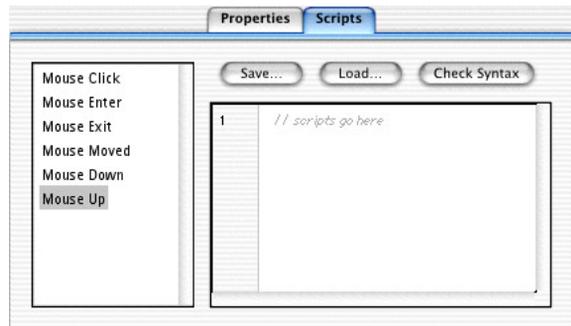
Mouse Over The mouse is currently over the hotspot but the mouse button is not being pressed.

Mouse Down The mouse is currently over the hotspot and the mouse button is being pressed.

Mouse Click The mouse is currently over the hotspot and has been clicked.

Scripts Tab

The Scripts Tab is where you enter any QScript statements that are to be associated with the currently selected hotspot or Node Handler. There are two primary sections in this tab, the left-hand side contains a list of Event Handlers and the right hand side contains a Script Editing Window where the QScript statements are entered.



The Event Handler list can be hidden by holding down the Control key and clicking on the separator bar between the list and the Script Editor. A separator bar will appear to the left of the Script Editor when the QTVR Event Handler list is hidden. Clicking on that separator will bring the list back.

If you have a Node Handler selected while viewing the Scripts tab, the event handler list will display a Frame Loaded and Idle event. Having a hotspot selected will present you with a longer list of event handlers that include Mouse Down, Mouse Click, etc.

Scripting VR Tracks

Once an Event Handler has been selected you may enter your script statements into the script editing window. If you had an Event Handler selected for a hotspot, selecting another hotspot will set the Script Editor to edit the same handler in the new hotspot. Like all other script editing windows in LiveStage, there are couple buttons above the window that allow you to load and save your QScripts. The Event Handler and hotspot in the Node Handler List will be displayed in bold if there is script entered for that Event within that hotspot.

Scripting support for VR Tracks gives you a lot of freedom to integrate various media elements with one or more VR Tracks, however, there are a couple items that you must be aware of when writing your scripts.

- In QTVR Tracks the use of variables other than MovieVars is not supported. This includes the use of the LocalVars, SpriteVars and GlobalVars. Using the other variable declaration statements will give you an error.
- QTVR Tracks also can not have custom event handlers.

Custom Cursors

A Custom Cursor is used to replace the default pointer cursor. QuickTime uses a standard set of cursors for representing various actions within a VR movie. LiveStage lets you provide your own custom cursors for some of these situations thus allowing you to customize the user's experience.

Adding custom cursors into LiveStage is a simple process; all you have to do is copy your cursor resource files into the Cursors folder located in the LiveStage Professional application folder. In order to ensure that your cursors operate properly in both the Macintosh and Windows environments, you will need to

provide both color and black and white versions of the cursors. This is because QuickTime does not support the use of colored cursors when playing back in the Windows environment.

LiveStage does not provide a way to create your custom cursors at this time so a third party tool will be required if you wish to create your own cursors. You may however choose to use cursors that have been created by other individuals. These can be downloaded off the Internet. When creating your own cursors be sure that their IDs are from 1001 onwards, otherwise LiveStage will not recognize them.

Overview

Non-scriptable Media Tracks are tracks which contain media that can be used in the creation of QuickTime movies. These tracks can be visual or audible but are not interactive. Although these tracks cannot be scripted, they can be controlled by Scriptable Tracks via the use of QScript Actions and Properties. Refer to the QScript Reference for further information on the QScript Actions and Properties available for each of the Non-scriptable Media Track Types.

Standard Non-Scriptable Media Tracks

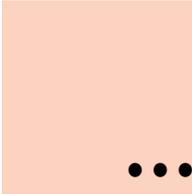
LiveStage Professional can utilize a number of Non-scriptable Media Track types. The standard non-scriptable track types are the Instrument, Effect, Picture, Color, Movie, Skin and Streaming Tracks. The following chapters provide details for each of these track types.

External Tracks

External Tracks are tracks that can be incorporated into a LiveStage Professional project but cannot be edited. External Tracks can only be added by dragging them into the Tracks Tab from the Finder or the Library.

External Tracks currently include Video and Audio Tracks. In the future, other external track types will be available for use in your QuickTime movies. All you have to do is to drag and drop the file into the Tracks Tab to incorporate these new track types into your movie.

External Tracks can be previewed by double clicking on the media sample in the Tracks Tab. However, you cannot edit the media contained in these tracks directly within LiveStage Professional.



Chapter 12 Instrument Tracks



Overview	145	Listening to Your Instruments	148
Creating an Instrument Track	145	Instrument Track Properties	149
Adding Instruments	147	Using Instruments	149
QuickTime Synthesizer Instruments	148	Exotica	151
Sampled Instruments	148		

Overview

An Instrument track enables QScript to play chords, create audio or audio feedback or manipulate tones to create new user experiences.

An Instrument track can contain one or more MIDI instrument specifications. An instrument specification includes an instrument category such as “Drum Kits”, or “Brass” or “Ensemble” and an instrument such as “SFX Kit”, “French Horn” or “Aah Choir”. An Instrument Tracks contain one or more Instrument Samples. An instrument sample may contain multiple instrument specifications.

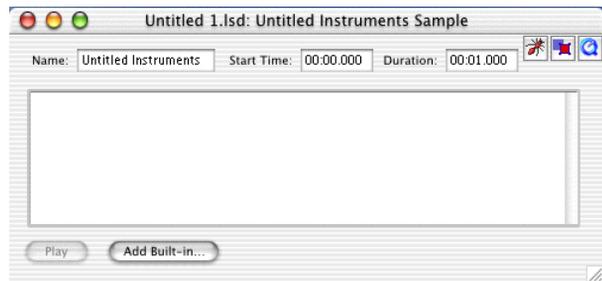
An instrument sample may also contain uncompressed digital audio. There is currently a limit of 256k per audio clip.

Instrument samples are accessed from QScript. When referring to the individual instruments from QScript an index is used which starts with the first instrument being index one. The QScript action used is PlayNote.

Creating an Instrument Track

To create one or more instruments you first need an Instrument Track (see Chapter 5, “Working in the Tracks Tab” on page 67 to learn how to create a n Instrument Track). After creating one of these tracks

there will be a default sample created automatically for you. Double clicking on the sample brings up the Instrument Sample window that lists all of the instruments contained in that sample. Initially this list will be empty since there are no instruments in the sample.



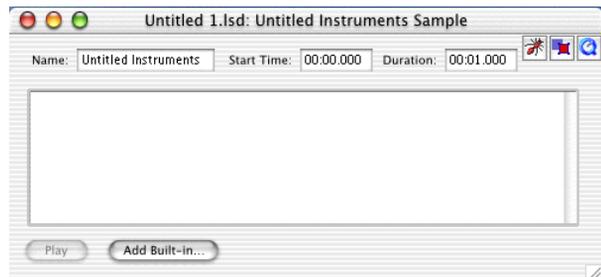
Adding Instruments

There are several different types of instruments that you can add to an Instrument Sample. These include QuickTime synthesizer instruments or sampled instruments.

QuickTime synthesizer instruments conform to the MIDI (Musical Instrument Digital Interface) standard which outlines a standard set of instruments that are always available for playback on any system that supports MIDI. MIDI is used so that people can create music and be sure that it will play back correctly on other computer systems. The instruments are built into QuickTime itself and use sampled sounds to generate instrument sounds and sound effects.

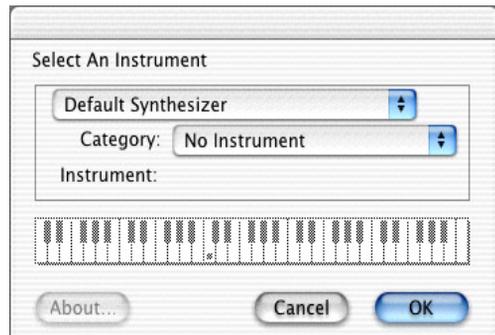
Sampled sounds are sounds you have created yourself or have obtained from another source. These come in the form of AIFF, WAV or any other digital sound format that QuickTime can import. These can be added to the instrument track and used just like the other instruments. Note however that unlike the QuickTime synthesizer instruments, sampled sounds must be stored in their entirety within the movie since they are not built into QuickTime. This will increase the size of your movie.

To add a sound to the Instrument Sample you need to view the properties of the Instrument Sample by double clicking on it. The Instrument Sample window displays the sample's name, start time, duration and a list of instruments that exist in this sample. For new samples there will be no instruments in this list.



From this window new instruments can be added to the sample. If you want to add a MIDI instrument or QuickTime synthesizer instrument to the sample, click the “Add Built-in” button near the bottom of the window. Doing so displays the “Select An Instrument” dialog which allows you to choose and preview the available instruments.

The Select An Instrument dialog is made up of several popup menus which are, for the most part, self explanatory. The first popup menu indicates the synthesizer you would like to use. By default this is set to Default Synthesizer. Using this setting provides you with all of the standard MIDI instruments. The QuickTime Music Synthesizer provides similar functionality.



QuickTime Synthesizer Instruments

For QuickTime Synthesizer instruments, the popup menu presents a list that allows you to choose a type of instrument (the Category popup), and then the actual instrument. For example, selecting the Reed Category provides you with a list of reed instruments such as the Soprano Sax, Oboe and Clarinet.

To hear a particular instrument, select it in the list, then click on a key in the piano keyboard across the bottom of the dialog. The notes you click on will be played using the selected instrument.



Sampled Instruments

Sampled Instruments come in handy when you have a unique sound that you wish to play or just can't find the right sound in the MIDI and QuickTime Music Synthesizer lists. These provide a way to create your own instruments (much like those of the QuickTime Music Synthesizer) using digital audio samples.

To create a sampled instrument, drag and drop a sound file (WAV, AIFF, SND, etc.) from either the Library Window or the Finder into the Instrument List in the Instrument Sample window. The new sample will appear in the list and may then be used in the same way as any built-in instrument.

Note: although sampled sounds work well as instruments you should be aware that QuickTime imposes a 256K sound size limit on sampled sounds. Sounds that are larger than 256K will not be added to your instrument list.

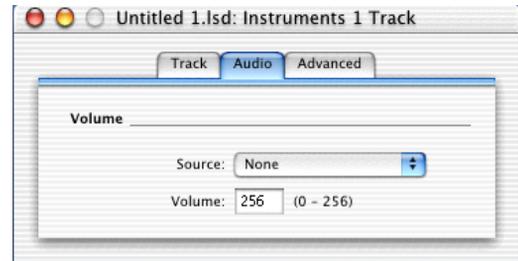
Listening to Your Instruments

If you want to listen to your instruments after they have been created you may do so in the Instrument Sample Properties window. Click on an instrument then click the Play button. In the case of MIDI or QuickTime Synthesizer instruments, a short tune will play so you can hear the instrument in action. For digital samples, clicking on the Play button simply plays back the sound as it would in any other sound program.

Instrument Track Properties

An instrument track contains the standard General Properties which are discussed in “Common Track Properties” on page 72. Instrument Tracks also contain audio properties.

The audio properties are stored in the Audio Tab of the Instruments Track Properties window. This section contains two items: the first is a Source popup menu for the sound volume and the second a Volume edit field. The source popup lists any Tween or Modifier Tracks that may be used to specify values for the volume property of this track (refer Chapter 21, “Modifier Tracks” on page 213). If you do not use a Modifier Track you may enter a value from 0 to 256 into the Volume edit field. The volume ranges from 0 (off) to 256 (full volume).



Using Instruments

Once you have one or more instruments (sampled or not) in your instrument track, you will probably want to use these instruments in your movie. You can play these instruments within a script by calling the PlayNote command.

The PlayNote command plays a note using the specified music instrument from your instrument track. The syntax for the PlayNote is as follows:

```
PlayNote(Instrument, Delay, Pitch, Velocity, Duration)
```

The *Instruments* parameter specifies the index of the instrument in the first sample of your instrument track.

The *Delay* parameter specifies a delay time before the note starts playing. The value is based on the movies Time Scale which is 600 which is equivalent to a one second delay.

The *Pitch* parameter specifies the notes' frequency. The accepted value range is from 0 to 127. The following graphic displays the pitch values equivalent to each musical note on a keyboard.

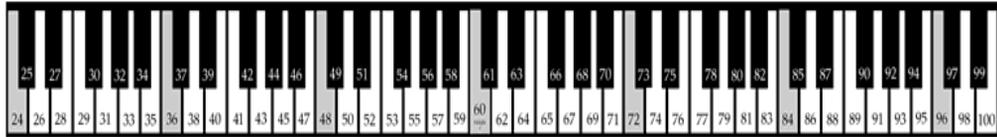


Fig 12-1: Midi Pitches for Notes

The *Velocity* parameter specifies the volume the note is to be played at. The accepted values range from 0 to 100.

The *Duration* parameter specifies the length of time the note will be played. This value is also measured in the movies Time Scale.

In order to use the PlayNote command in a script for an Instrument Track you will need to specify a track and movie target. If no movie target is specified then the current movie is considered to be the target. If no track target is specified then the track that contains the object that is currently executing the script is considered to be the target.

Here is an example of how the script will look:

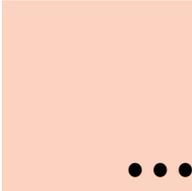
```
TrackNamed("Instrument 1").PlayNote(1, 0, 60,128,1000)
```

There are several ways to specify movie and track targets. For details on this please refer to the Target section in the QScript Reference documentation.

Exotica

In addition to using the standard instrument tracks mentioned above, you may also incorporate digital audio in your movie by adding an audio track. This works well but these tracks are only played while the movie is playing, thus are not normally flexible enough for use as sound effects.

If however, your target playback environment is going to be QuickTime 4 or later, LiveStage allows you to use the inter-movie communication commands to control other movies in order to play a sound track in them.



Chapter 13 Effect Tracks

Overview	155	Viewing Transitions	158
Creating an Effect Track	155	Filter Effects	158
Setting the Effect Properties	156	Creating a Filter Effect	159
Transitions	157	Setting the Filter Effect Properties	159
Creating a Transition Effect	157	Special Effects	160
Adjusting Transition Effect Timing	157		

Overview

An Effect Track is used to render the wide array of QuickTime visual effects available when the user has performed a full installation of QuickTime. There are single source effects which use a single visual track; dual source effects linking two visual tracks through a visual transition; and zero source effects which require no external visual media. These effects are processed on the end-user's machine during the playback of the movie. Because each frame is rendered as needed on the user's computer the file size of the movie can be dramatically smaller than the comparable pre-rendered file.

Effect Tracks require that the end-user has performed a complete QuickTime installation. This does not require the end-user to purchase a QuickTime Pro license, but that the full installation must be used on the machine where playback will occur. However, QuickTime 5 has the capabilities of detecting if a required component is missing for playback of the movie and will automatically download that component from the internet.

Given that Effect Tracks render each frame as needed the CPU requirements for smooth playback are higher than would be associated with pre-rendered video. Tests should be done on the minimum target machine to ensure quality playback.

An Effect Track is a compound track type in LiveStage Professional. This may result in more than one track being stored in a QuickTime movie file when a compound track type is exported in a LiveStage project.

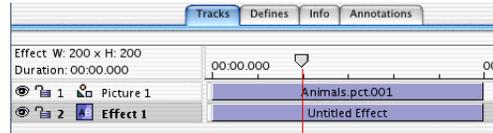
Creating an Effect Track

Creating an Effect Track is done in the same way that you create any track in LiveStage. Select the Effect Track type from the Create Menu. For more details refer to Chapter 5, “Working in the Tracks Tab” on page 67. A default Effect Sample will be added to the Effect Track for each effect you want to add.

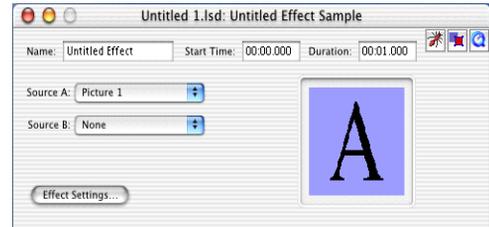
Since common uses of Effect Tracks involve one or more source media tracks, it is mandatory that the Effect Track be on a drawing layer above its source media tracks. It is also important to spatially align your source media tracks with the Effect Track so that there is no unintended visual artifacts.

Setting the Effect Properties

To set up the effect type and properties, double click on the Effect Sample. This will display the Effect Sample window.

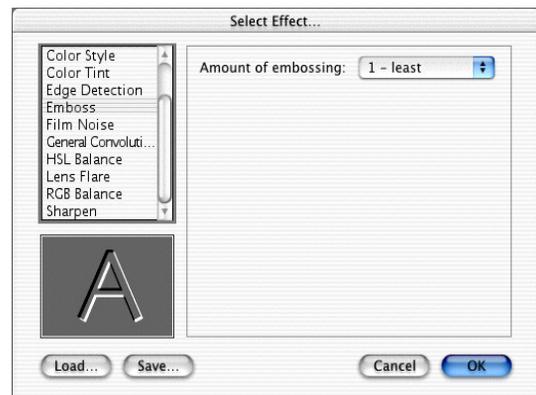


An effect can use zero, one or two tracks as sources. You select the tracks to be used in the effect from the two source popup menus. Different effect types will be available depending on how many source tracks you have chosen. Once you select the source tracks you may then choose your effect.



When two source tracks are chosen for an effect, the effect is known as a Transition effect. When only one source track is chosen, the effect is known as a Filter effect. When no source tracks are chosen the effect is known as a special effect.

To select the effect, click on the Effect Settings button. This will display the Effects Window where you choose the effect you want and set the properties for that effect.



Once you have chosen the effect, click on the OK button. You will see a mini preview of the effect in the effect sample window. You may want to name the effect for easier reference. The name you enter will be displayed in the Effect Sample. To change the name of the effect, replace the default name “untitled” in the name field with the new name.

Transitions

Transitions are most often used for linking two visual elements together smoothly. As an example you may want to fade out one piece of video while fading in a second. Transitions may also be used to create nice transitions between pictures in a Picture Track. The three sub-sections below will explain how to create, adjust and view a Transition Effect but for more details on how to use the Effects Track please refer to the Tutorial that is installed with your software.

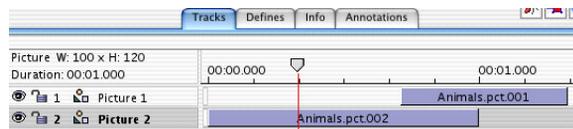
Creating a Transition Effect

To create a transition effect you need two visual tracks that are to be used for the transition. In order to create a smooth transition effect you will want to overlap the end of the first visual track with the beginning of the second visual track.

Make sure you create enough of an overlap that the transition will have time to be displayed smoothly. For a

smooth effect you should ensure that the transition lasts for at least one second. Now you can create an Effect Track and move its sample to the start of the second visual track's sample and drag the end of the Effect Sample to where the first visual track ends.

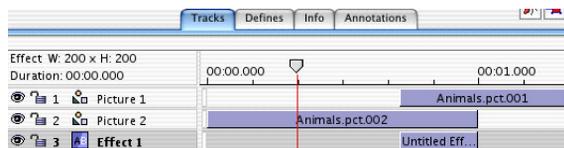
Specify the type of transition by double clicking on the Effect Sample and setting the first Source popup to the first visual track you created. Likewise, set the second Source popup to the second visual track. Setup the effect settings as mentioned earlier and then run the movie.



Adjusting Transition Effect Timing

By modifying the duration and start time of the Effect Sample you can adjust the timing of a transition effect. Generally you will need to move the sample

to the time that you want the transition to take place. Now that you have the rough positioning set for the transition you will probably need to modify its start



and end points. Dragging out the ends of the sample (thus making the sample longer) will cause the effect to take place over a longer period of time. To shorten the transition drag one or both of the end points inward, thus shortening the sample's duration. The tracks used as sources for the effect must have samples that overlap the time of the effect sample. If at any point in time that the effect sample is active there is no sample in one of its source tracks, then nothing will be displayed.

Note that if you want to perform very fine adjustments on the start time or duration of the transition effect you may do so in the Sample Window. In the Sample Window you can edit the start time and duration in a minutes, seconds, fractions (1/600ths) of seconds format. These values are entered in the Start Time and Duration fields in this window.

For details on adjusting the size and layering please refer to “Common Visual Track Properties” on page 75.

Viewing Transitions

To view your transition you need to either export your movie or use the Run Wired Movie option in the file menu. This will generate a QuickTime movie which you can then play.

The stage does not provide any preview facility for transitions and effects.

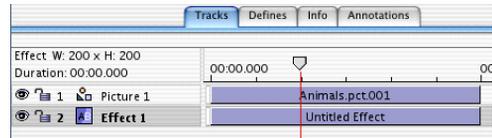
Filter Effects

Unlike transition effects, filter effects are typically applied to a visual track while it is playing instead of being used at the beginning or end to transition to another video stream. They can be used to modify the coloring of the visual track, add blur or lens flare effects or otherwise warp your visual track into white noise. All effects are applied to the visual track data in real time, that is, the resultant image is calculated only when the movie is playing and is not stored in the movie file. Filter Effects may also be applied to still images in a picture track.

Creating a Filter Effect

Filter Effects can be added to any visual track type and can be active at any point in time. Unlike transition effects where you provide two visual tracks as the sources, filter effects need only one visual source.

Filter effects are created just like Transition Effects. To create an effect, create an Effect Track in your movie and then move the default sample to the location in which you want the effect to take place. If you want to apply multiple effects over time you can either create multiple Effects Tracks or multiple Effect Media Samples.



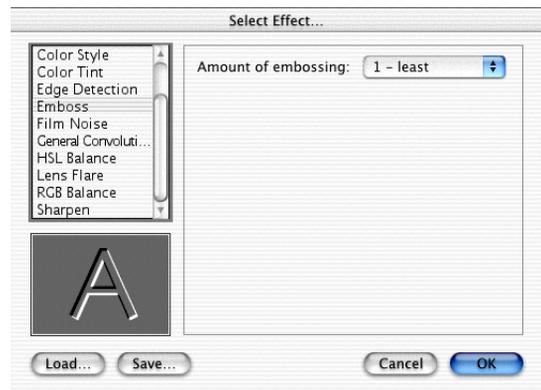
For details on adjusting the size and layering please refer to “Common Visual Track Properties” on page 75.

Setting the Filter Effect Properties

To set the effect type and properties, double click on the effect’s sample. This will display the Effect Sample window. Note that this is identical to how you set up a Transition Effect.

Unlike setting up a transition, with a filter you specify a visual track as the Source A value. The Source B will be set to “None”.

Now when you click on the Effect Settings button a different list of effects will be presented to you.



Some of the effects you will see are Blur, Film Noise, Lens Flare and Sharpen. However, there are many more effects that you can choose from. Once you have decided which effect you would like to apply, select it in the effects list. On the right hand side of the Select Effect dialog you will see a list of properties for that effect.

These properties will affect how the effect looks when being applied to your visual track. Note the preview in the bottom left corner. This gives an indication of what the effect will look like in your movie.

After you have the effect set up the way you want it, click on the OK button and you will return to the Sample Properties window. You may want to name your effect before closing the Sample Properties window. This will make it much easier for you to work with it in the Tracks Tab.

If you run the movie you will now see the effect.

Special Effects

When no source tracks are chosen for the effect you will create a special effect. QuickTime has three special effect types: fire, cloud, and ripple.

- | | |
|---------------|---|
| Ripple | Generates a ripple effect like that of a rock being thrown into a pool of water. |
| Fire | A fire effect that you can use with alpha blending to create some spectacular fire effects. |
| Cloud | Produces cloud effects for use in your movie. |

Unlike other video packages that pre-render effects while generating the output movie, QuickTime generates these effects in real-time while the movie is playing. This results in much smaller file sizes, which are ideal for Web delivery.

Overview

Picture Tracks collect and sequence still images. Each sample is a specified image. This track type can be quite useful for creating slide shows or background images in your movie. A Picture track may also be used as a source of image data for a sprite. The Codec of each sample can be set to change the compression applied to images thus giving maximum flexibility in the balance between quality and file size.

Creating a Picture Track

The creation of a Picture Track is the same as other track types in LiveStage. Click on the Tracks Tab and then select Create Picture Track from the Tracks menu. Initially the track will contain one sample that is empty. To add picture data into your picture track you will need to view the Picture Sample's properties.

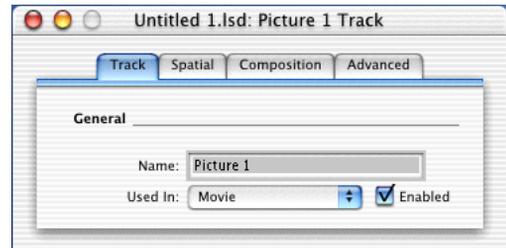
Dragging one or more images from the Library into the Tracks Tab will create a Picture Track. Be sure you drag the images into an empty area of the tab to create a new Picture Track that only contains those images. If you want to add one or more images to a Picture Track you can drag and drop those images onto the track in the Tracks Tab.

Dragging an image onto a sample in a Picture Track will replace the picture in that sample with the dropped image. If you drag and drop more than one image onto a Picture Track sample, only the first image in the group will replace the image currently in the sample. New samples will be added for all remaining images.

Each sample in a Picture Track may only hold one image so if you want to display multiple images over time you will need to add more samples into the Picture Track.

Picture Track Properties

A Picture Track, like any visual track, has a standard set of properties that you can use to modify how and where it is drawn. To view the properties for a Picture Track, double click on the Picture Track in the Tracks Tab.

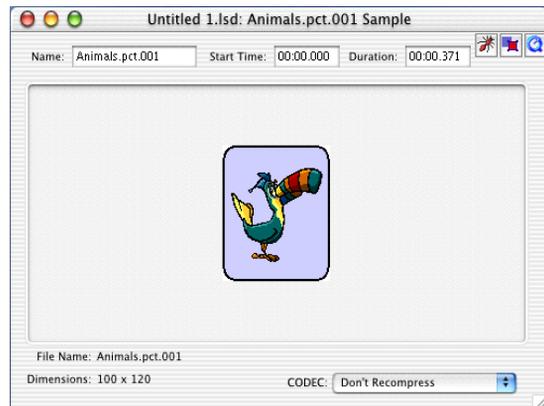


For more information on these properties please refer to Chapter 5, “Working in the Tracks Tab” on page 67.

Picture Sample Properties

Properties for a Picture Sample may be retrieved by double clicking on the Picture Sample in the Tracks Tab. A window is displayed for each picture sample.

The properties in the Picture Sample window are explained below:



Name

This is the name of the sample and is displayed in the Tracks Tab. By default the name is set to “untitled”.

Start Time

Indicates the starting time of the sample. This setting changes as you move the sample in the Tracks Tab. You may also enter a specific value here which will update the visual representation of the sample in the Tracks Tab.

The Start Time is specified as minutes, seconds, fractions of a second. The fractions of a second value are designated in the movie's time scale which is set to 600 in LiveStage Professional. To specify a full second you enter 00:00.600 or 00:01.000, a half a second is specified by 00:00.300.

Duration

The duration property indicates the sample's duration. This value changes if you drag the edge of the sample in the Tracks Tab. You may also enter a time value in the Duration property which updates the value in the Tracks Tab.

Image

The Image area is the area used to preview the image stored in this Picture Sample. If there is no picture data in the sample this area remains blank. You may set picture data in the sample by dragging an image from the Library into the Image area. You can also view the image full size by double clicking the image area to display the image in its own window.

File Name

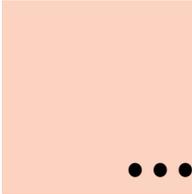
Displays the name of the file that contains the image data. You cannot edit this field.

Dimensions

Displays the size of the image in this sample. If there is no image data this field displays "0 x 0".

Codec

The Codec popup menu provides a list of codecs (Compressor\Decompressor) which can be used to recompress the image data. For a description of the compressor types refer to Appendix IV - QT Codec Reference. Most images are already compressed. Specifying a compressor setting other than "Don't Compress" for an image that is already compressed will recompress the image and add compression artifacts.



Chapter 15 Color Tracks



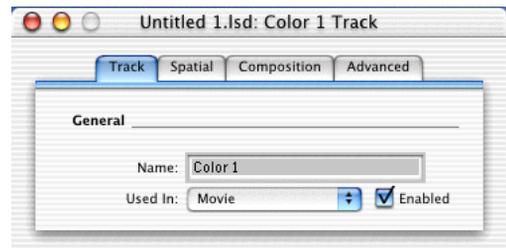
Overview	169
Creating Color Tracks	169
Color Sample Properties	169
Gradient Editor	171

Overview

Color Tracks are visual tracks that allow you to quickly create a color backdrop for your movies. They make use of QuickTime's vector Codec to render areas of solid color or gradient patterns. The data stored in the track is very small no matter what the dimensions of your track may be.

Creating Color Tracks

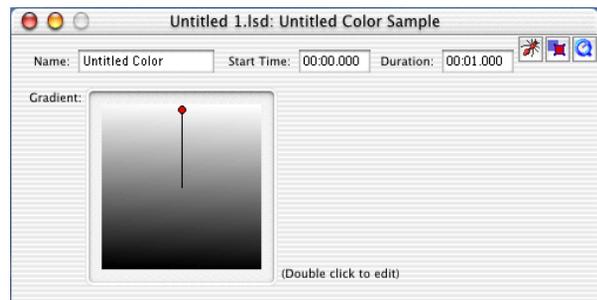
You can create a Color Track from the Tracks menu. Once created, you can set the properties for the track. A Color Track is the same as any other visual track, so you can configure it as you would any visual track. For more information on these properties please refer to Chapter 5, "Working in the Tracks Tab" on page 67.



Each sample in a Color Track can contain a solid color or a color gradient. Adding multiple samples allows you to alter or animate the gradient over time.

Color Sample Properties

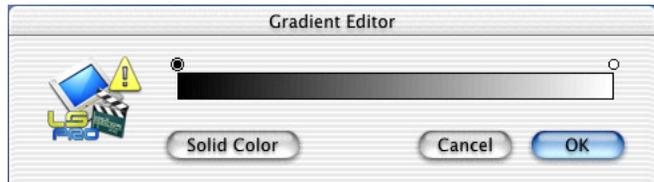
Properties for a Color Track Sample can be set by double clicking on a Color Sample in the Tracks Tab. The properties for each Color Sample are displayed in its own window thus allowing you to compare two or more samples.



Name	This is the name of the sample and is displayed in the Tracks Tab. By default the name is set to “untitled”.
Start Time	<p>Indicates the starting time of the sample. This setting changes as you move the sample in the Tracks Tab. You may also enter a specific value here which will update the visual representation of the sample in the Tracks Tab.</p> <p>The Start Time is specified as minutes, seconds, fractions of a second. The fractions of a second value is designated in the movie’s time scale which is set to 600 in LiveStage Professional. To specify a full second you enter 00:00.600 or 00:01.000, a half a second is specified by 00:00.300.</p>
Duration	The duration property indicates the sample’s duration. This value changes if you drag the edge of the sample in the Tracks Tab. You may also enter a time value in the Duration property which updates the view in the Tracks Tab.
Gradient	<p>The gradient area displays the current gradient settings for the sample. It indicates the angle of the gradient by the direction that the angle control is pointing.</p> <p>You can change the angle by clicking on the red circle in the control and dragging it to a new angle. The control will snap to 30 degree angles. To prevent snapping, hold down the option key when setting the angle.</p> <p>If the gradient is set to be a solid color then this control is hidden. You can double click in this area to open the Gradient Editor window where you can edit the gradient.</p>

Gradient Editor

The gradient editor allows you to edit the gradient, or reset your colors to one solid color. You can add or remove colors to your gradient as well as edit the color of each color in the gradient.



To edit a color, double click on the small circular or rectangular color chip just above the gradient. This will open the standard color chooser where you can choose a new color. You can also re-position any rectangular color chip by dragging it to a new position.

To remove a color, first select it by clicking on its color chip, then press delete. The selected chip is shown outlined in white. You can only delete rectangular color chips and not the two circular chips at the ends.

To add a new color, drag either circular chip. Once you release the chip, a new chip will be created at that spot and the color chooser will be shown allowing you to choose a color for the chip.

When you are done editing your gradient, click the OK button.

Clicking the “Solid Color” button will remove all rectangular color chips and set the two circular chips to the same color. The color will be the current color in the left circular chip.

Overview

Movie tracks allow you to embed several small movies into a single container movie. You can also dynamically link to other movies based on their URL. A movie playing in a movie track can play independently of its container movie. This means that time passing in one movie does not affect other parts of the movie. For example, one movie could be stopped while the other plays, or both could play. Movies that contain Movie Tracks are usually referred to as MIAM (Movie In A Movie).

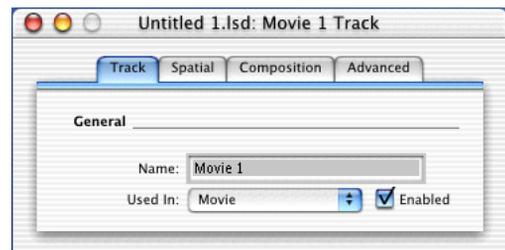
Using Movie Tracks you can create movies that have background music that plays continuously, even if the container movie is stopped. You can also create wired movies that you use in multiple projects and attach them by URL to your container movie. This makes it much easier to update all your projects by just updating your one movie. You can also use Movie Tracks to divide your project into smaller parts that are easier to manage and can be worked on by different team members.

Creating Movie Tracks

Create a Movie Track by selecting “Movie Track” from the Tracks menu. As with any other track type, a single sample will be automatically added for you.

Movie Track Properties

The Movie Track properties include all of the standard properties for a visual track. For more information on these properties please refer to Chapter 5, “Working in the Tracks Tab” on page 67.

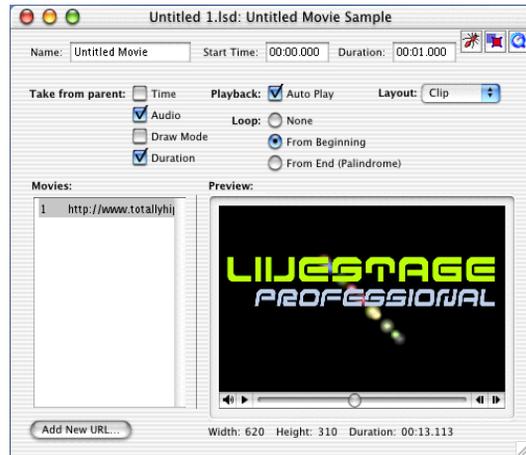


Movie Samples

After creating a Movie Track, an initial sample will be added for you. You can add more samples using the “Add Sample” command. To view the properties of a Movie Sample, double click on the sample. This will display the Movie Sample property window which contains all the settings for the sample.

At the top of this window are the standard set of fields for all tracks, the name, start time and duration of the sample. You should name all your samples to help you remember what they are. You can also fine tune the start time and duration here.

The settings following will apply to all of the movies that this sample represents.



Take from Parent

These settings allow the parent movie to establish a child movie. The parent movie is the movie that contains this Movie Track while the child movie is the movie that is currently loaded into the Movie Sample.

Time

When this is checked, time in the child movie is established by the parent. If the parent movie is stopped, so too is the child. If the parent movie plays at half speed the child movie will play at half speed as well. This is sometimes referred to as “time slaving”, that is the time and rate of the child movie is completely determined by the parent movie. If the child movie is intended to have a playback rate independent of the parent movie this setting should be turned off.

Audio When this is checked, the audio settings (volume and balance) of the child movie are determined by the setting of the parent's movie track in which the child is loaded.

Draw Mode When this is checked, the drawing mode of the parent movie will be linked to the drawing mode of the child movie. This means that the child movie will use the drawing mode that is set for the Movie Track in the parent movie. *Currently this will cause QuickTime 4.1.2 to crash.*

Duration When this is checked the duration of the child movie will be stretched or shrunk to match the duration of the Movie Sample. This means that the child movie could play at a slower rate than is intended. When this is un-checked, then the duration of the child movie will be clipped to the duration of the Movie Sample.

Playback

Auto Play When checked, the child movie will automatically start playing.

Loop

This set of radio buttons control the how the movie loops during playback.

None When selected, the child movie will stop when it reaches the end.

From Beginning When selected, the child movie will start again at the beginning when it reaches the end.

From End Palindrome) When selected, the child movie will reverse direction when it reaches either end.

Layout Popup

This popup menu allows you to select how the movie will draw in the Movie Track. The dimensions of the Movie Track may not match those of the child movie. When they do not, the settings of this popup menu tell QuickTime what to do.

None, Fill	These settings are the same. The child movie will be scaled in size to exactly match the dimensions of the Movie Track.
Scroll	This is not currently implemented (as of QuickTime 5) and currently has the same functionality as Clip.
Clip	The child movie will not be re-sized at all and if it is bigger than the dimensions of the Movie Track, then the portions below and to the right will not be visible.
Meet	The child movie will be re-sized proportionally so that it just fits within the Movie Track's bounds. If the proportions of the Movie Track do not match those of the child movie, then some blank space will show to the bottom or right of the child movie.
Slice	The child movie will be re-sized proportionally so that it completely covers the Movie Track and there are no blank areas. Portions of the child movie will not be visible.

Movies

This is the place where you specify the child movies that will be contained within this Movie Sample. There are two ways to add a child movie, one is to completely embed the child movie within the Movie Sample, and the other is to refer to the child movie by a URL.

The list to the left of this group is called the movie list. The child movies are shown in this list. You can add a child movie to this list by clicking on the "Add New URL" button. This will display a dialog where you can enter the URL for the child movie. You can also embed a movie by dragging a movie from the Library into the movie list. The file size of embedded movies needs to be very small since they will be embedded in the parent movie and QuickTime will also load them completely into RAM. Embedded movies should be less than 100K bytes in size.



When you have more than one movie in this list, you can change which movie is shown using QScript actions from a Scriptable track. This allows you to alter what is shown in the Movie Track under script control. You can also add and remove movies from this list using QScript actions. The numbers at the left of the list indicate the ID number of the movie in this list, which you will need to use in your scripts.

There is also a preview area to the right where you can preview the selected movie in the movie list. URL movies can take a while to download so patience is required. The height, width and duration of the current movie is displayed below the preview.

Using a Movie in Your Scripts

In order to use the QScript to script for the Movie Track you will need to specify a track and movie target. If no movie target is specified then the current movie is considered to be the target. If no track target is specified then the track that contains the object that is currently executing the script is considered to be the target.

There are several ways to specify movie and track targets. For details on this please refer to the Target section in the QScript Reference documentation.

Tracks which are scriptable are Flash, Sprite, Text and QTVR. The following script is an example of how the script would play the movie.

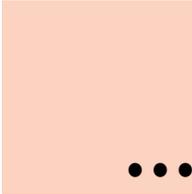
```
TrackNamed ("Movie 1" ) .GoToBeginning  
TrackNamed ("Movie 1" ) .StartPlaying
```

The following script shows you how to change a movie within your Movie Track.

```
TrackNamed ("Movie 1" ) .LoadChildMovie (10)
```

Refer to the Properties and Action for the Movie Track and Movie in the QScript Reference documentation for more details on how to script child movies.

Also see www.totallyhip.com/lsdn for more articles on MIAM.



Chapter 17 Skin Tracks



Overview	183
Creating Skin Tracks	183
Skin Track Properties	184
Skin Samples	184

Overview

The Skin Track allows you to provide a custom user interface and helps you to create a new environment or window for your movie that can be any shape or size.

A new QuickTime 5.0.1 feature, a Skin Track can create non-rectangular windows when played back in the QuickTime player or certain other players. Windows can have transparent areas and complex shapes. The Skin Track added to your movie cannot be displayed in the Stage. However, Skins can be previewed using the Movie Previewer or by QuickTime Player as an exported movie.

To describe the shape QuickTime uses a mask. The mask is a black & white image showing the silhouette of the window's shape. The black pixels in the image will describe the shape of the window while the white pixels will indicate transparency.

A second mask is needed to describe where the user can click and drag to move the window. The black pixels in this mask indicate the area where clicking and dragging would move the window rather than interacting with media within the movie.

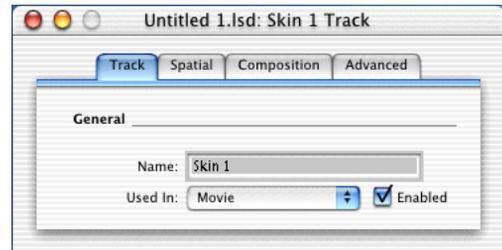
Since QuickTime 5 does not support multiple skins, the Skin Track is applied to the entire movie. You should only use one Skin Track with one Skin Sample for your movie.

Creating Skin Tracks

You can create a Skin Track by selecting “Skin Track” from the Tracks menu. As with any other track type, a single sample will be automatically added for you.

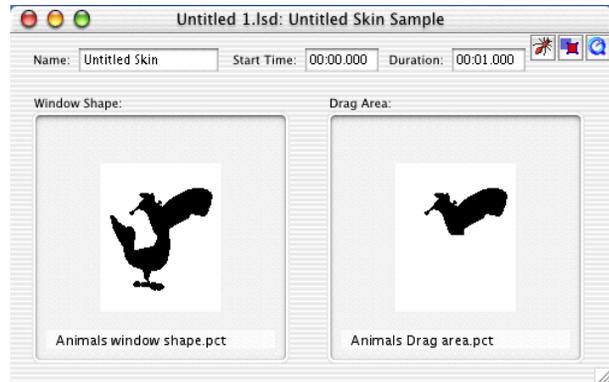
Skin Track Properties

The Skin Track properties include all of the standard properties for a visual track. For more information on these properties please refer to Chapter 5, “Working in the Tracks Tab” on page 67.



Skin Samples

An initial sample will be added after a Skin Track is created. You can add more samples using the “Add Sample” command. However, only the first sample will be used by the Movie Player. To view the properties of a Skin Sample, double click on the sample. This will display the Skin Sample property window which contains all the settings for the sample.



At the top of this window is the standard set of fields for all tracks, the name, start time and duration of the sample. You should name all your samples to help you remember what they are. You can also fine tune the start time and duration here.

The properties in the Skin Sample window are explained below:

Name This is the name of the sample and is displayed in the Tracks Tab. By default the name is set to “untitled”.

Start Time

Indicates the starting time of the sample. This setting changes as you move the sample in the Tracks Tab. You may also enter a specific value here which will update the visual representation of the sample in the Tracks Tab.

The Start Time is specified as minutes, seconds, fractions of a second. The fractions of a second value is designated in the movie's time scale which is set to 600 in LiveStage Professional. To specify a full second you enter 00:00.600 or 00:01.000, a half a second is specified by 00:00.300.

Duration

The duration property indicates the sample's duration. This value changes if you drag the edge of the sample in the Tracks Tab. You may also enter a time value in the Duration property which updates the view in the Tracks Tab.

Window Shape

The Window Shape area displays the black and white mask image used to define the shape and size of the window created when your movie is played. The black area of the mask indicates where you want to display the window and white everywhere else. You can add the image used for the Window Mask shape by dragging it into this area.

Drag Area

The Drag Area is where you define the draggable portions of the frame. This image must be the same size as the mask. You can add the image used for the Drag Mask shape by dragging it into this area.

Overview

A Streaming Track is a reference to an RTSP stream. RTSP streams can be audio, video or both. A Streaming Track is an External track since it is not created within LiveStage.

Adding Streaming Tracks

You can add a Streaming Track to your project by selecting “Streaming Track” from “Create” submenu within the Tracks menu. An Open dialog will appear and you can then select a QuickTime movie containing a Streaming Track and import into your project.

Alternatively, you can drag and drop a QuickTime movie from any folder into the Tracks Tab. This will add the Streaming Track from that movie into your project.

Streaming Track Properties

The Streaming Track properties include all of the standard properties for a visual track. For more information on these properties please refer to Chapter 5, “Working in the Tracks Tab” on page 67. However, there are additional properties contained in the Streaming Track Properties window which are unique to a Streaming Track. Double click on the Streaming Track name in the Tracks Tab of the Project Window to view and edit these properties.



In the Streaming Track property window select the Tracks Tab. There you will see an Audio Properties section that has not been discussed previously.

The Audio Properties contains characteristics that are specific to Streaming Tracks as well as settings that affect how the Streaming Track will play back in your movie. It contains the properties that are described below:

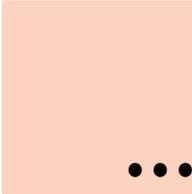
- Source** The source popup displays any Tween or Modifier Tracks that may be used to specify values for the volume property of this track (refer Chapter 21, “Modifier Tracks” on page 213).
- Volume** This edit field allows you to set the volume for the audio in the Streaming track associated with this track. If you do not use a Modifier Track you may enter a value from 0 (off) to 256 (full volume) into the Volume edit field.

Streaming Track Samples

The Streaming Sample displays a preview window allowing you to see the streaming track attached to the Streaming Sample in this Track. Some streaming tracks don't have definite durations (i.e. live stream). In order to prevent the movie from ending prematurely, specify a duration equal to or greater than the expected duration of the Live Stream.

You can open the Movie Monitor tool movie by selecting “Movie Monitor.mov” from the Tool menu. The Movie Monitor tool provides information about the any QuickTime movie selected.





Chapter 19

Data Generating Tracks



Overview

193

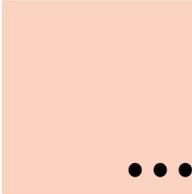
Overview

Data Generating Tracks have no visual or audio representation. A Data Generating Tracks generates a series of numeric values for use by other tracks in the project.

Data Generating Tracks provide the data for the various “Source” popup menus located in the LiveStage Professional user interface.

Data Generating Tracks can be used to create effects like changing the volume for an Audio Track, changing the image used by a Sprite in a Sprite Track, changing the pan angle in a VR Track, set a path for a Sprite in a Sprite Track and so much more.

The two Data Generating Track Types available in LiveStage Professional are Tween Tracks and Modifier Tracks. For more details regarding these tracks, read the following two chapters.



Chapter 20 Tween Tracks



Overview	197	Matrix Tween	205
What is a Tween?	197	Path to Matrix Translation	205
Tween User Interface	198	Path to Matrix Rotation	206
Tween Editor	199	Path to Matrix Translation and Rotation	206
Name	199	Path X to Y	206
ID	199	Path Y to X	207
Start & End Time	199	Polygon	207
Interpolator	200	Spin	207
Type	202	Graphics Mode Tween	208
Tween Types	203	VR Angle	208
Index / Layer / Flag Tweens	203	MultiMatrix Tween	208
Time / Event Trigger Tweens	205	Tutorial	209

Overview

Tween Tracks may be one of the most powerful features in LiveStage but they are also one of the more complex features. In this chapter we will go through a basic introduction to tweens, followed by an explanation of the various editing windows, an overview of the various tween types and finally a short tutorial to give you some hands-on experience with tweens.

What is a Tween?

In their simplest form, Tweens are used to provide a range of values algorithmically over time. This means that as your movie is playing, the Tween Track will produce values that are derived from a set of starting and ending values.

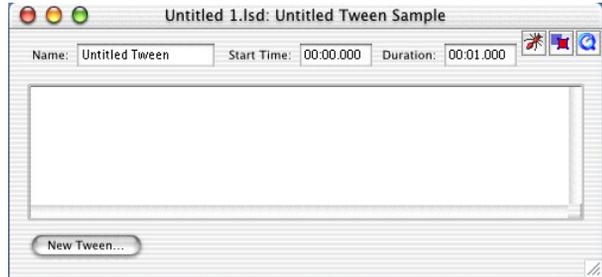
Tween tracks offer a file-size efficient mechanism for animating visual media in LiveStage. Tween tracks, or Tweens, can also be used to fade-in audio or alter the user's speaker balance. Tweens can smoothly animate a cinematographic path in a QTVR panorama. The underlying connection between these capabilities is the Tween track's ability to express change as a function of the passage of time.

There is a large number of sub-species for Tweens. For instance, Matrix tweens express changes in a visual entity's position, size, angle of rotation, or visual skew. While Drawing Mode Tweens can be used for such useful tasks as fading in/out a sprite or track, VR Angle Tweens can be linked to the pan, tilt and field of view of a panorama to animate the camera's perspective. Audio fade in/out can be controlled using an Index/Layer/Flag tween.

It is sometimes useful to store different tweens for specified time ranges. For instance, one might place a fade-in tween at the beginning of a song and a fade-out tween at the end of the song. This linkage is accomplished across samples by reusing the same ID for the individual tween.

Tween User Interface

Creating a Tween sample can be a simple process (see Chapter 5, “Working in the Tracks Tab” on page 67 to learn how to create a Tween Track). There are a couple of windows that you need to become familiar with in order to create your own tweens.



The Tween Sample window is displayed when you double click on a sample in a Tween Track.

The Tween Sample window contains some standard sample information that includes the Name, Start Time and Duration. Below these fields is a list which displays all the tweens contained in the sample. At the bottom left of the Tween Sample window is the “New Tween” button. This button is used to create a new tween in the current sample.

Name

The name field defaults to “untitled” but you can change it to anything you like. The name you enter here will be displayed in the Track List of the Tracks Tab and is strictly for your own use.

Start Time

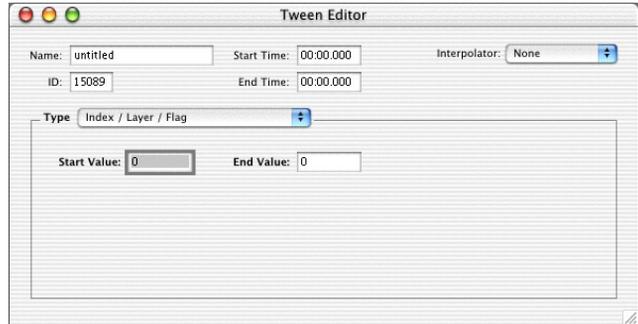
Start Time is the starting time of the Tween Sample. By changing the time value in this field you can adjust when the Tween Sample starts in the Tween Track. As you adjust this value you will see the sample move in the Tracks tab. If you move the Tween Sample in the Tracks Tab you will also see the start time update in the Tween Sample window.

Duration

The Duration indicates how long the sample is. You can edit this field to adjust the sample’s duration. This is a much more precise way of adjusting the duration than just dragging the Tween Sample in the Project Window.

Tween Editor

After you have created a new tween or double clicked on a tween in the Tween list, the Tween Editor window will be displayed. The Tween Editor window is used to specify the type of tween and the related pieces of data required for that tween type.



The first five fields you see across the top of the Tween Editor window are common to all Tween types. These fields indicate:

Name

The Name field allows you to enter a meaningful name for this tween. Any name you enter here will be displayed in the Tween Sample's tween list and Source popup menus. Be sure to put in something useful as it will be used as the sole identifier of the tween.

ID

The ID field is the unique ID for the tween object.

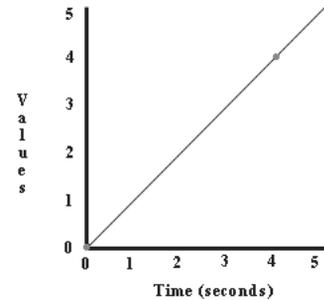
Start & End Time

The Start Time and End Time fields allow you to enter time values that will affect the results of the tween. These fields do not affect the actual starting/ending time of the tween. They are used to adjust the time values used as inputs for the tween. Let's take a look at how these values can affect the outcome of a tween. Note that the Start and End Time values are based on the sample's time, not movie time. A Start Time of "00:01.000" means that the time for the tween doesn't start until you are one second into the tween sample. The End Time value is also a time value based on the start of the sample, not movie time.

To keep it simple we will use the “Index / Layer / Flag” tween which generates simple integer values. For this example we will set the duration of the Tween Sample to “00:05.000” (5 seconds). The Start Value will be set to 0 and the End Value set to 5. For the first iteration of this example we will leave all of the other items at their default values.

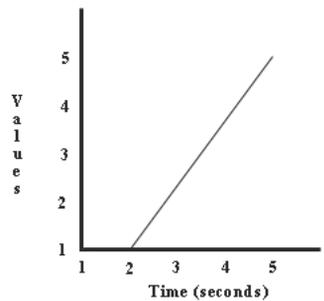
Let’s view this tween graphically, the X axis is the elapsed time and the Y axis is the value the tween will generate.

Looking at the values produced we can see that at the start (0 seconds) you will get a value of 0. As time moves along the values will increment directly in relation to the time passing. At the 4-second time in the tween a value of 4 is produced and at the end of the tween we get a value of 5.



Now lets adjust the time information for this tween to see how it effects the output of the tween. Open the Tween Editor window for the tween and set the Start Time field to “00:00.000” and set the End Time field to “00:03.000”. This tells the tween when the start and end values occur.

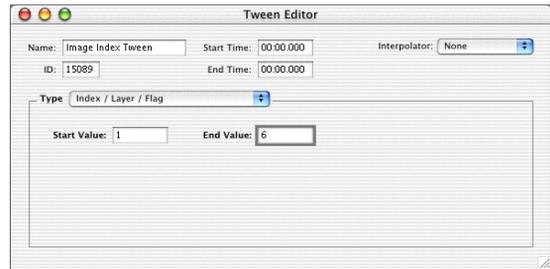
The graphical representation of this modified tween will look like the image at the right. The values produced by this modified tween will be the following. For the start of the tween we will get a value of 1, as the time moves along to the 3 second mark we get a value of 2. Note that this is different than it was with the previous version where we would get a value of 3 at the same time. Where we see things getting interesting is after the 3 second position in the movie. Between 3 and 5 seconds the tween now starts generating values lower than 5.



Interpolator

The Interpolator popup menu presents you with a list of tweens that can be used as interpolators for this tween. An interpolator acts as a time clock for a tween. It generates the time values a tween uses to calculate its current value. These time values are artificial time values that will be used in place of the actual time.

By using interpolators you can produce outputs from a tween that would otherwise be extremely difficult if not impossible. Lets look at a simple example to see how you may use an interpolator. In this example we will create an “Index / Layer / Flag” tween that generates image indexes from 1 to 6 over 5 seconds.



If you use this tween as the source for a sprite’s image index (note that the sprite sample should have 6 images) the sprite will display each image as the tween track plays. To better view the results you may want to set both the Tween Sample and Sprite Sample to start at time “00:00.000” and end at time “00:05.000”.

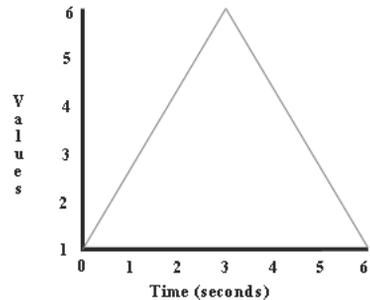
Now we can create a new tween that will be used as an interpolator for the Index tween. Create a new tween in the same tween sample and set its type to “Path X to Y”. This tween operates using the passage of time along its X axis and the value range starting at the top of the Y axis and increasing as we go down the Y axis. Click in the path area of the tween to create three points in the shape of a inverted “V”. Change the name to “interpolator” or to something else that is meaningful.

Close this Tween Editor window and open your original tween for editing. Click on the Interpolator popup menu and select the “interpolator” tween. Now you can close the Tween Editor window and run your wired movie. The behavior of your sprite will be a little different than it was before. Instead of displaying the six images in sequence you will now see the six images in the first half of the movie and then they will be displayed in reverse order.

Lets look into why using these two tweens together gives us the results it does. First, the original tween we created gives us an image index as time goes on. Here is a quick outline of the values it gives us

Time (seconds)	Value
0	1
.	.
.	.
.	.
5	6

As mentioned above, when you use an interpolator the tween takes the values from the interpolator and uses those as its time values. The interpolator tween that we created which is in the shape of a inverted “V” produces values from 0 to 5 between the 0 to 2.5 second range. Then from 2.5 to 5 seconds it produces values that go from 5 to 0 (see graph).



Interpolators can be stacked. This means one interpolator can be used in another.

Type

The Type popup menu indicates the type of tween you are working with. When you change the type of tween the view below the Type popup menu will change to reflect the data for that tween.

Some of the tween editors have edit fields where you enter numeric values, others display the tween graphically.

For tweens that are displayed graphically you may need to move the Tween Editor area in order to access path points. To do so press the Command and Option keys and click in the Tween editor area. As you move your mouse, the tween edit area will move with you. If you need to return the Tween edit area

back to its original position press the Command and Option keys and double click in the Tween edit area. This will reset the display area back to its original location.

Tween Types

In this section we will explain the different types of Tweens available, how they can be used and give examples of using them. Also note that this will go into detail with regards to the values you can specify for the different tweens and what they mean.

First we will start with some of the simple tweens you can use in everyday projects. This way you can get more familiar before getting into some of the more complex tween types.

When using a tween as the source for a property you should note that the property may not be changed by any other means. As an example, if you are controlling a sprite's Drawing Mode through a tween, you cannot enter other values for the sprites Drawing Mode. Also, any changes that you may make to the Drawing Mode through QScript would have no effect.

Some of the Tweens allow you to plot path points in the preview area displayed. You can plot out a smooth path by holding down the Option key (for Macintosh) or Control and Alt keys (for Windows) and then start plotting the line. To select a group of points, select the starting point and then hold down the Shift key and select the last point.

You can modify the Grid settings (See “Grid Menu” on page 56) and the View settings (See “View Menu” on page 57) for some of the Tween Types.

Index / Layer / Flag Tweens

The “Index / Layer / Flag” tween generates integer values in a linear manner over time. It can be used as the source for the Image, Layer or Visible properties of a sprite. It can also be used to control the volume of a Sound Track or to change images for a slide show. For example, let's say we create an “Index / Layer / Flag” tween that has a starting value of 0 and an ending value of 5. If the tween's duration in time is from time 00:00.000 to time 00:05.000 (5 second)

then the output from the tween for a time of 0 seconds is 0. The output for a time of 01:00.00 is 1, the output for time 02:00.050 is 2 and the output for time 05:00.00 is 5.

Only two values were provided for the tween, 0 and 5, and the other values were all calculated using simple linear interpolation.

Using this tween allows you to change the image index property of the sprite. This will cause the sprite to animate over time. Another use is to change the layer of a sprite as the time values change.

You could do this using a script but that would mean a lot of unnecessary work. Let's use the "Index / Layer / Flag" tween to step through a series of sequential images in a sprite over a period of time.

First of all, we need a sprite and a series of three images to put into that Sprite Sample. The images should be in the order in which you want to display them. That way we can iterate through them in a linear manner. Now we create the tween that will do the work for us. Create a Tween Track in the same way you created one in the previous example. Double click on the Tween Sample to display the Tween Sample window. Clicking on the New Tween button will create a default tween and the Tween Editor window will be displayed so you can edit the tween.

Now we need to set the tween's name; we will call it "animator". Go to the Type popup and select "Index / Layer / Flag". Notice how the fields in the Type group changed to reflect the data values that we need to enter. Now set the starting value to 1 and the ending value to the number of images you have in the sprite sample. Use 3 for the End Value in this example.

We have now created a tween that will generate values that range from 1 to 3 over the time that the tween exists in the movie. By default the sprite and Tween Tracks will only be 1 second long. If you prefer, these can be made longer or shorter by dragging the edges of the samples in those tracks.

Now close the Tween Editor and Tween Sample windows. The final step required is to link the tween that we just created to the image index property in our sprite. By doing this the sprite will get its image index values from the tween. Go back into the sprite info window and select the sprite (in the Sprite Tab) that you created. If you look at the Image Index section of the sprite

properties you will notice a Source popup which currently has a value of zero. If you click on it you should now see the tween that you created listed in there: select that tween.

We have finished with this sample. All we need to do before running the movie is go into the Info tab and make sure that the Allow Controller option is checked. Then you can use the Run Wired Movie selection from the File menu. As the movie plays each image in the sprite will be displayed.

Time / Event Trigger Tweens

A Time / Event Trigger Tween generates values that may be used to specify time values or event ID's. Event Tweens are used to trigger events in a sprite over time. This can be used to trigger events that otherwise wouldn't happen without user interaction.

Matrix Tween

The matrix tween allows you to vary starting and ending values for a series of parameters in a matrix. These include stage coordinates, percentage of scaling, percentage of skewing and degrees of rotation. For example, if you wanted to create an animation of a clock hand moving, you could use this to rotate the hand of the clock around an axis.

These permutations would allow you to rotate a sprite across the stage from a beginning coordinate position to a final position while simultaneously changing its length, width, vertical and horizontal skew. Predicting the exact outcome is difficult because of the interactions of the parameters through time, but a little experimentation will provide you with some rather spectacular results.

Path to Matrix Translation

This tween generates matrix data based on a path that you create by clicking on a representation of the stage in the Tween Editor window. For example, you could create a path for a bee flying through a field from one flower to the next. A red square represents any newly-created point. After you have created two or more points, the first point you create is symbolized by a black square. Black circles represent intermediate points. To change a path click on any point and drag it to a new location. To delete a point, select it and press the delete key on the keyboard. The selected point will be shown in red.

This tween is best used as an interpolator.

Path Y to X

This tween is similar to the Path X to Y tween except that the values are switched. With this tween the movie's time runs along the Y axis and the values are generated along the X axis.

This tween is generally used strictly as an interpolator.

Polygon

Generates polygons going from the start polygon value to the end polygon over time. This generates a matrix that can be used to morph an object. You can think of the polygon as the four corners of a sheet of rubber that has a picture on it. Pulling any of the corners will stretch and distort the image.

You may need to experiment to get the effect you desire. A good approach is to set each of the Start Values to the existing X, Y coordinate positions of the sprite, then increase all the End Values by a given factor. This will give you a predictable “zoom in” effect. You can then return to the End Values and modify them to get just the effect you want.

Spin

Generates spin data based on the starting angle and the number of rotations you want. This creates a matrix that rotates the object. You should note however that since this only produces rotation data for the matrix, a sprite using this matrix would be located in the top left corner of the Sprite Track. The Sprite Tween is most useful when used as a component of a Multi Matrix Tween.

Graphics Mode Tween

The graphics mode tween generates a range of RGB color values and a graphics mode to use in drawing operations. The results of this tween can be used anywhere that a drawing mode is required. One application is using this tween to smoothly fade a sprite or visual track in and out. For example, you could use this to show a change in color in a cartoon face from being red to a pale white color.

To use a Graphics Mode Tween, select the tween in the Drawing Mode Source of that item that is to use the tween.

VR Angle

Generates values of type float which are used for specifying viewing angles in VR tracks.

MultiMatrix Tween

This tween allows you to combine multiple matrix type tweens into a single tween. You could, for example, create a spin tween that rotates the object, and a path tween that moves the object, and a polygon tween to distort the object. The values from all of these individual tweens would be combined to produce a single matrix that can then be used as the source value for any matrix property. Note that the order of the tweens here is significant and can affect the output.

The MultiMatrix Tween Editor window is of a different format than the other tweens. In addition to the usual general information at the top, including editable name, ID, Start Time and Edit Time fields, it has another editable field at the bottom. To the right is a “New Tween” button. When you click on this button a new, “untitled” tween will be placed in the MultiMatrix tween. Double click on this tween and you will be presented with another Tween Editor window similar to those used in other tweens. This window has a reduced number of tween types from which to choose. The presented tweens are those that can affect the Matrix of a visual object. This Tween Editor window operates exactly like the previous Tween Editor windows.

To remove a tween from a MultiMatrix tween, select the tween and press the delete key.

To reorder tweens from a MultiMatrix tween, select the tween and drag it to a new position in the list.

Tutorial

The best way to understand how a tween operates is to go through an example. We will start with some simple tweens you can use in everyday projects. The example we've chosen is a tween that will cause a sprite to spin in the movie. Follow the steps below:

Create a new project in LiveStage Professional. Be sure the Info Tab of the Project Window is visible. Check "Autoplay On Open" in the Playback section. Choose "Standard Movie" from the Controller popup. Checking these will play the movie immediately after you have created it, as well as to give you the opportunity of stepping through it.

Next click on the Tracks Tab in the Project Window. In the Tracks Menu choose Create/Sprite Track. A new Sprite Track called "Sprite 1" will appear. It will contain one sample called "untitled". The timeline will show a duration of one second.

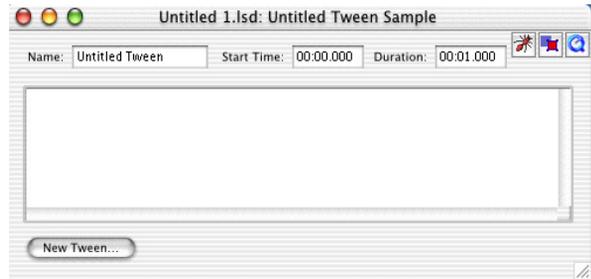
Click on the sample called "untitled". For easier reference we'll change the name of the sample to "Bird". Use the small edit box in the upper left of the Sample Window. You'll see the title of the window change to "Bird Sample".

Open the Library next by choosing "Show Library Window" from the Window Menu. Click the "Media" tab. Open the folder called "Images". If it's not visible, select "All Media" from the Display popup near the top of the Library Window. Next open the folder called "Animals". You will see a listing of various images of cartoon animals. Click on "Animals.pct.001". An image of a bird should appear in the image box in the Library Window. Next be sure that the Images Tab of "Bird Sample" has been selected, then drag the name (not the image itself) of the bird into the Images Tab area. The image name should appear in the left hand pane, and the image itself to the right. You can close the Library Window now to give yourself more screen space.

Now click the Sprites Tab in the Bird Sample Window. The sprite name column to the left will be empty. Now click on the “New Sprite” button in the lower left of this window. A new sprite, named “untitled” will be placed in the sprite column. Double click on it to get a window called “untitled Sprite”. Change its name to “Toucan” or any name you wish.

You can close all windows except the Project Window now, however, we will return to the “Toucan” sprite a little later.

In the Project Window choose “Create/Tween Track” from the Tracks Menu. You will see a new track appear in the Tracks Tab. It’s called “Tween 1” and has a single sample named “untitled”. Double

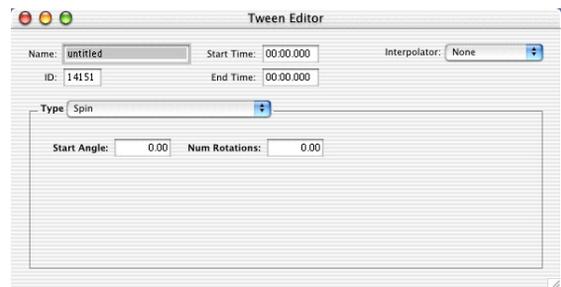


click on the sample and you will get a window titled “untitled Sample”. Change its name to “FirstTweenTrials” or anything else that seems appropriate.

Next, click the “New Tween” button in the lower left of the “FirstTweenTrials” window. A new, untitled entry will appear in the window. Double click it to show the Tween Editor window. There are numerous user inputs possible in the Tween Editor menu. It’s here that you will select and modify the many tween effects available in LiveStage Professional.

In this example we are going to work with one fairly simple effect: spinning a sprite on its registration point.

Click on the Type popup and select “Spin” from the options available. The Tween Editor window will change to reflect the options available for spinning a sprite.



You’ll have to make some modifications to the tween parameters next. First, change its name to “SpinThatToucan”.

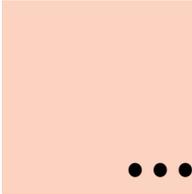
Next change the number in the “Num Rotations” edit field to “00:02.000”. You can now close the Tween Editor and the Tweens Sample windows. Next we have to make an edit change in the sprite. Double click on the Sprite Sample called “bird”. Find the properties section called “Matrix”. Click on “Source”. There will be two options “None” and “SpinThatToucan”. Select “SpinThatToucan”.

You are now ready to try out the effect you have defined. You can generate a trial QuickTime movie in three different ways: Select “Run Wired Movie” from the File Menu; Type Command R on the keyboard; or Click on the movie frame icon in the upper right corner.

A movie will now be generated. Since you chose “Autoplay on Open” earlier it should run immediately, spinning the bird twice. We didn’t modify the registration point on the image so that rotation will be in the upper left corner.

If you chose Standard Movie Controller you’ll be able to move the bird through its rotations with the mouse.

You may wish to experiment with various parameters now. Bring the Tween Editor up. Some of the things you can adjust include the start time and end time. You can also adjust the number of rotations and the angle of the sprite at start time.



Chapter 21

Modifier Tracks



Overview	215
Modifier Track Properties	215
Adding Sequences	216
Using Modifiers	217
Modifier Types	217

Overview

Modifier Tracks are one of the non-visual data generator tracks in LiveStage. The sole purpose of a Modifier Track is to generate numeric values over a period of time. Unlike Tween Tracks, which generate numeric values through an algorithm, Modifier Tracks only provide values from a sequence of numbers stored in the Modifier Track.

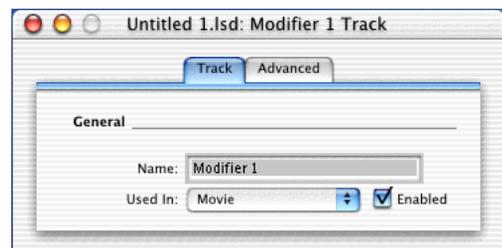
Modifier Tracks are used to supply discrete data patterns. For instance, to animate a sprite, a modifier track could be used to specify a pattern of images to be displayed over time. Alternately, a modifier track could be used to trigger events synchronized with the playback of the movie. Modifier tracks are an extremely efficient playback control mechanism.

The values generated by a Modifier Track can be used by other media tracks as data for their respective source popup menus. The usefulness of this may not be obvious at first, but there are capabilities that Modifier Tracks provide which would be difficult or impossible otherwise.

Lets say for example that you want to animate a sprite that contained three images. You can create a Modifier Track that provides the numeric values 1, 2, 3 at a rate of 5 frames per second, this can then be told to repeat 10 times. Now this Modifier Track can be used as the source for the sprite's Image Index. The result is that the sprite will display images 1, 2 and 3 in sequence at a rate of 5 FPS. Of course the sequence could be changed so that the sprite animates backwards by changing the values used in the Modifier Track to be 3, 2, 1. You can also add more complex series of values such as 3, 1, 3, 2, 2.

Modifier Track Properties

The properties for a Modifier Track are the same as the standard set of properties available in all tracks. To obtain more information on what



fps, repeat 2, sequence 1,2,3. Secondly, add another sequence which will be on the following line with a rate of 10 fps, repeat 1, sequence 4,5,6,7,8,9. Finally, add a final sequence in the last line with a rate of 10 fps, repeat 2, sequence 1,2,3. Now in the Sprite's properties tab, select this modifier from the source popup for image index. Now when you play this movie, the sprite will go through an animated sequence.

Using Modifiers

In order to use a Modifier Track you need to set the Source for the property that will receive its values from the Modifier Track. For example, if you were to use a "Index / Layer / Flag" Modifier Track to set the Image Index property for a sprite, you would go into the sprite properties and in the Image Index group specify that the source is to be my "new Modifier Track".

You should note that using modifiers to control sprite properties has some caveats and they are that modifiers can not be switched in/out at different times in the sprite, and once a Modifier is in control of a property, that property can not be changed with a script. However you can disable and then enable the modifier track. Also, a modifier track needs to be playing for its values to change.

Modifier Types

Event Trigger

These types of modifiers are used to generate events in sprites. Events are triggered when the values change, thus you need to intersperse the event numbers with 0, i.e. 1000, 0 at 5 fps.

Currently there is a bug in QuickTime 4.1 with Windows that does not convert the event number to a format that Windows understands. Use any of these event numbers if you run into this bug: 65792, 13154, 197376, 263168, 328960, 394752, 460544, 526336, 592128, 657920, 723712, 789504, 855296, 921088, 986880.

Index / Layer / Flag

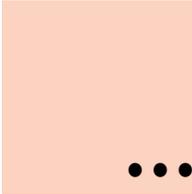
A whole number value which may be used in cases where an Index, Layer or Flag value is required.

VR Angle

Generates floating point numbers which can be used for the panning, tilt and zoom in QuickTime VR movies.

Volume / Balance

These are floating point values that are used for setting the volume and/or balance for an audio track. For Volume settings the range is from -256 to +256. Balance settings use -128 for full left balance and +128 for full right balance.



Chapter 22

Introduction to QScript



Overview	221	Comments	232
QScript Basics	221	Target Objects	232
Data	221	Cross-Platform Issues	235
Constants	222	Putting it all together	235
Variables	222	Events	235
Properties	226	QScript Actions	236
Statements	227	Continuous Lines	237
Assignment Statements	227	Line Continuations	237
Expressions	228		

Overview

LiveStage provides an extremely powerful user interface for the assembly and control of movie elements. However, the addition of scripting opens up a whole new dimension (i.e. interactivity) for LiveStage created wired movies. QScript is the scripting language used within LiveStage.

In short, scripting is a simplified form of programming. QScript within LiveStage allows elements of a wired movie to respond to events (i.e. mouse clicks, key-strokes, etc.) and change the behavior of the wired movie. Scripting provides the “wired” part of LiveStage authored wired movies.

QScript is a simple object-oriented scripting language created by Totally Hip Software. The syntax of QScript is very similar to many of the common programming languages such as JavaScript, Basic, Lingo for Director, and C++. People with prior experience with any of these programming languages should find QScript familiar and easy to learn.

QScript Basics

QScript is a simple scripting language that utilizes a number of standard programming elements. In this section, we will define and explore these elements.

A script is composed of one or more statements that are executed in sequence.

Scripts are composed of two basic components:

- Data
- Statements

Data

Data, when used within the context of a scripting language, refers to information stored in a form useful to a script. Data for a script is stored in either constant or variable form.

Constants

Constants, as the name implies, are data that can not be altered when the script is executed. Constants are also called “Literals”. The following are some examples of constants in QScript:

- 1234** Numeric constant indicating the whole number 1234
- 1234.56** Numeric constant indicating the real number 1234.56
- 1** Negative numeric constant indicating the whole number -1
- “Hi”** Text or String constant containing the characters “H” and “i”
- TRUE** Predefined Boolean constant for TRUE
- FALSE** Predefined Boolean constant for FALSE

Constants are used within QScript statements.

Variables

Variables are objects that store information while the script is running. Variables differ fundamentally from constants because variables have names and they have a value associated with them. In addition, the value stored in variables can be changed during the execution of statements. Variables can contain either numbers or strings and will be automatically converted by QuickTime to the type that it requires when needed. This means that a variable assigned the number 25 will be converted to the string “25” automatically by QuickTime when the variable is passed into a call that takes a string as a parameter.

Variable Declaration

Variables must be declared before they are used in a script. This tells LiveStage Professional the name and scope (see below) of the variable being declared. This requirement is similar to most programming languages and helps to reduce the number of errors that can be introduced into a script.

The following are some examples of variables in QScript:

```
GlobalVars Total //Declare a variable named "Total"  
LocalVars Count //Declare a variable named "Count"
```

Array Variables

Array variables are variables that can contain multiple values. The following example is a declaration of an array variable that contains 5 values:

```
GlobalVars ItemCounters [5]
```

The square brackets (“[” and “]”) enclose a numeric constant specifying the number of elements the array variable “ItemCounters” can contain. In the example above the variable “ItemCounters” has been declared to have five elements.

In the following example, values have been stored in each of the five elements of an array variable called “MyArray”.

```
GlobalVars MyArray [5]

MyArray [0] = 5
MyArray [1] = 7
MyArray [2] = 16
MyArray [3] = 4
MyArray [4] = 22
```

The different elements of array variables are accessed using an index. The index for these different values starts at 0. The following QScript statement accesses the fourth element of “MyArray” by specifying the numeric value “3” within the square brackets as the index to retrieve:

```
LocalVars CurrentIndex

CurrentIndex = MyArray [3]
```

After this script has executed the variable “CurrentIndex” will contain a value of 4.

You can use a “For” loop to access each value in an array like this:

```
LocalVars x

For x = 0 to 4
    MyArray [x] = 0
Next
```

You must take special care not to use array bounds that are outside of the range that you declared for the array. Doing so will overwrite the values stored in other variables and will lead to unpredictable program behavior.

You can also store strings in arrays like this:

```
SetString(MyArray[3], "this is a string")
```

Variable Scope

The term “Scope” refers to how persistent a variable is. There are four scopes for variables in LiveStage Professional:

- Movie
- Global
- Sprite
- Local

Movie Scope

If a variable is declared to have Movie Scope, then that variable can be accessed from any script within the wired movie. In addition, the value stored in a Movie variable is persistent while the movie is being played. Movie variables are used in places where persistent information needs to be stored (i.e. the total number of times a button has been clicked). The following is an example of a Movie Variable declaration:

```
MovieVars TimesButtonClicked
```

One very interesting application of Movie Variables is their use in inter-movie communication. Movie Variables can be declared using ID's which may in turn be used in the SetVariable and GetVariable QScript statements. Using these ID's Movie Variables may be addressed directly by other movies.

To declare a Movie Variable with an ID you simply add the ID you want to the end of the variable declaration, separating it from the name by a colon. See the example below:

```
MovieVars TimesButtonClicked : 5
```

This example creates a Movie Variable called TimesButtonClicked that is stored at position 5 in the list of movie variables. To access this variable in another movie you simply declare it as we did above in the second movie and then either use the variable by name or through the SetVariable and GetVariable QScript statements.

You can use the addresses from 1 - 9999 for this purpose. All other movie variables will be stored using an address above 10000.

Flash, Text and QTVR scripts must use MovieVars for storage. No other variable scope is accessible by these tracks.

Global Scope

If a variable is declared to have Global Scope, then that variable can be accessed from any script within the current Sprite Track in the wired movie. In addition, the value stored in a Global Variable is persistent over the duration of the Sprite Track. Global Variables are used in places where persistent information needs to be stored (i.e. the total number of times a button has been clicked). The following is an example of a Global Variable declaration:

```
GlobalVars TimesButtonClicked
```

Sprite Scope

A variable declared to have Sprite Scope is available only to scripts belonging to the same sprite. This means that a value stored in a Sprite Variable can not be accessed by other sprites. The following is an example of a Sprite Variable declaration:

```
SpriteVars LoopCounter
```

Local Scope

A variable declared to have Local Scope is available only during the execution of the current script in the movie. This means that a value stored in a Local Variable in one invocation of a script will not be persistent the next time the script is invoked. Local Variables are used in places where persistence is not required. Local Variables are generally used for storing intermediate information during calculations within a script. The following is an example of a Local Variable declaration:

```
LocalVars LoopCounter
```


Statements

Statements in QScript are instructions for what to do with the data elements within a wired movie. Statements direct the wired movie to perform some action. Statements fall into the following categories:

- Assignment Statements
- Control Statements
- Actions
- Comments

Assignment Statements

Assignment statements work in conjunction with constants, variables and properties to perform data processing. Assignment statements are made up of two parts, a left-hand-side and a right-hand-side. The following shows the basic format of an assignment statement:

```
Some_Variable = Expression
```

The left-hand-side (`Some_Variable`) of the assignment statement refers to a target variable. The right-hand-side (`Expression`) of the assignment statement is a numeric or algebraic expression whose value is evaluated and assigned to the target variable (left-hand-side). The two sides are connected with the “=” equal sign.

The following are valid assignment statements:

```
GlobalVarsTotal, CurrentImage  
LocalVarscount
```

```
count = count + 1           // Increments the local  
                             // variable count  
Total = Total + count      // Adds the value of the  
                             // localvariable "count"  
                             // to the global variable  
                             // Total  
CurrentImage = ThisSprite.ImageIndex  
                             // Assign the current  
                             // sprite's image index to
```

```
// the global variable
// CurrentImage
```

Expressions

Any of the following operators may be used in the evaluation of the right-hand-side expression:

+	Addition
-	Subtraction
*	Multiplication
/ or ÷	Division
DIV	Integer division
REM	Remainder after division
OR	Binary OR
AND	Binary AND

Parenthesis, “(” and “)”, may be used to override the default order of evaluation for an expression:

```
1 + 2 * 3 - 4           the result is 3
(1 + 2) * (3 - 4)      the result is -3
```

Strings cannot be assigned or compared in the same way as numbers. If you use a string variable in a numerical expression, then it will be automatically converted to a number by QuickTime. To assign a value to a string variable use the statement “SetString(myStringVar, “the new string value”)”. You can also join two strings together and create a third string using “AppendString(startString, endString, resultString)”.

Boolean Expressions

Boolean expressions are similar to Assignment statement expressions (see Assignment Statements). In addition to algebraic operators used in assignment expressions, Boolean expressions also use the comparison operators:

<	Left side is less than the right side
>	Left side is greater than the right side
<= or ≤	Left side is less than or equal to the right side
>= or ≥	Left side is greater than or equal to the right side
!= or ≠	Left side is not equal to the right side

Another difference is that there is no left-hand-side to Boolean expressions. The evaluated result of Boolean expressions is not assigned to a variable (as in assignment expressions) but is used to determine if a condition has been met.

Boolean expressions always evaluate to TRUE or FALSE. Several conditional expressions may be joined together using the logical operators “AND” and “OR” to create complex conditional expressions. Control Statements

Control Statements

Control statements add decision-making capability to scripts. Control statements fall into three groups:

- Conditional Statements
- Switch Statements
- Loop Statements

Conditional Statements

Conditional statements use expressions to determine if a part of the script should be executed. The following are the formats of the various conditional statements:

```
IF (boolean_expression)
    // script statements executed when "boolean" is TRUE
ENDIF
```

```
IF (boolean_expression)
    // script statements executed when TRUE
ELSE
    // script statements executed when FALSE
ENDIF
```

```
IF (boolean_expression_1)
    // script statements executed when TRUE
ELSEIF (boolean_expression_2)
    // script statements executed when TRUE
ELSE
```

```

        // statements executed when neither expression is
        TRUE
    ENDIF

```

Switch Statements

Switch statements provide multi-way conditional execution of scripts similar to conditional Statements. The following is the structure of a Switch Statement:

```

SWITCH variable
    CASE constant
        // statements executed if the variable = constant
    ENDCASE
    CASE constant
        // statements executed if the variable = constant
    ENDCASE
ENDSWITCH

```

The variable supplies a value that is matched to the constant in the CASE statement. If the value stored in the variable matches the constant in a CASE statement, the statements of that CASE statement are executed. At most only one of the CASE statements will match the variable and possibly none will match.

Loop Statements

Loop statements allow a sequence of statements to be executed a number of times. Following are the formats of loop statements:

```

WHILE (boolean_expression)
    // script statements executed while "expression"
    // is TRUE
ENDWHILE
FOR variable = expression_from TO expression_to
    // script statements executed
NEXT

```

In the first format above (WHILE loop), “expression” is a Boolean expression (see Boolean Expressions). The script statements within the Loop statement structure are executed while the “expression” is evaluated to be TRUE.

In the second loop statement format (FOR loop), “expression_from” and “expression_to” are numeric expressions (expressions with a numeric result) which govern the number of times the FOR loop is executed and “variable” is a variable storing the current state of the FOR loop. The value stored in “variable” is used to determine if the FOR loop is to continue. The “expression_from” specifies the starting value that will be set into “variable” when the FOR loop is initiated. The value in “variable” is incremented by 1 every time the FOR loop “loops” until the value exceeds the value of “expression_to”. When the value in “variable” exceeds the value of “expression_to”, the FOR loop terminates and execution continues with the script statements following the NEXT statement in the script.

The following script executes the script statements in the loop statement 5 times:

```
LocalVars count
count = 0
WHILE (count < 5)
    count = count + 1
ENDWHILE
```

***Note:** Loop statements must be used with care as the only thing which determines how many times the loop executes is the evaluation of the loop control expression. If this expression never evaluates to FALSE (or in the case of the FOR loop, the variable never exceeds the value of the termination expression), the loop will continue to execute forever. This is known as an Infinite Loop and is a common programming error.*

Actions

Action statements are used to issue commands in QScript. Refer to the Appendix I PDF documentation for the lists of available Actions in the QScript language.

Comments

Comments allow the inclusion of text within a script that is ignored by the QScript compiler. This allows the inner workings of a script to be documented in detail. It is recommended that comments be used to document the workings of scripts. Undocumented scripts quickly begin to lose their meaning the longer you leave them undocumented. The following is an example of a comment:

```
// This is a comment
```

Any text following the double forward slash characters “//” up to the end of the line is ignored by the QScript compiler. Comments may begin at any point in a line of script.

```
LocalVarscount// declare a local counter variable
```

In the preceding example, all text following “//” is ignored. Comments are also useful for temporarily disabling script statements. By placing the “//” in front of a line of script, the line is ignored by the QScript compiler. “/*” and “*/” are used for making multiline comments. All text between the “/*” and the “*/” is ignored.

Target Objects

When working with actions in a script, a target object must be specified. For instance, to hide or show a sprite, you should use the following script statement:

```
ThisSprite.SetVisible(FALSE) // Hides the current  
                             // sprite  
ThisSprite.SetVisible(TRUE)  // Shows the current  
                             // sprite
```

In the above statements, the target object is specified by the keyword “ThisSprite” in front of the period “.”. The following is the format for specifying target objects:

```
<Target_Specifiers>.<QScript_Action>
```

The Target Specifiers is one of the following:

ThisMovie	Specifies the current movie (may be omitted),
ThisTrack	Specifies the current track (may be omitted).
ThisSprite	Specifies the current sprite (may be omitted). You can not use ThisSprite in a Frame Loaded event handler since these handlers do not actually belong to any Sprites. You should use SpriteOfID(\$ThisSpriteID) instead.
MovieNamed("Movie Name")	Specifies the movie by supplying the name of the movie.
MovieOfID(Movie_ID)	Specifies the movie by supplying the ID of the movie.
TrackNamed("Track Name")	Specifies the track by supplying the name of the track.
TrackOfType(Track_Type [, Index])	Specifies the track by supplying the type of the track. Index is optional.
TrackOfIndex(Track_Index)	Specifies the track by supplying the index order of the track.
TrackOfID(Track_ID)	Specifies the track by supplying the ID of the track.
SampleNamed("SampleName")	Specified a sample within a track. Compute time target.
SpriteNamed("Sprite Name")	Specifies a sprite by supplying the name of the sprite.
SpriteOfIndex(Sprite_Index)	Specifies a sprite by supplying the index order of the sprite.
SpriteOfID(Sprite_ID)	Specifies a sprite by supplying the ID of the sprite.

ChildMovieNamed(“ChildName”)	Specifies a child movie by supplying the name of the child movie.
ChildMovieOfID(Movie_ID)	Specifies a child movie by supplying the ID of the child movie.
PreviewMovie	Specifies a target movie in the front most position. This only works in LiveStage and is used to make Tool movies. This will allow you to inspect other movies for information.
ThisEvent	Specifies the local data stored in the current event. This is used to access certain event data structures such as QTLists.

In the above list, Target Specifier with the prefix of “This” can be omitted since Movie, Track and Sprite actions (see QScript Actions below), by default, use the current movie, track and sprite respectively as the target object if it isn’t explicitly specified. For example, the following statements perform the same actions when they refer to the current sprite:

```
SetVisible (FALSE)
ThisSprite.SetVisible (FALSE)
```

By supplying a Target Specifier, a script statement can target any object in any Movie.

There are special defines created for each sprite that you can use in your scripts. Each image name will have a define created that refers to the image index. This means you can use \$”My Image Name” instead of the image index wherever you need to refer to the image index. Each custom event handler name will have a define created that refers to the custom event handler’s ID. This means you can use \$”My Custom Event Handler Name” instead of the custom event handler ID whenever you need to refer to a particular custom event handler. The ID of the Sprite that contains this script is defined as “ThisSpriteID” so you can refer to the current sprite using `SpriteOfID ($ThisSpriteID)`. This becomes useful in Frame Loaded events since you can not use ThisSprite in them.

Cross-Platform Issues

If you are planning to do work on both the Mac and the Windows operating systems, you should be careful how you write your QScript. There are a few symbols which are understood on the Mac Environment but not on the Window Environment.

This issues occur when using the operators and mathematical constants. In the earlier section, there were two methods introduced to represent these operators and mathematical constants. One is using a single character and the other is using two characters. Both methods are valid in the Macintosh Operation System but only the two character representation is valid on the Window Platform. You can refer to the QScript Reference window in the application for any others.

The list below are a few operators and constants which has two display methods:

<code><=</code> or <code>≤</code>	Left side is less than or equal to the right side
<code>>=</code> or <code>≥</code>	Left side is greater than or equal to the right side
<code>!=</code> or <code>≠</code>	Left side is not equal to the right side
<code>/</code> or <code>÷</code>	Division
<code>PI</code> or <code>π</code>	Constant for Pi

In order to have the script move between both platforms, it is suggested that you use the two character symbols instead of the single character symbol.

Putting it all together

Now that we have covered the basics of QScript, we will examine how scripts are executed when a wired movie is played.

Events

Events are triggers that cause a wired movie to execute its scripts. Following is a list of some of the activities that can trigger an event in a wired movie:

- Mouse entering a sprite
- Mouse exiting a sprite
- Mouse click
- Time passing
- Mouse Moved
- Key Pressed

At this time we should point out one item when using scripts in Event Handlers. While the script in an Event Handler is being executed the movie will not be visually updated. This is an important item to be aware of because it means that if you use a loop to change the visual appearance of a sprite you will only see the end result of those changes, not the individual steps. This effect is very noticeable when you try to use a loop to animate a sprite.

A simple solution to this problems is to use the Idle event to increment through a sequence of images. Of course if you prefer to leave the scripting to someone else you can use the animation behaviour which does all the work for you.

Please refer to “Event Handlers” on page 85 from the Scriptable Tracks chapter for information on the list of Event Handlers.

In addition to the Standard Event Handlers, Custom Event Handlers can be created for Sprites. Custom Event Handlers provide the ability to modularize your Sprite scripts. Custom Events are triggered by the QScript action `ExecuteEvent` (see QScript Reference).

QScript Actions

Actions provide the means for scripts to control the objects in a movie. In the following example, the Action is “Rotate”:

```
ThisSprite.Rotate(90)
SpriteNamed("SpriteName").Rotate(90)
```

The Rotate action causes the current sprite to be rotated 90 degrees clock-wise. The number 90 within the parentheses is a constant specifying the parameter of the action.

The following are examples of actions:

```
ThisSprite.ToggleVisible// no parameters
ThisSprite.MoveTo(0, 0)// 2 parameters

SpriteNamed("SpriteName").ToggleVisible
SpriteNamed("SpriteName").MoveTo(0,0)
```

For additional information on actions, please see the QScript Reference in the User’s Manual.

Continuous Lines

If you have a need to put more than one QScript statement on a line you can use a “;” character as a line terminator. This allows you to have lines like the following:

```
SpriteVars test; test = 0
```

Without the continuation character you would have to write the above QScript like this:

```
SpriteVars test
test = 0
```

Line Continuations

There may be times when you need to enter a long string of QScript commands. Normally this would create a long line in your script editor which would require you to scroll the editor’s view just to see the complete line. QScript uses the \ as a line continuation character. This allows you to break a line of QScript commands into multiple lines. Note that you must put the line continuation in logical locations. It would not be wise to put a line continuation character in the middle of a keyword.

```
MovieName("My long movie name").\
    TrackNamed("My Long track name").\
    SpriteNamed("My Long Sprite Name").\
    ExecuteEvent($"My Long Event Name")
```

Special Characters in Strings

You can use the following special character sequences in a string to enter special values into the string:

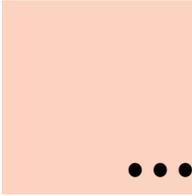
- \r Enter a carriage return character indicating a new line.
- \n Enter a new line character indicating a new line.
- \t Enter a tab character.
- \f Enter a form feed character to indicate the start of a new page.

You can also use just a `\` at the end of a line to continue the string on the next line like this:

```
"this is a very long string that goes on to the next line\  
and ends here"
```

You can also include a line break in the string like this:

```
"this is line one of this string  
and this is line two"
```



Chapter 23 Behaviors



Overview	241	Description Section	249
Using Behaviors	241	Parameters Section	250
Parameters	242	Event Handlers	251
Pre-Defined Behaviors	242	Some Guidelines	252
Authoring Behaviors	247	Variables	252
Sections	249	Parameter Passing	253
Name Section	249	Custom Events	254

Overview

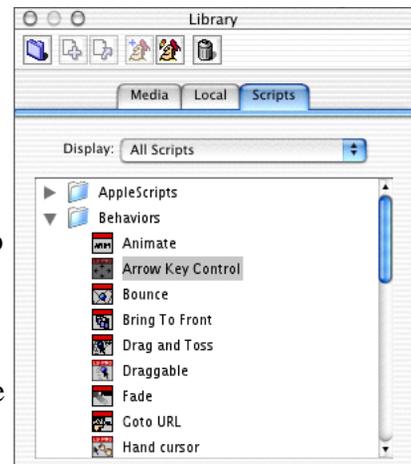
Behaviors are a powerful addition to LiveStage Professional. They allow you to create rich interactive movies with little or no scripting. Behaviors are pre-made scripts that you use by dragging and dropping them into a sprite. A sprite can have one or more behaviors, thus giving you a great deal of capability without ever writing a line of QScript.

LiveStage comes with a library of standard behaviors you can use in your own projects. If you want specialized behaviors you can create your own which will operate in the same manner as the standard behaviors.

Using Behaviors

Behaviors are added to sprites through the Behaviors tab in a Sprite sample. Traditionally the behavior objects are located in the Scripts tab of the Library window inside the Behaviors folder.

Behaviors are made to be used alone or in groups. They are designed to do one simple task and do it well. For example, a behavior could be written that enables a sprite to be draggable, changes the mouse to a hand cursor and allows you to position the sprite with arrow keys. This would be a bad way to make a behavior. Instead we have made a Draggable, Hand Cursor and an Arrow Key Control behavior. You can then add any combination of these behaviors to your sprite



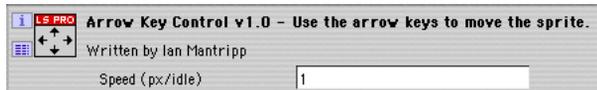
until the sprite acts the way you want it to. This design strategy allows Totally Hip Software and other developers to add new behaviors that work with existing behaviors to increase functionality.

In a standard installation there will be many behaviors in the Behaviors folder. Some of these include the Sway, Slide To and Drag & Toss behaviors.

To use a behavior, simply drag and drop it from the Library window into the Behavior tab of the sprite editor. Once you have done this you can view a description of the behavior by clicking on the info button  located to the left of the behavior icon. If this is not a behavior you want then press delete to remove it. You can click on the parameters icon  to the lower left of the behavior icon to set any options for the behavior. If the behavior takes no parameters, this icon will not be shown.

Parameters

When you click on the parameters icon, you will see a list of all of the parameters



for the behavior and their current settings. You can enter both text and numbers in the parameter fields depending on what the behavior wants for input. You can also enter defines as parameters in the same way you refer to defines in your scripts by preceding the define name with a "\$". For example, a behavior may call a custom event when certain conditions are met. You can enter the ID of your custom event or enter its name which is pre-defined to be its ID like this: \$ "My custom event". You can do the same with image names.

Pre-Defined Behaviors

The following descriptions provide a basic introduction to the capabilities of each behavior, including specific parameters for the behavior, if needed. These behaviors were either created by the LiveStage team members or dedicated LiveStage developers. See the particular behavior in LiveStage Pro for specific details.

There are Automatic behaviors and behaviors that require specific events to trigger them. Automatic behaviors mostly function on their own once they are activated within a QuickTime movie.

Arrow Key Control

When each of the arrow keys is held down, the sprite will move in that direction. The Speed at which the sprite will be moved is based on the number of pixels entered in the Speed parameter. During each time through the Idle event, the sprite will move by the number of pixels specified if the particular arrow key is held down. More than one arrow key can be held down at a time. This sprite will not move past the edges of the sprite track.

Bounce

The sprite will start moving in a random direction and will bounce when it hits the edges of an imaginary box specified in the parameters. When the ball bounces, the custom event will be called. You can play a sound in your custom event for example.

Bring To Front

When the mouse button is pressed on the sprite, the sprite will be moved forward by 1000 layers and put back when the button is released. You can use this with the Draggable behavior so that the sprite stays in front of all other sprites while it is being dragged.

Drag and Toss

Allows a sprite to be dragged and when released, the sprite continues to move in the direction it was dragged. No checks are made to see if the sprite goes off the edge of the screen.

Draggable

Allows a sprite to be dragged around. Once released, the sprite will stay where it is dropped. Use the Bring to Front and Hand cursor behaviors to enhance the dragging if desired.

Fade

Makes a sprite gradually appear or disappear. You can set the fade speed anywhere from 1 to 20, with 1 being the slowest and 20 the fastest. The Fade behavior must be activated by calling either the "\$"Fade In" or "\$"Fade Out" custom handler from your script. You can set up your sprite to fade in by putting a call to the fade-in custom handler in your frame loaded event like this:

```
SpriteOfID("$ThisSpriteID").ExecuteEvent("$"Fade In")
```

In the following example the sprite will fade out when clicked on.

```
[Mouse Click Handler]
ExecuteEvent("$"Fade Out")
```

Goto URL

This behavior will cause the user to go to a specified URL (universal resource locator on the Internet) when the sprite is clicked. Enter the URL in the parameter.

Example URL string: "http://www.totallyhip.com"

Hand cursor

When the mouse is over the sprite the cursor will change to the Open Hand cursor. When the mouse button is pressed while over the sprite the cursor will be changed to the Closed Hand. When the mouse button is released over the sprite the cursor will be changed to the Open Hand cursor. Use this behavior with the Draggable behavior to enhance its effect.

Keyboard Control

This is the same as the Arrow Key behavior except that you can specify what keys control the sprite. When each of the specified keys is held down, the sprite will move in that direction. The Speed at which the sprite will be moved is based on the number of pixels entered in the Speed parameter. During each time through the Idle event, the sprite will move by the number of pixels specified if the particular key is held down. More than one key can be held down at a time. This sprite will not move past the edges of the sprite track. Enter each direction key in the parameters.

Mouse Magnet

This behavior will cause the sprite to vibrate and tend to move towards the mouse. Enter the amount of vibration in the Wobble Factor parameter. A low Wobble Factor will cause the sprite to move more directly towards the mouse. The sprite will be kept within the bounds of the sprite track.

Rotate Continuous

This behavior will cause the sprite to rotate continuously. This behavior has two parameters, Rotate Delay and Rotate Delta. The Rotate Delay can be set from 1 to 10, with 1 being the slowest and 10 being the fastest. The Rotate Delta can be set from 1 to 20, indicating the number of degrees per step in the rotation.

Rotate To

This behavior causes the sprite to rotate to a specified angle when the "\$"Start Rotation" event is called. This behavior has three parameters, Rotate Delay, Rotate Delta, Rotate To Angle. The Rotate Delay can be set from 1 to 10, with 1 being the fastest and 10 being the slowest. The Rotate Delta can be set from 1 to 20, with the parameter being the number of degrees to rotate per step. The Rotate To Angle can be set from between -360 degrees to 360 degrees to tell the sprite what angle to stop at. You start the sprite rotating by calling the "\$"Start Rotation" event. You can reverse the rotation by calling the "\$"Reverse Rotation" event. You can reset the rotation by calling the "\$"Reset Rotation" event.

Scale

This behavior causes the sprite to change size when the mouse is over it. Specify the horizontal and vertical scale factors in the parameters. Entering 2.0 will cause the sprite to double in size.

Selector and Selector Controller

These behaviors must be used together on a group of sprites. They are used to create a user interface that consists of a series of buttons where only one button can be selected at a time. Many of the sample movies that demonstrate various features of wired sprites use these behaviors to control the user interface. You can examine these movies to see how these behaviors work. These behaviors are here for you to examine but are not intended for general use.

Slide To

This behavior causes the sprite to slide in from a specified starting x, y coordinate to an ending x, y coordinate. This behavior has six parameters. Use the Slide Delay and Slide Steps to control the speed and smoothness of the slide. The Slide delay has parameters from 1 to 10 and the number of Slide Steps can be set from 1 to 20. Supply the start and end positions by entering the start x and y coordinates and the end x and y coordinates.

Add the following calls to any sprite script:

Slide In is activated by this: `ExecuteEvent ($ "Slide In")`

Slide Out is activated by this: `ExecuteEvent ($ "Slide Out")`

To cancel the slide and reset to the start use this: `ExecuteEvent ($Reset Slide")`

Slider

This behavior can be used to implement a slider user interface control. Attach this behavior to the thumb control of the slider and enter the position of the left and right edge of the area it can slide in. Set the Min and Max values that the slider represents. Enter the name or ID of your custom event handler to be called when the slider is moved and it will be called whenever the slider changes position. The Sprite variable "Slider_CurValue" will contain the current value of the slider which will be a number from Min Value to Max Value.

Snap Up

This behavior causes the sprite to snap to a given x and y coordinate with a given tolerance when the mouse is released. Use this with the Draggable behavior. Enter the x and y coordinates that you want the sprite to snap to. Enter the snap tolerance. This will be how close the sprite needs to be to the coordinates before it will snap into place. If the sprite is within the tolerance when the mouse button is released, then it will be snapped to exactly the coordinates specified and the custom event specified will be called, otherwise it will not be snapped and the custom event for no snap will be called.

Sprite Track Mouse

This behavior causes the sprite to automatically move to the x and y coordinate location of the mouse pointer if the Start Track Mouse event is called and stops when the Stop Track Mouse event is called.

Sway

This behavior causes the sprite to sway or rotate from side to side. This behavior has four parameters. Use the Rotate Delay and Rotate Delta parameters to control the speed of the sprite sway. A Decay can be set from 0 to 10, with zero being no decay and 10 being the quickest decay. If there is a decay then the sprite will sway less and less until it stops moving. Call the custom event "\$Start Swaying" to start the sway motion.

Track Mouse Toggle

This behavior causes the sprite to follow the mouse after it is clicked on and then stop after it is clicked on again. This is similar to the Draggable behavior except you do not have to hold the mouse button down while dragging. This behavior will also set the cursor to a hand that closes while the sprite is being dragged.

Tween Motion v1.1

This behavior causes the sprite to automatically move from a specified starting x, y coordinate to an ending x, y coordinate when the custom event handler "\$Start Tween Motion" is called. You also need to specify the number of steps to take to move the sprite as well as the custom event and sprite ID to be called when the motion is completed.

Vibrator

This behavior causes the sprite to wobble or vibrate randomly. Enter the x wiggle amount from 0 to 10, with 0 being the least amount and 10 being the most wiggle. Enter the y wiggle amount from 0 to 10, with 0 being the least amount and 10 being the most wiggle.

Authoring Behaviors

If you are already familiar with writing QScripts you will find creating your own behaviors to be a snap. Behaviors are really just QScripts that are packaged into one file and separated into multiple sections. Each event handler is contained in its own section and there are custom sections for the name, description and parameters for the behavior.

Opening a behavior file in a text editor will reveal scripts that look very familiar to those you are using in your sprites. The primary difference between a behavior's script and that of an event handler's script is that all the scripts for each event handler are included in one file.

Take the following behavior as an example:

[Name]

MyBehaviour v1.0 - Behaviour for doing some cool things
by Jack Smith, jsmith@somewhere.com

[Description]

This behaviour is a collection of cool routines

Revision History:

v1.0 - Release version published (1/10/2000)

v0.9 - Beta version published (1/1/2000)

Usage Instruction:

Simply attach this behaviour to a sprite. The behaviour
will do the rest.

Parameters:

This behaviour does not use any Static Parameters

Custom Events:

ExecuteEvent("\$MyBehaviour_Init") : This event
initializes the behaviour

ExecuteEvent("\$MyBehaviour_DoCoolThings") : This event
modifies the sprite in a cool way.

Sections

The behavior text file is broken into several sections. Each section starts with a line of text that begins with some text enclosed in [] (square brackets). The information contained within the square brackets defines what type of section this is. There are four main types of sections: Name, Description, Parameters, and standard event handlers. You can add a comment on the same line as the section heading if you wish.

Name Section

The name section has the word “Name” between the square brackets like this:

```
[Name]
```

In this section you can provide a brief description of the behaviour as well as the author name, contact information and version number of the behaviour. When you are writing this section be sure you keep the length of each line to around 50 characters as this will display best in the Behavior tab.

Line lengths greater than this will still work but the user may need to re-size the window to view all of the text.

```
[Name]
```

```
MyBehaviour v1.0 - Behaviour for doing some cool things  
by Jack Smith, jsmith@somewhere.com
```

Description Section

The description section is used to provide more detailed information about the behavior. This section is only displayed when the user clicks on the information icon  shown when looking at the behaviors that have been added to the sprite.

Generally the Description section provides a detailed description of the behaviour. This information is provided as the documentation for the behaviour and should provide details of how the behaviour should be used:

Description Synopsis of the functionality of the behaviour.

Revision History A list of the versions, dates and revision notes of each version of the behaviour.

Usage Instructions	Describe how the behaviour should be attached to a sprite in order for it to function properly.
Parameters	Describe each of the static parameters of the behaviour detailing their use.
Custom Event	List of the custom events provided by the behaviour. Each listing must provide details describing the event, the parameters (if any) used by the event as well as any return variables or error code.

Parameters Section

This section is used to list the parameters for the behavior. The parameters allow users to customize the behavior to suit their needs. The parameters will be put into a define statement when the movie is compiled and can be used anywhere in the behavior scripts.

Each parameter is listed on a separate line and follows this basic form:

```
Name (value range), Define, Default value, Minimum
value, Maximum value
```

The Name component is a textual label that the user will see when viewing the parameters for the behavior. The value range is simply a part of the name and gives the user an indication of what values are expected. You should always include this in order to make it as easy as possible for the user to understand your behavior.

The Define component specifies what name will be used to store the value for this parameter. You need to ensure that this name is unique to your behaviour since this define will be global to the sprite and could conflict with other behaviors the user may add to the sprite. You can refer to this define in your behavior script just as you would any other define by preceding it with a “\$”.

Totally Hip Software has reserved all names starting with “THW_” for variable and define names.

The next three components are optional but they must appear in order. So if you want to include a minimum value, then you must include the default value first.

The Default value is either a numerical or textual value that will be used if the user does not enter a value. You should always include a default value.

The Minimum and Maximum values are numerical values that indicate the range of valid values that the user can enter.

For Behaviors that use parameters we recommend that you put the parameter limits within parenthesis. For example, in the Fade Behaviour there is a fade speed parameter, the parameter name is as follows; “Fade Speed (1-20)”. This tells the user of the Behavior that there is a reasonable range of values for the fade speed.

Event Handlers

Each event handler has its own section which contains the QScript that will execute in that handler. The scripts in these sections are no different than a traditional script you would find in a sprite. There is only one constraint that you need to be aware of: you should only use SpriteVars or LocalVars when declaring variables. There may be isolated circumstances where GlobalVars may be needed but these will traditionally be few and far between. For instance two or more behaviors could share the same global variables in order to communicate with each other. Keep in mind that the user can add your behavior to more than one sprite if you are using global variables. SpriteVars keep the variables localized to the sprite and thus ensures that you are not conflicting with other variables being used in the movie.

An event handler is defined by using the square braces to contain the name of the event. For example, if you were going to define a script that would execute during the frame loaded event, you would define it by using the following syntax: [Frame Loaded]

Any lines following this section heading would be executed as code in the frame loaded event handler. You will need to use the pre-defined define “\$ThisSpriteID” to refer to the current sprite in the Frame Loaded handler. Here is the Frame Loaded handler from our example behavior.

```
[Frame Loaded]
SpriteVars THW_rotateTimer
THW_rotateTimer = 0
```

In this example we are declaring a variable and then initializing it to zero in the frame loaded event.

Each event handler has a specific name you use when identifying it. Here is a list of the standard event handlers in LiveStage Professional:

```
Mouse Down
Mouse Up
Mouse Click
Mouse Enter
Mouse Exit
Mouse Moved
Frame Loaded
Key Pressed
Idle
List Received
Request to Modify Movie
```

You may also include custom events which are labelled using the id of custom event handler followed by the textual name you want to use for the custom event handler. An example would be [123007 Fade In]. For this example the custom event handler id is 123007 and its name is Fade In. The textual name will be automatically defined for you so you can refer to it in your scripts as: "\$Fade In"

Some Guidelines

The following are a few guidelines for the usage of variables within behaviors. These guidelines are intended to prevent conflicts with other behaviors. Following these guidelines will minimize any potential conflict your behaviour may have with other behaviors published by Totally Hip software or other Behaviour Authors.

Variables

All variable and define names you use in your Behaviors are shared. If you define a Local or Sprite variable, it will be visible to all scripts for that handler and that sprite. Also during the compile process, all behavior scripts for a sprite are collected into one script and then compiled. With this in mind you can see how conflicts can occur.

When defining variables in your behaviors you should always use `SpriteVars` instead of `GlobalVars`. `SpriteVars` keeps the variables localized to a sprite, if you use `GlobalVars` that variable will be shared among all instances of the behavior in each sprite. You may find this useful for advanced behaviors.

Naming Conventions

Since names are shared you must use unique names for `Movie`, `Global` and `Sprite` scope variables. For example:

```
GlobalVars THW_MyBehaviour_Counter
```

The above statement declares a `Global` variable for use in the behaviour. The “`THW_`” prefix identifies the author (i.e. `Totally Hip Software`). The next section “`MyBehaviour_`” identifies the name of this behaviour. The last section of the variable name identifies the actual variable name.

This method of variable naming minimizes the chance of variable conflict with other published behaviours.

Scope Identification

Use a single letter prefix to identify the scope of the variable. This improves the readability of the script. For example:

```
MovieVars mTHW_MyBehaviour_Counter  
GlobalVars gTHW_MyBehaviour_Value  
SpriteVars sTHW_MyBehaviour_Status
```

The above examples use the lower-case letters “`m`”, “`g`” and “`s`” to indicate `Movie`, `Global` and `Sprite` variables respectively.

Parameter Passing

`Global` variables must be used in order to pass parameters to custom events. These parameter variables should follow the `Variable Guidelines` stated in the above `Variable` section. The following are parameter passing guidelines which behaviour scripts should follow when receiving and returning values.

Cache value locally

Values passed into a custom event should be stored locally. This ensures that the parameter variable is not altered during the execution of the custom event.

Return variables

If the custom event returns a value, define a global variable for this purpose:

```
GlobalVars gTHW_MyBehaviour_Return
```

Error codes

If you want to provide an error code to indicate success or failure of the custom event, define a global variable for this purpose:

```
GlobalVars gTHW_MyBehaviour_ErrorCode
```

Set the variable to the appropriate state before exiting the custom event. For example:

```
0 = Success  
1 = General Failure  
etc...
```

Use the value 0 to indicate success. Any value other than 0 would indicate a failure. The non-zero value will indicate the error code.

Custom Events

Custom Events provide a way for the functionality of the behaviour to be executed.

Event IDs

Any custom event ids you use are also shared. Totally Hip Software has reserved custom event ids from 100000 to 999999. All event ID's need to be unique, because of this Totally Hip Software has a range of ID's that are issued (on request) to Behavior writers who are intending to write behaviors for use by the general public.

Contact Totally Hip Software to obtain a range of Event IDs that you can use for your own behaviours. Totally Hip Software maintains an order to minimize ID conflicts between published behaviours. For Event ID allocation please send e-mail to livestage@totallyhip.com.

Get/Set Accessor Events

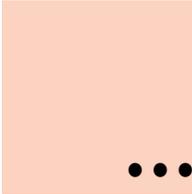
Provide Get/Set accessor events for accessing behaviour state variables. This method allows the state of the behaviour to be changed at run-time without the restrictions imposed by the use of Behaviour Parameters.

Initialize Event

Provide an Initialize event for initializing the behaviour (if necessary). Call this event from the Frame Loaded event.

Use of Standard Events

Place script that you want to execute within standard event handlers (i.e. Idle, Mouse Enter, Mouse Frame Loaded Exit, etc...) into custom events. Call these custom events from the standard event handlers. This method isolates the behaviour from any variable declaration conflicts that may arise when the same standard event handler is being used by the sprite and other behaviours.



Chapter 24 XML



What is XML?	259	The Basic Movie Script	263
What does XML look like?	259	Referring to Movie Tracks in XML	264
How does LiveStage utilize XML?	260	Common Sub-Element Groups	264
What are the uses of the LiveStage XML Project Files?	260	Track Sample Sub Element Group	265
LiveStage XML Project Files	263	Track Information	265
LiveStage Professional XML Project File Rules	263		

What is XML?

XML (Extensible Markup Language) is a popular format for defining and storing documents today. XML, like HTML, is a meta-markup language that is useful for storing documents in a textual form that is computer platform independent. Unlike HTML, XML is a completely customizable. Thus it is left up to the creators of a particular document format to supply meaning to its contents.

What does XML look like?

XML files are text files. All the data appearing within an XML file is represented as text characters. This makes XML editable using a simple text editor.

XML is also a “tagged” format. This means that elements within an XML file are defined with Start (`<item>`) and End (`</item>`) tags. Here’s an example of an element of an XML file:

```
<some_Item>This is the content.</some_Item>
```

In the above example, the element named “some_Item” begins with the Start tag “`<some_Item>`” and ends with the End tag “`</some_Item>`”.

Note: XML element names cannot contain spaces thus we replaced the space with an underscore “_” character.

The contents of the element are located between the Start and End tags.

XML is also an hierarchical format. This means that you can place elements within other elements. For example:

```
<level_0>
  <level_1>
    <level_2>
      This is the content of level 2.
    </level_2>
    This is the content of level 1.
  </level_1>
  This is the level 0 content.
</level_0>
```

How does LiveStage utilize XML?

LiveStage Professional can utilize XML in two ways.

- Input for Lists
- Storage format for LiveStage Professional Projects

The usage of XML to supply List data is discussed in the QScript Reference under “List Properties and Actions”.

It is the use of XML for storing LiveStage Professional Projects that we will address in this chapter. Please keep in mind that the use of XML is an advanced topic. You should do some background reading on XML before attempting to work with LiveStage Professional XML Project files.

What are the uses of the LiveStage XML Project Files?

The main advantage of the LiveStage XML Project File format is that it is a textual file format. You don’t need any special tools in order to edit an XML project file. All you need is a text editor.

***Note:** You must be careful when you alter an XML project file. If you make a serious mistake in altering the XML project file, LiveStage Professional may operate incorrectly or even crash.*

In addition to being easily editable, you can write a program to generate XML project files dynamically using data obtained from an online data source (i.e. Stock Quotes, Weather reports, etc...). Using a server configured with LiveStage Professional, you can generate up-to-the-minute QuickTime Movies based on live content. This is the power of XML project files.

LiveStage XML Project Files

LiveStage Professional has the ability to export an existing Project to the LiveStage Professional XML document format. Since this format conforms to the XML document format specifications, it is a tagged text file. Give it a try! Here are the steps for exporting a simple LiveStage Professional XML Project file:

- 1) Create a blank LiveStage Professional project from the File menu.
- 2) From the File menu, select "Export To XML...". You will be shown a File Save dialog.
- 3) Enter a name for the XML project file. If you have Microsoft Internet Explorer 5.0, you can use it to view your XML file. However, you must specify the file extension ".XML" for your exported XML file.

You have just exported your first LiveStage Professional XML Project file.

Here is what you will see if you opened this file with a text editor:

```
<Movie>
  <Movie_XML_File>
    <Media_File_Frame>0</Media_File_Frame>
    <Media_File_Type>1852796517</Media_File_Type>
    <Media_File_Folder_ID>-1</Media_File_Folder_ID>
    <Media_File_Spec>
      <VRefNum>0</VRefNum>
      <ParID>0</ParID>
      <FileName>" "</FileName>
    </Media_File_Spec>
  </Movie_XML_File>
  <Output_Name>"untitled"</Output_Name>
  <Window_Left>15</Window_Left>
  <Window_Top>67</Window_Top>
  <Window_Width>555</Window_Width>
```


In the above example, you'll see the tag "Movie" that encompasses the entire contents of the XML file. This is the Movie element of the XML project file. All other elements of the XML project file reside within the Movie Element.

Looking further down in the file, you'll find elements for describing various attributes of the LiveStage Professional project. In fact, you'll find most of the settings in the "Info Tab" of the LiveStage Professional document window itemized here. Simply put, the contents of the Movie element in the XML project itemizes each attribute of the LiveStage Professional Project.

LiveStage Professional XML Project File Rules

Here are some rules you can use to decipher the contents of a LiveStage Professional XML Project file:

- 1) Hex data **MUST** be enclosed in quotes
- 2) File names **MUST** be in quotes
- 3) Tags for data like a color **MUST** be in order
- 4) Tags do not have spaces and should use Underscore to separate words.
- 5) Fixed Point or Point data **MUST** be in the following form:

```
<tag name><X>xpos</X><Y>ypos</Y></tag name>
```

e.g.

```
<Image_Registration_Point><X>0</X><Y>0</Y></Image_Registration_Point>
```

The Basic Movie Script

The XML Tags created when exporting out a LiveStage project file define the information needed to create the required objects for the LiveStage Movie and its contents. The basic XML script for the movie contains information for setting the QuickTime Window size and settings. This information is within the root elements <Movie> that is used to define a LiveStage movie. The script in "LiveStage XML Project Files" on page 261 shows the sample script of a LiveStage project.

Referring to Movie Tracks in XML

Each track represented in the XML script is wrapped between the <Track> and </Track> elements. There are different information elements depending upon which kind of track is being represented. The XML track elements are equivalent to the settings you see in the Track and Sample windows within LiveStage Professional. As mentioned before an XML file is not meant to be read but to contain information. So if you wish to make modifications to your XML file, do so within the LiveStage Professional environment.

Common Sub-Element Groups

There are two common sub element groups that are nested within the Track element for all the tracks.

The first sub element group provides information about the image files used to provide the Matte effect for that track. The following sample displays the items tag and layout of the sub element. The only thing not displayed is the data between some of the tags.

```
<Track_Matte_File>
  <Media_File_Frame>0</Media_File_Frame>
  <Media_File_Type>1852796517</Media_File_Type>
  <Media_File_Folder_ID>-1</Media_File_Folder_ID>
  <Media_File_Spec><VRefNum>0</VRefNum><ParID>0</ParID><FileName>""</File
leName></Media_File_Spec>
</Track_Matte_File>
```

The second sub element group provides information to the QTList file associated with that track. The following sample displays the items tag and layout of the sub element. The only thing not displayed is the data between some of the tags.

```
<Track_XML_File>
  <Media_File_Frame>0</Media_File_Frame>
  <Media_File_Type>1852796517</Media_File_Type>
  <Media_File_Folder_ID>-1</Media_File_Folder_ID>
  <Media_File_Spec><VRefNum>0</VRefNum><ParID>0</ParID><FileName>""</File
eName></Media_File_Spec>
</Track_XML_File>
```

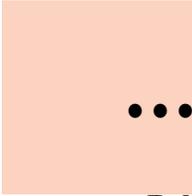
Track Sample Sub Element Group

This sub element group contains all the information for the track sample specific for that track type. This contains data for media used within this track. There can be multiple groups of the track sample sub-element groups. This is similar to track samples in the LiveStage Project Window. The track sample in each track contains its own start and end tags indicating the type of sample being stored. The tag name for all the current track samples used in LiveStage are `Sprite_Sample`, `Text_Sample`, `Effect_Sample`, `Picture_Sample`, `Color_Sample`, `Movie_Sample`, `Skin_Sample_Parent`, `Tween_Sample`, `Music_Sample`, `External_Track_File`, `Flash_Media_File` and `External_Movie_File`.

Track Information

The remaining sub elements are information relating to the track. This holds the information on how it is displayed, spatial positioning, track information and other advanced information for that track type. All the tracks contain the `Track_Type` tag. The following are the ID for each of the tracks

SPRT	Sprite Track
TXTt	Text Track
EFCt	Effect Track
CLRt	Color Track
MOVt	Movie Track
SKNt	Skin Track
TWEN	Tween Track
MODI	Modifier Track
MUSI	Music Track
Strm	Streaming Track
flsh	Flash Track
VRtk	VR Track



Glossary



Behavior

Pre-defined actions that can be applied to sprites. These can be used instead of writing a script to do common actions.

Bitmap Image

Bitmap Images are graphics composed of a 2D array of pixels. Bitmap images are also known as Raster images.

Codec

Compressor-Decompressor. Provides compression and decompression services for media samples and other data.

Drawing Mode

See graphics mode.

Effect

An effect is a type of visual media that takes 0, 1 or 2 sources of visual data and uses it to calculate the image it will display. Effects are typically used to do visual transitions such as wipes and dissolves.

Effect Track

A track in a movie that contains effect samples.

Event Handler

A script within a sprite that will be run in response to a user action. For example, the mouse click event handler will be run when the user clicks the mouse on the sprite. Event handlers can also be run by the scripts themselves.

External Sprite Track	A track in a movie that contains sprite media samples. Those sprite media sample are imported from an external Movie file. The sprite media sample cannot be modified.
Flash Track	A track in a movie that contains Macromedia Flash Media samples. Scripts can be attached to specific frames of the Flash Sample by double clicking on the Flash Sample. The Flash Track can be modified by double clicking on the Track Header.
Flash Sample Window	This window is shown when a Flash Sample is double clicked. In this window, scripts similar to the ones used for sprites can be written and attached to any of the frames in the Flash Sample.
Graphics Mode	The property of a sprite or visual track that determines how it will draw. One useful mode is transparent. Used along with an OpColor it will cause areas in the image with that color to be see-through.
Image	An image is a graphic employed in a LiveStage Professional project. Images can be bitmap or vector based.
Image Index	A property of a sprite that specifies the image to display. The index of an image is simply its ordinal position within the Image List of a given sprite sample, with 1 being the first image.
Images List	This list is visible in the sprite sample window when the images tab is selected. This list shows all the images stored in the sprite sample.
Images Tab	This tab is visible in the sprite sample window. Selecting this tab shows a list of images contained in the sprite sample.

Instrument Sample	A list of MIDI instruments. These instruments are used by sprites when the “PlayNote” QScript action is run. The index of the instrument in this sample is passed as a parameter in the “PlayNote” action. These MIDI instruments can be built-in MIDI instruments, or you can create your own instruments by using a sound file for the instrument.
Instrument Track	A track in a movie that contains MIDI instrument samples.
Interpolate	To calculate an intermediate value between two values. For example, starting with the end values of 0 and 5, the interpolated values would be 1, 2, 3, and 4.
Interpolator	An interpolator is a special type of tween that is used to interpolate time values. A tween normally interpolates its data in a linear way as time goes by. A time interpolator will cause the time to go by in a non-linear way. It is most common to use an X-Y Path to do time interpolation. One would create a path that looks like a wave form. Time would be plotted along the X-axis and the result would be read from the Y-axis. This result would be the time actually used in the tween that has this X-Y path as its time interpolator. You can use this to create an ADSR (Attack Decay Sustain Release) type volume envelope for sounds, or to make sprites move in more realistic ways by making them start slowly, accelerate, and then stop slowly.
Layer	A mechanism for determining drawing order for visual tracks and sprites in a movie. When QuickTime plays a movie, it displays the movie’s tracks according to their layer. Tracks with lower layer numbers are shown closer to the front; tracks with higher layer numbers are shown behind those tracks.

Linear	In order or sequence. To linearly change from 1 to 5 means to change like this: 1 to 2 to 3 to 4 to 5. A non-linear change from 1 to 5 would be like this: 1 to 2 to 5. The non-linear change is not smooth. Each change is not by the same amount.
Mask	A black and white image used to remove areas from the original image. The mask must be the same size as the original image and use only black and white. The black indicates that the image in this area is to be drawn and the white indicates that this area should be left alone.
Matrix	See transformation matrix or the sprite's matrix (using a path tween).
Matte	The matte defines which of the Track's pixels are displayed in a movie. A Matte can be 8 bits deep and can specify various degrees of transparency.
Media	Media actually contains the data used in the track for drawing, playing sound or other things. Media usually refers to the media samples in a track.
Media Sample	One indivisible piece of data stored in a track. Sound typically has 22,000 samples per second, while video has 24 samples per second. In the case of video, each sample is one still image.
Modifier Track	A track in a movie that contains data samples. The data samples are usually simple data types like integers or real numbers. Modifier tracks are typically used to modify some attribute in another track. One example would be to have a modifier track that contains a sequence of integers and to use that modifier track to control the image displayed by a sprite.
Movie	A collection of tracks. Each track contains media that is used to display pictures or play sound for example. A movie is typically used to display video and play sound.

Movie Controller	A user interface element displayed at the bottom of the movie that provides a set of controls to control the movie's playback.
Music Track	A track in a movie that contains sound samples. The Music Track is imported from an External Video file. The Music data cannot be modified.
OpColor	This is a color value used by some drawing modes to provide special effects. For transparent drawing it indicates the color that will be transparent. For blend drawing it indicates the blend percentage for each color channel (red, green, blue).
Playhead	A visual indicator in the tracks tab that indicates the current time in the movie.
Property	Information about an object. Sprites have properties that specify what image to draw, where to draw it and how to draw it.
Property Chip	These items can be seen in the sprite timeline. Each one represents the properties for the sprite at that point in time. Placing more than one of these in the sprite timeline will allow the properties for the sprite to be changed. The changes applied by subsequent property chips are persistent until the next property chip that changes that property.
Picture Track	A track in a movie that contains picture samples. Each sample is a single still image. A picture track is typically used to provide a static backdrop for a sprite track.
QScript	The scripting language used within LiveStage Professional to add interactivity to a movie. This language is compiled into a form that QuickTime understands and is thus limited by the capabilities of QuickTime.
QTList	A flexible data storage mechanism that is attached to QuickTime Movies and Tracks.

Rate	Specifies how fast the movie plays in multiples of its normal speed. The rate also indicates which way the movie plays. A negative rate will make the movie play backwards. A rate of -1 will make the movie play backwards at normal speed. A rate of 2.0 will make the movie play twice as fast, 0.5 half as fast.
Registration Point	The location in an image that will be used to locate the image spatially in the sprite track. When a sprite uses an image, the location of the sprite coincides with the registration point of the image. This is used to align animating images, like a walking figure. If the registration point is placed on the foot of the figure that is on the ground, then the figure will appear to walk as the images are cycled through.
Sample	See media sample.
Script Chip	These items can be seen in the sprite timeline. Each one represents the event handler scripts for the sprite at that point in time. Placing more than one of these in the sprite timeline will allow the event handler scripts for the sprite to be changed at that point in time. The changes applied by subsequent script chips are persistent until the next script chip that changes that event handler.
Sample Duration Bar	This is a blue bar that is shown in both the tracks tab and the sprites tab. It visually indicates the time that the sample is active. In the tracks tab it shows this for the selected sample only.
Script / Property Well	This is an area to the left of the sprite time line. You can drag new script or property chips from here onto the time line.
Selection Properties Panel	Displays information in the tracks tab about the currently selected track or sample.

Source	Most properties also have a source popup menu in which an alternative method of determining the property can be chosen. When the popup indicates “none”, then the property can be manually altered by the user. Tracks and tweens that can be used to supply data for the property will show up in the popup menu and can be selected as the source of information for that property. For example, if you have a modifier track with integer data in it then you can use it as the source for the image index property of a sprite.
Sprite	An object that can display an image and contain actions to be performed in response to user interaction. Sprites can move and change their image so as to produce animated sequences.
Sprite Behaviors Tab	This tab is visible in the sprite sample window when the sprites tab is selected and a sprite is selected in the sprites list. The behaviors of the sprite are shown and edited here.
Sprites List	This list is visible in the sprite sample window when the sprites tab is selected. This list shows all the sprites stored in the sprite sample. Double clicking a sprite in this list will open a window allowing you to edit the properties of the sprite and to add QScripts and Behaviours to it.
Sprite Properties Tab	This tab is visible in the sprite sample window when the sprites tab is selected and a sprite is selected in the sprites list. The properties of the sprite are shown and changed here.
Sprite Sample Window	This window is shown when a sprite sample is double clicked. In this window you can add images and sprites to your sprite sample.

Sprite Scripts Tab

This tab is visible in the sprite sample window when the sprites tab is selected and a sprite is selected in the sprites list. The scripts for the various event handlers are shown and edited here.

Sprites Tab

This tab is visible in the sprite sample window. Selecting this tab shows a list of sprites contained in the sprite sample.

Sprite Timeline

The timeline for a sprite is visible in the sprites tab when a sprite is selected. It shows the time line for the sprite sample that this sprite is contained in. The property and script chips are shown in the timeline. A repository of blank chips is to the left of the timeline where new chips can be dragged out and placed in the time line.

Sprite Track

A track in a movie that contains sprite media samples.

Time Scale

The number of time units that pass per second in a time coordinate system. A time coordinate system that measures time in sixtieths of a second, for example, has a time scale of 60. Movies typically have a time scale of 600 since 60 frames per second and 24 frames per second can both be expressed as whole numbers in this time scale.

Some example are as follows:

10 = 10 minutes

10: = 10 minutes

10:1 = 10 minutes, 1 second

:1 = 1 second

10:1.100 = 10 minutes, 1 second, 100/600ths of a second

Track	A track is a container in a movie. Each track is independent of other tracks but all tracks are linked in time. When the movie is at a particular time, then all tracks are at that time. A track contains one or more samples of a single kind of media. All media samples in a track MUST be of the same type. The media does not have to actually be stored in the movie.
Tween	Performs linear interpolation between values of various data types based on an algorithm. For instance, if a tween has a starting value of 1.0 and an ending value of 2.0 and a time duration of 1 second, then at a time of 1/2 a second the value returned from the tween would be 1.5.
Tween Track	A track in a movie that contains tween samples. A tween track can be used as the source for various object properties. They are typically used to smoothly alter a property, like the blend percentage of a sprite, or the sprite's matrix (using a path tween).
Transformation Matrix	A 3 by 3 matrix that defines how to map points from one coordinate space into another. This can be used to scale, skew, rotate, distort or translate sprites or visual tracks. The matrix tells the sprite or track where to draw. Use this to position and distort a sprite.
Vector Image	A mathematical description of an image that is more compact than a bitmap. Vectors also scale in size much better than bitmaps since each pixel is derived mathematically.
Wired Movie	Wired movies have tracks that contain scripts that run in response to a user's actions.
XML	XML stands for Extensible Markup Language, XML is a method to describe data and to focus on what the data is.

Zoom in/out buttons

These buttons allow the user to change the scale that the timeline is drawn at. Pressing zoom in will show more detail, allowing precise placement of samples.

- Creating 169
- Gradient Editor 171
- Overview 169
- Properties 169
 - Duration 170
 - Gradient 170
 - Name 170
 - Start Time 170
- Colors 99
- Composition Tab 77
- Constants
 - Custom Events 28
 - Image Names 28
- Creating Movie Tracks 91
- Creating Skin Tracks 91
- Current Time 105
- Cursors 94
- Custom Event Handlers 137

D

- Data Generating Tracks 193
- Debug Console 44
- Debug icon 44
- Debugging Console 23
- Defines Tab 28
 - Constants 28
- Delete Item 39
- Digital Audio 145
- Drag and Drop 34, 85, 91, 132
- Drawing Mode 267
- Drum Kits 145

E

- Effect 267
- Effect Track 267
 - Alignment 155
 - Creating 155
 - Creating Filters 159

- Creating Transitions 157
- Description 155
- Filter Settings 159
- Filters 158
- Multiple Sources 156
- Overhead 155
- Properties 156
- QuickTime Installation 155
- Special Effects 160
 - Cloud 160
 - Fire 160
 - Ripple 160
- Stage 158
- Tracks Created 155
- Transition Previewing 158
- Transition Timing 157
- Transitions 156, 157
- Else 229
- ElseIf 229
- EndCase 230
- EndIf 229
- EndSwitch 230
- EndWhile 230
- Errors Window 23
- Event
 - Mouse Click 85
 - Triggered 85
- Event Handler 87, 267
 - List 85
 - Scripting 85
- Event Handlers 85, 93, 110, 122, 136, 236
- Events 235
 - Frame Loaded 86
 - Idle 86
 - Key Pressed 86
 - List Received 86
 - Mouse 87
 - Mouse Click 87
 - Mouse Down 87
 - Mouse Enter 87
 - Mouse Exit 87
 - Mouse Moved 86
 - Mouse Up 87
 - Object Events 86, 87
 - Standard 86



Track Events 86
External 141
External Movies 91
External Sprite Track 268

F

Flash
Buttons 87, 94
Frame Loaded Event 87
List Received Event 87
Flash Sample Window 268
Flash Track 91, 268
Augmenting 91
Button Handlers 91
Buttons 92, 93, 94
Create 91
Frames 92
Importing 91
Object/Event Handler List 93
Sample 92, 94
Scripting 92, 94
Scripts 93
Timeline 93
Track Properties 92
Variables 93
Flash Track Events
List Received 93
Fonts 117
For 230

G

General Actions
ExecuteEvent 111
Graphics Mode 268

H

Hiding
Event Handler List 110, 136
Flash Object/Event Handler List 93
Hotspots
QTVR 133
Text 86
HTML 15

I

Idle Frequency 86, 99
If 229
Image 268
Image Index 268
Images List 268
Images Tab 268
Info Tab 29
Loop
From Beginning 31
From End 31
None 31
Palindrome 31
Looping 31
Movie
Auto Alternates 30
Compress 30
Controller 32
Copy Protect 30
Full Screen Options 32
Tag 31
Movie Name 31
Movie Settings 30
Movie XML File 31
Play
All Frames 30
AutoStart 30
Selection Only 30
Playback 30
QuickTime Version 32



- Installation
 - Adobe Acrobat Reader 10
 - QuickTime 10
- Instrument Actions
 - PlayNote 149
- Instrument Sample 269
 - Adding MIDI 146
 - Adding sounds 146
 - Playing 147
 - QuickTime Synthesizer 147
 - Sampled Sound 148
- Instrument Samples
 - AIFF 148
 - Listening to 148
 - Pitch Chart 150
 - Play Note 149, 150
 - SND 148
 - Using 149
 - WAV 148
- Instrument Track 269
 - Properties 149
- Instruments
 - MIDI 146
 - Sampled 146
- Instruments Track
 - Adding Instruments 146
- Inter-movie Communications 151
- Internet 15
- Interpolate 269
- Interpolator 269

J

- Java 15
- JavaScript 15, 221

K

- Keyboard Focus 92, 100
- Kiosks 15

L

- Layer 269
- Library 23, 37
 - Filters 39
 - Local Tab 41
 - Media Preview 40
 - Media Tab 41
 - Scripts Tab 42
 - Toolbar 38
 - Add Aliases 38
 - Add Items 38
 - Create a New Image 39
 - Edit Current Image 39
 - New Folder 38
- Linear 270
- Lingo 221
- LiveStage Developer Network 9
- LiveStage Professional
 - CD-ROM 4
 - Overview 4, 23
 - Uses 15
- LiveStage Talk Discussion List 9
- LSDN 9

M

- Macintosh 16
- Matrix 270
- Media 270
- Media Sample 270
 - Differencing 18
 - Loading 19
- Media Samples 80
 - Adding 80
 - Moving 80
- Media samples 18, 80
- Menus 47
 - Edit Menu 51
 - File Menu 48
 - Font Menu 56

- Grid Menu 56
- Help Menu 61
- Path Menu 55
- Script Menu 60
- Search Menu 52
- Time Menu 55
- Tools Menu 60
- Track Menu 54
- View Menu 57
- Window Menu 59
- MIAM 175
- Modifier Track 71, 270
 - Create Sequences 216
 - Description 215
 - Edit Sequences 216
 - Modifier Types 217
 - Event Trigger 217
 - Index/Layer/Flag 218
 - Volume/Balance 218
 - VR Angle 218
 - Properties 215
 - Using 217
- Modifier Tracks 107, 193
- Modifier Type
 - Event Trigger 217
 - Index / Layer / Flag 218
 - Volume / Balance 218
 - VR Angle 218
- Movie 270
- Movie Actions
 - GoToBeginning 179
 - StartPlaying 179
- Movie Controller 271
- Movie Monitor 190
- Movie Track
 - Child Movies 178
 - Creating 175
 - Movie Sample Properties
 - Add New URL 178
 - Layout 177
 - Loop 177
 - Movie List 178
 - Movies 178
 - Playback 177
 - Preview 179

- Take from Parent 176
- Movie Sample Window 176
- Movie Samples Properties 176
- Overview 175
- Properties 175
- Using 179
- Movie Track Actions
 - LoadChildMovie 179
- Movie Tracks
 - Preview 179
- Multimedia 15
- Music Track 271

N

- Navigation bar 15
- Next 230
- Non-scriptable Media Tracks
 - External Tracks 141
 - Overview 141
 - Types 141

O

- Object Events
 - Flash Buttons 86
 - QTVR Hotspots 86
 - Sprite 86
 - Text Hotspots 86
- Object Oriented 221
- OpColor 271

P

- Picture Track 271
 - Creating 163
 - Drag and Drop 163



- Multiple Images 163
- Overview 163
- Properties 164
- Replacing Images 163
- Sample Properties 164
 - Codec 165
 - Dimensions 165
 - Duration 165
 - File Name 165
 - Image 165
 - Name 164
 - Start Time 164
- Playhead 271
- Plug-in 16
- Pre-Defined Behaviors 242
- Preferences 61
 - Compiler Tab 64
 - Compiling 64
 - Running 64
 - Editor Tab 65, 87
 - Syntax Style 65
 - Media Tab 63
 - Caching 64
 - Searching 63
 - Misc Tab 62
 - Auto create Sprite properties 62, 104
 - Multiple Preview Windows 63
 - Startup Action 62
- Preprocessor Directives
 - #debug 44
- Product Registration 5, 9
- Project Window 23
- Property 271
- Property Chip 271

Q

- QD3D 16
- QScript 20, 87, 221, 271
 - Actions 231, 236
 - Basics 221
- Comments 232
- Continuous Lines 236, 237
- Control Statements 229
 - Conditional Statement 229
 - Loop Statement 230
 - Switch Statement 230
- Cross-Platform Issues 235
- Data 221
 - Constants 222
 - Global Scope 225
 - Local Scope 225
 - Movie Scope 224
 - Sprite Scope 225
 - Variable Declaration 222
 - Variable Scope 224
 - Variables 222
 - Arrays 223
 - Variables Declaration 222
- Event Handlers 236
- Events 235
- Expressions 228
 - Boolean 228
- Line Continuation 237
- Properties 226
- Special Characters 237
- Statements 227
 - Assignment 227
- Syntax 20
- Target Specifier 232
- Target Specifiers 233
- Targeting 232
 - ChildMovieNamed 234
 - ChildMovieOfID 234
 - MovieNamed 233
 - MovieOfID 233
 - PreviewMovie 234
 - SampleNamed 233
 - SpriteNamed 233
 - SpriteOfID 233
 - SpriteOfIndex 233
 - ThisEvent 234
 - ThisMovie 233
 - ThisSprite 233
 - ThisTrack 233
 - TrackNamed 233

TrackOfID 233
TrackOfIndex 233
TrackOfType 233
QScript Reference 23, 43
QTList 86, 271
QTList Actions
 ExchangeList 86
 QueryListServer 86
QTVR Track
 Custom Cursors 137
 Node 131
 Custom Cursors 136
 Hotspot list/Node handlers 134
 Hotspots 133, 134
 ID 134
 Name 134, 135
 Pan Tween Source 134
 Properties 134
 Properties Tab 135
 Remove Native Behavior 135
 Scripting 137
 Scripts Tab 136
 Tilt Tween Source 134
 Type 134
 Zoom Tween Source 134
 Object Movie 131
 Panorama 131
 Properties 132
 Sample 133
 Time 131
 Using 132
QuickTime 9, 16
 Authoring 4
 Player 4
QuickTime Pro 16
QuickTime Synthesizer 147

R

Rate 272
Registration Point 272

S

Sample 272
Sample Duration Bar 272
Script
 Check Syntax 29, 88, 93, 122
 Load 29, 88, 93, 122
 Management 88
 Save 29, 88, 93, 122
Script / Property Well 272
Script Chip 272
Script Editor 87, 93, 94, 110, 122, 135, 137
Scriptable Tracks 121
 Flash 92
 QTVR 131
 Sprite 97
 Text 117
Selection Properties Panel 272
SFX 145
Skin Track
 Creating 183
 Drag Area 185
 Overview 183
 Properties 184
 Samples 184
 Window Shape 185
Source 273
Source Popup Menus 193
Sprite 273
 Animation 85
 Image 85
Sprite Actions
 MoveTo 236
 Rotate 236
 SetImageIndexTo 101
 SetVisible 234
 ToggleVisible 236
Sprite animation 20
Sprite Behaviors Tab 273
Sprite Properties
 IsVisible 226
Sprite Properties Tab 273
Sprite Sample Window 273

- Sprite Scripts Tab 274
- Sprite Timeline 274
- Sprite Track 274
 - Animation 97
 - Properties 98
 - Sample 97, 100, 103
 - Sprite
 - Behavior
 - Adding 112
 - Information 112
 - Parameters 112, 113
 - Removing 113
 - Behaviors
 - List 112
 - Behaviors Tab 112
 - Button Cursor 109
 - Button Images 109
 - Button States 108
 - Button Tab 108
 - Button Variable 109
 - Chips 104
 - Adding 105
 - Deleting 105
 - Editing 105
 - Property 104
 - Script 104
 - Creating 104
 - Custom Event Handler
 - Creating 110
 - Executing 111
 - ID 111
 - Custom Event Handlers 110
 - Custom Events 111
 - Event Handler List 110
 - Image Codecs 102
 - Image Index 100, 107
 - Image Index Source 107
 - Image Name 103
 - Image Override 98, 102
 - Image Preview 101
 - Image Proxy 103
 - Image Registration Point 101
 - Image Source 102
 - Image Transparent Color 102
 - Image URL 103
 - Images 98, 100
 - Images Tab 100
 - Layer 108
 - Layer Background mode 108
 - Layer Source 108
 - List 103, 104
 - Location 106
 - Matrix 106
 - Properties Tab 103, 105
 - Registration Point 102
 - Rendering 106
 - Rendering Mode Source 106
 - Rendering OpColor 107
 - Rotation 106
 - Scaling 106
 - Scripting 101
 - Scripts Tab 110
 - Skewing 106
 - Sprites Tab 103
 - Timeline 103, 104
 - Using Image Names 101
 - Visibility 107, 108
 - Visibility Source 108
 - Sprite Matrix Source 106
 - Sprites 19
 - Image Properties 101
 - Scaling 99
 - Vector Images 100
 - Sprites List 273
 - Sprites Tab 274
 - Stage 23, 33
 - Drag and Drop 34
 - Grid 35
 - Moving
 - Sprites 35
 - Object
 - Deleting 35
 - Editing 35
 - Moving 34
 - Resizing 35
 - Playhead 33
 - Selecting
 - Sprite Track 35
 - Sprites 35
 - Tracks 34

- Toolbar 36
 - Align Bottom Edges 36
 - Align Horiz. Centers 36
 - Align Left Edges 36
 - Align Right Edges 36
 - Align Top Edges 36
 - Align Vert. Centers 36
 - Object Dimensions 36
 - Object Position 36
 - Show Grid 36
 - Snap to Grid 36
 - Spread Horizontally 36
 - Spread Vertically 36
 - Zoom In 36
- Usability Issues 34
- Start Time 72
- Streaming Track
 - Creating 189
 - Overview 189
 - Properties 189
 - Samples 190
- SWF Files 91
- Switch 230
- System Requirements 10

T

- Targeting
 - SpriteNamed 101, 111
 - SpriteOfID 226
 - Syntax 86
 - ThisSprite 232
 - ThisTrack 226
 - TrackNamed 111, 150, 179
 - TrackOfID 111
 - TrackOfIndex 111, 226
- Technical Support 5
- Text Track 70
 - Chapter Tracks 128
 - Creating 128
 - Fonts 117
 - Hotspots 86

- Properties 117
- Sample 117, 118
 - Appearance 125
 - Anti Alias 125
 - Auto Scale 125
 - Keyed Text 125
 - Visible 125
 - Background Box 126
 - Background Color 126
 - Delay 124
 - Drop Shadow 127
 - Offset 127
 - Opacity 127
 - Duration 119
 - Event Handler List 121
 - Font Settings 119
 - HotSpot
 - New 121
 - Hotspot
 - Creating 120, 121
 - Deleting 121
 - Scripting 121
 - Text Range 121
 - Hotspot Editor 119, 120
 - Hotspots 120
 - Hyper Text 120
 - Layout Tab 118, 125
 - Name 119
 - Properties Tab 118, 123
 - Scripting 120, 122
 - Scroll
 - Continuous 124
 - Horizontal 123
 - Reverse 123
 - Scroll In 123
 - Scroll Out 123
 - Scrolling 117, 123
 - Start Time 119
 - Text Attributes 120
 - Text Box 126
 - Text Editor 119
 - Text Highlighting 127
 - Color Chip 127
 - Invert 127
 - Text Color 127



- Text Range 127
 - Text Justification 125
 - Text Ranges 117
 - Text Tab 118
 - Word Wrap 124
- Using 118
- Time Scale 274
- Track 275
 - Event Handlers 85
 - Sample 86, 91
- Track Properties
 - Advanced Tab 72, 74
 - Alternates 74
 - Playback 75
 - Audio Source 149
 - Audio Tab 72, 149
 - Background Color 99
 - Bit Depth 99
 - Can Have Focus 92, 100, 118
 - Chapter Track 118
 - Composition Tab 72, 77
 - Compress 92
 - Compressed 100
 - Duration 92, 133
 - Idle Frequency 99, 118, 132
 - Is Visible 100
 - Scale Sprites With Movie 99
 - Spatial Tab 72, 76
 - Drawing 77
 - Matte 77
 - Track Tab 72, 73
 - Enabled 73
 - Name 73
 - Used In Movie 73
 - TrackEnabled 226
 - TrackVolume 226
 - Visual Track 98, 117
 - Visual Tracks 75
 - Volume 149, 190
 - Volume Source 190
- Track Tab
 - Editing 78
- Tracks 17
 - Audio 17
 - Augmentable 18
 - Compound 18
 - Creating 24
 - Data Generators
 - Modifier Tracks 18
 - Tween Tracks 18
 - Editable 18
 - Effect Tracks 18
 - Events 85
 - Flash 85, 94
 - Instruments 145
 - Movie 175
 - Picture 163
 - QTVR 85, 131
 - QTVR Tracks 18
 - Scriptable 18, 85
 - Skin 183
 - Sprite 85, 97
 - Streaming 189
 - Text 85
 - Timeline 24
 - Visual 17
- Tracks Tab 24
 - Creating
 - Color Tracks 70
 - Effect Tracks 70
 - External Tracks 71
 - Flash Tracks 69
 - Instrument Tracks 70
 - Movie Tracks 70
 - Picture Tracks 70
 - QTVR Tracks 70
 - Skin Track 71
 - Sprite Tracks 70
 - Streaming Tracks 71
 - Tween Tracks 71
 - Creating Tracks 69
 - Cut / Copy / Paste 78
 - Deleting 78
 - Drawing Layer 25
 - Duplicating 78
 - Index 79
 - Media Samples
 - Adjusting 27
 - Editing 27
 - Media samples 26

- Order 79
- Overview 69
- Playhead 79
- Re-ordering 79
- Reordering 25
- Selecting 72
- Selection Properties 80
- Selection Properties View 25
- Timeline 25
- Track Header 26
- Track List 25
- Track Locking 26
- Track properties 72
- Tracks Hiding 26
- Zoom In/Out 27
- Training seminars 9
- Transformation Matrix 275
- Tutorials 3
- Tween 275
- Tween Track 107, 193, 275
 - Description 197
 - Editor 199
 - Sample
 - Interpolator 200
 - Tween Sample 198
 - Tween Tutorial 209
 - Tween Type
 - Graphics Mode 208
 - Index / Layer / Flag 203
 - Matrix 205
 - MultiMatrix 208
 - Path to Matrix Rotation 206
 - Path to Matrix Translation 205
 - Path to Matrix Translation and Rotation 206
 - Path X to Y 206
 - Path Y to X 207
 - Polygon 207
 - Spin 207
 - Time/Event Trigger 205
 - VR Angle 208
 - Tween Types 203
- Tween Tutorial 209

V

- Variables
 - GlobalVars 92, 137, 225
 - LocalVars 92, 137, 225
 - MovieVars 92, 137, 224
 - SpriteVars 92, 137, 225
- Vector Image 275
- Version 32
- Video Tracks 141
- VR 132
- VR Track Properties 132
- VR Track Samples 133

W

- While 230
- Widgets 34
- Windows 16
- Wired Movie 221, 275
- Wired Movies 4

X

- XML
 - Common Element 264
 - Description 259
 - Hierarchy 259
 - Rules 263
 - Track Information 265
 - Track Sample Sub Element 265
 - Uses 260

Z

- Zoom in/out buttons 276



Zoom Out 36

