

LIVESTAGE™
PROFESSIONAL
QUICKTIME AUTHORIZING ENVIRONMENT

Copyright © 1999 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 of 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 book.

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

Totally Hip, the Totally Hip logo, LiveStage, the LiveStage logo, WebPainter, the WebPainter logo and QScript are trademarks of Totally Hip Software Inc. Apple, the Apple logo, Mac, Macintosh, the Mac logo, QuickTime and the QuickTime logo are trademarks of Apple Computer, Inc. Flash is a trademark of Macromedia, Inc. Acrobat is a trademark of Adobe Systems Incorporated. All other trademarks are the property of their respective holders.

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

Contents

Introduction - Welcome

About this Manual.....	i
Learning LiveStage Professional	i
Contents of the LiveStage Professional CD.....	ii
Registration.....	ii
About Totally Hip Software Inc.....	iii

1. Getting Started

Where to Get Information.....	1-1
System Requirements.....	1-1
Before Installing LiveStage Professional.....	1-2
Installing LiveStage Professional	1-2

2. LiveStage Professional Basics

What is LiveStage Professional?	2-1
What can I do with it?.....	2-1
What is QuickTime?	2-2
What is QuickTime Pro?.....	2-3
How do we do it?	2-3
What is a Track?.....	2-3
What is a Media Sample?	2-4
What is a Sprite?	2-4
What is Sprite Animation?	2-4

3. The LiveStage Professional Environment

Project Window.....	3-1
Tracks Tab.....	3-2
Defines Tab.....	3-5
Info Tab.....	3-6
Stage Window.....	3-8
Library.....	3-11
Media Tab.....	3-13
Local Tab.....	3-14
Scripts.....	3-14
Using Media in the Library.....	3-15
QScript Reference.....	3-17
Debug Console Window.....	3-17
Menu Overview.....	3-20

4. Working In The Tracks Tab

Overview.....	4-1
Creating Tracks.....	4-1
Selecting Tracks and Media Samples.....	4-2
Track Properties.....	4-3
Editing Tracks and Media Samples.....	4-7
Track Order.....	4-8
The Playhead.....	4-8
Selection Properties View.....	4-8
Media Samples.....	4-9

5. Sprite Tracks

Sprite Track Overview.....	5-1
Sprite Samples.....	5-3
Images Tab.....	5-4
Sprites Tab.....	5-8

6. Instrument Tracks

Adding Instruments.....	6-1
MIDI Instruments.....	6-3
QuickTime Synthesizer Instruments.....	6-3
Sampled Instruments.....	6-4
Listening to your Instruments.....	6-4
Using Instruments.....	6-5
Exotica.....	6-6

7. Effect Tracks

Setting the Effect Properties.....	7-1
Transitions.....	7-2
Creating a Transition Effect.....	7-2
Adjusting Transition Effect Timing.....	7-3
Viewing Transitions.....	7-4
Filter Effects.....	7-4
Creating a Filter Effect.....	7-4
Setting the Filter Effect Properties.....	7-5
Special Effects.....	7-6

8. Tween Tracks

Introduction.....	8-1
What is a Tween?	8-1
Tween User Interface	8-1
Tween Editor	8-2
Name	8-3
Start & End Time	8-3
Interpolator.....	8-4
Type.....	8-6
Tween Types	8-7
Index / Layer / Flag Tweens.....	8-7
Time / Event Trigger Tweens.....	8-8
Matrix Tweens.....	8-9
Path to Matrix Translation	8-9
Path to Matrix Rotation.....	8-9
Path to Matrix Translation and Rotation.....	8-10
Path X to Y.....	8-10
Path Y to X.....	8-10
Polygon	8-11
Spin	8-11
Graphics Mode Tween	8-11
VR Angle	8-11
MultiMatrix Tween	8-12
Tutorial.....	8-12

9. Modifier Tracks

Properties	9-1
Adding Sequences.....	9-1
Using Modifiers	9-3
Modifier Types	9-3

10. Picture Tracks

Creating a Picture Track.....	10-1
Picture Track Properties.....	10-2
Picture Sample Samples.....	10-2

11. Text Tracks

Text Track Properties	11-1
Using Text Tracks	11-1
Text Tab.....	11-3
Hot Spots.....	11-4
Display raw Text.....	11-4
Font, Styles, and Sizes	11-4
Interactive Text	11-5
Text Samples.....	11-5
Properties Tab	11-7
Chapter Tracks	11-14

12. External Movies

Movies in the Library.....	12-1
Using External Movies	12-2
External Track Properties.....	12-2
Copying External Tracks.....	12-3
Removal of External Tracks.....	12-3
Flash Movies.....	12-3
Flash Samples	12-4
Script Management.....	12-5
QuickTime VR Movies.....	12-6
QuickDraw 3D Movies.....	12-6

13. Behaviors

Using Behaviors.....	13-1
Parameters.....	13-2
Pre-Defined Behaviors	13-3
Writing Behaviors	13-9
Sections	13-10
Name Section.....	13-10
Description Section.....	13-11
Parameters Section.....	13-11
Event Handlers.....	13-12
Some Guidelines	13-13

14. Introduction to QScript

QScript Basics.....	14-1
Data.....	14-1
Constants.....	14-2
Variables.....	14-2
Properties	14-5
Statements.....	14-6
Assignment Statements.....	14-6
Control Statements.....	14-8
Target Objects	14-12
Putting It All Together.....	14-14
Events.....	14-14
QScript Actions.....	14-16
Continuous Lines	14-16
Line Continuations.....	14-17

15. QuickTime VR Movies

Introduction.....	15-1
What is QuickTime VR Movie	15-1
Using VR Movies	15-2
VR Tracks Properties	15-2
VR Track Samples	15-3
General Properties.....	15-4
Scripting VR Tracks.....	15-7
Custom Cursors.....	15-7

Glossary

I N T R O D U C T I O N

Welcome

About this Manual	i
Learning LiveStage Professional	i
Contents of the LiveStage Professional CD	ii
Registration	ii
About Totally Hip Software Inc.	iii

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, DVD or any other manner that allows QuickTime movie playback.

You'll find that LiveStage Professional is an extremely powerful and flexible tool only limited by the creativity of the designer. Unleash your creativity with this comprehensive QuickTime media creation and integration mechanism.

About this manual

The LiveStage Professional User's Manual includes the product information as well as the main product tutorials. The User's Manual provides detailed information about the LiveStage Professional tools and commands. It is designed to be used as a reference guide in your every day work with LiveStage Professional and is also included electronically in PDF format. The tutorials are designed to show you the basic operations of the product and content creations methods. You will need Adobe Acrobat Reader 4 to view the PDF version of the manual.

Before using this manual, you need to install the program by following the instructions given in Chapter 1. Chapter 1 also describes the minimum system requirements for 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.

Learning LiveStage Professional

LiveStage Professional includes the following materials to assist you in learning and using the product. The LiveStage User's Manual contains information on using LiveStage commands and features. The manual also includes tutorials covering introduction to the product and some of its main features. The LiveStage Professional

CD and the Totally Hip Web Site also contain additional sample projects that are either documented or commented to provide further assistance and actual examples for you to use and follow.

Totally Hip also manages an email discussion list for LiveStage users that you can subscribe to. Here you can get some valuable information such as asking specific questions, reporting bugs, showing your work, seeing sample projects and scripts, requesting features in future releases etc. The list is open to the public and active members include Totally Hip Software staff and experienced users. To sign up to the list see Chapter 1 of this manual.

Contents of the LiveStage Professional CD

The LiveStage Professional CD contains the following:

- Readme File
- Software License Agreement
- LiveStage Professional Software
- LiveStage Professional User's Manual
- WebPainter 3 Software by Totally Hip
- QuickTime 4 Software by Apple
- Adobe Acrobat Reader 4 Software by Adobe
- Animation Library
- Music Library
- GIF Utilities
- Web Badges

Registration

Totally Hip Software would like you to register your software so that we can provide you with technical support, inform you about new software developments and send you important LiveStage Professional related information. Please register your copy online at: <http://www.totallyhip.com/Link/ProductsRegister.html>. You must register your copy of LiveStage Professional to receive the QuickTime Pro upgrade offer.

About Totally Hip Software Inc.

Totally Hip Software Inc., founded in 1995 is a developer and marketer of interactive Web development tools, including its award-winning LiveStage Professional, LiveStage and WebPainter software. The company is located in beautiful Vancouver, British Columbia, Canada and is a public company that is traded on the Alberta Stock Exchange (Trading Symbol: ASE:THW).

For more information and current news about Totally Hip Software you can check the Totally Hip home page at <http://www.totallyhip.com> or send an email to livestage@totallyhip.com.

CHAPTER 1

Getting Started

Where To Get Information	1-1
System Requirements	1-1
Before Installing LiveStage Professional	1-2
Installing LiveStage Professional	1-2

Getting Started

Where To Get Information

We highly recommend that you read through the LiveStage Professional manual first as it contains a lot of valuable information. Many of the concepts and techniques in LiveStage Professional may be relatively new to users and can take a little while to master. Be sure to go through the tutorials; this will help you understand the product and its features better.

If you are having problems with LiveStage Professional you may contact our tech support department at techsupport@totallyhip.com. They will be happy to assist you with installation support or bug reports.

To discuss ideas, share tips and tricks or ask questions visit the Totally Hip LiveStage Talk discussion list. This service is available to all users of LiveStage Professional and can be joined by sending an email to requests@lists.totallyhip.com with the subject line of “subscribe LiveStage-Talk”.

Also for current software news visit the Totally Hip web site at <http://www.totallyhip.com> and the LiveStage Developer News at <http://www.totallyhip.com/lsdn/>.

System Requirements

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

- Power Macintosh 266 MHz or faster
- Mac OS 8.0
- QuickTime 4 Software
- 10 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 or later has been installed onto your computer. We recommend that you install all 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 updater.

To install QuickTime follow these steps:

1. Disable virus protection extensions and restart the computer
2. Double-click “QuickTime Installer.smi” file
3. Double-click ‘QuickTime Installer’ to launch the installer
4. Follow the installation instructions as they appear on the screen.
5. After completing all the installations, restart the computer.
6. To receive a free upgrade to QuickTime Pro register your copy of LiveStage Professional online at <http://www.totallyhip.com/Link/ProductsRegister.html>.
7. Once registered you can request to receive the QuickTime Pro unlock code and installation instructions from Totally Hip Software.

To install Adobe Acrobat follow these steps:

3. Double-click ‘Reader Installer’ to launch the installer
4. Follow the installation instructions as they appear on the screen.

Installing LiveStage Professional

You do not have to remove LiveStage 1.0 to install LiveStage Professional.

1. Disable virus protection software and restart your computer
2. Double-click the LiveStage icon on the CD-ROM
3. Follow the installation instructions as they appear in the dialog box.
4. Accept the license agreement.
5. Enter your registration information, if asked.
6. Select the destination directory. (must be different than LiveStage 1.0)
7. Launch LiveStage Professional from the application icon.

The first time LiveStage is launched, 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 2

LiveStage Professional Basics

What is LiveStage Professional?	2-1	What is a Track?	2-3
What can I do with it?	2-1	What is a Media Sample?	2-4
What is QuickTime?	2-2	What is a Sprite?	2-4
What is QuickTime Pro?	2-3	What is Sprite Animation?	2-4
How do we do it?	2-3		

LiveStage Professional Basics

What is LiveStage Professional?

Before we get started, perhaps a simple question should be asked: What is LiveStage Professional? LiveStage Professional, from Totally Hip Software, is a powerful and extensive multimedia development tool designed for creating interactive rich media content for the Internet, CD-ROM and DVD-ROM. In short it is a QuickTime Authoring Environment.

It provides designers with tools to develop compelling interactive content of the highest quality and the lowest-bandwidth. LiveStage Professional creates interactive QuickTime Wired Movies that are viewable by a large audience because QuickTime is the most popular multimedia technology for both Windows and Macintosh computers.

What can I do with it?

LiveStage Professional can be used to create engaging interactive presentations that encourage viewers to explore the designer's web site or where ever 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 web master working on your site and you want to create a new navigation bar for your pages. Where do you start? Well, first we put the graphics together, 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 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? Lets look at the same scenario with LiveStage Professional. You put together graphics in your favorite graphics' program. Now you need to lay these out for 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. Drop a reference in your web page to the fully self-contained QuickTime movie 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, tell 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. They 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 individual abilities will dictate user comfort and learning progress.

What is QuickTime?

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

QuickTime is an enabling technology comprised of pieces of software that extend the ability of the Macintosh or Windows operating system to handle dynamic media. Totally Hip designed and developed LiveStage to allow end-users to harness the power of QuickTime technology and turn it into an interactive application development environment. The combination of LiveStage 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 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 for free by Apple Computer, Inc.

What is QuickTime Pro?

QuickTime Pro gives the user full access to the power of QuickTime by expanding on the powerful editing capabilities of QuickTime. With QuickTime Pro your QuickTime Player application goes from being just a media viewer to being a powerful editing tool.

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.

Totally Hip provides a free upgrade to QuickTime Pro when LiveStage Professional is registered. Once registered, we will send the upgrade key for QuickTime Pro with instructions on how to install it.

How do we do it?

To some, LiveStage Professional may feel complex, however by understanding a few basic terms and concepts you can make your use of LiveStage Professional much easier to understand.

For those of you who are familiar with QuickTime, you will find many concepts from there that are equally applicable in LiveStage Professional. If you are new to QuickTime then the following sections will outline some of the basic concepts and ideas that are used in LiveStage Professional.

A glossary of terms is also included at the end of the User Manual.

What is a Track?

Tracks operate as the basic building blocks of a LiveStage Professional project, they act as the containers of media within a LiveStage Professional project. Tracks contain units of media called Media Samples which define the type of the track. By incorporating multiple tracks you can create sophisticated effects like interactive video, scrolling text and synchronized sound.

In LiveStage Professional you can have editable and non-editable track types. Editable tracks are those in which the user has the ability to change or edit samples. Sprite, Instrument, Effect, Tween, Modifier, Picture and Text tracks are all fully editable within LiveStage Professional. Flash tracks are augmentable. Non-editable tracks are those which can be brought into LiveStage Professional but may not be directly changed or edited. These include video, sound, QuickDraw 3D, VR and any other QuickTime supported media types.

What is a Media Sample?

Media samples are indivisible units of data for a particular media type. A media sample contains chunks of media within a track, for example, each media sample of a picture track would contain a single image, while each sample of an effect track would contain information on the effect type and its properties.

What is a Sprite?

When creating interactive content, sprites become very important. 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 in 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 their environment, including interaction with other sprites or with the user.

What is Sprite Animation?

Sprite animation is very 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. This allows the animation to be different each time. Sprite animations can be altered from frame to frame and have infinite durations.

CHAPTER 3

The LiveStage Professional Environment

Project Window	3-1	Local Tab	3-14
Tracks Tab	3-2	Scripts Tab	3-14
Defines Tab	3-5	Using Media in the Library	3-15
Info Tab	3-6	QScript Reference	3-17
Stage Window	3-8	Debug Console Window	3-17
Library	3-11	Menu Overview	3-20
Media Tab	3-13		

The LiveStage Professional Environment

LiveStage Professional is a project based environment which makes the creation of interactive QuickTime movies relatively simple whether you are a developer or content creator. LiveStage Professional can incorporate many different types of content into your movies. Some of these items are created within the LiveStage environment and others are imported from external files.

Creating a QuickTime movie with LiveStage Professional involves creating a project, adding media to it by dragging them from the Library Palette or from the Finder to the Tracks Tab in the Project Window and then generating the movie.

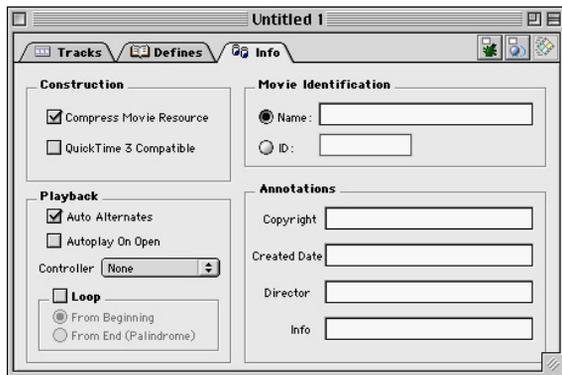
When you first open LiveStage Professional you will be presented with a dialog box asking if you want to open an existing project or create a new one. After a file has been opened or created at least one window, the Project Window, will be displayed. Other windows may be displayed depending on what was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

Clicking on the close box of the Project Window closes the project. If you have made editing 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 outlines your project and is the main jumping off point for all operations you will perform in the LiveStage Professional environment.

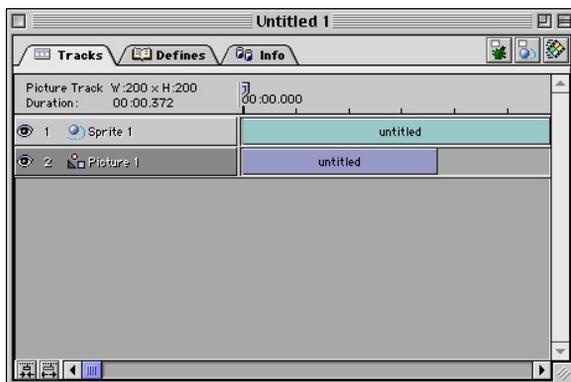
There are three tabs in the Project Window: Tracks, Defines and Info. All editing of the projects tracks and samples is accessed through the Tracks Tab. The Defines Tab is used for creating any constants that you use in scripting. General movie properties are specified in the Info Tab.



When each tab in the Project Window is clicked, a corresponding editor is displayed where you may view or edit characteristics of the project. The following section outlines each of the three 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.

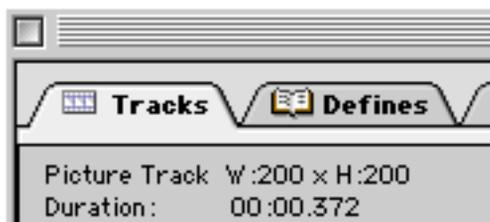


Tracks are created using the Create item in the Tracks Menu. To use previously prepared movies such as QuickTime or Macromedia Flash movies, drag them from the Finder directly into the Tracks Tab of the Project Window.

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 (Command K). These samples can then be dragged to new positions within the track that contains them or resized by dragging their ends.

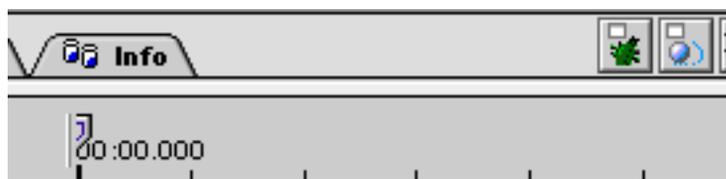
Selection Properties View

Displays information about the selected track or sample. If you have selected a track, the height, width and duration will be displayed. When a sample is selected, the start time and duration of the sample will be shown.



Timeline

Shows a visual representation of time. Time increases as you go to the right. The Timeline also shows the Playhead. Moving this sets the current time and simultaneously updates the Stage Window (described later). The scale used to show time in the Timeline may be adjusted using the Zoom In and Zoom Out buttons.



Track List

A list of tracks in the project is displayed in ascending numerical order along the left side of the window. The order of the tracks in the track list in no way affects the visual organization of the tracks.

1	Sprite 1	Button Sprit...
2	T Text 1	Credits
3	Sound 1	Sound track from file ...
4	Video 1	Video track from file : ...

Track Header

The Track Header is part of the Track List. It appears along the left side of each track in your document and shows an icon indicating whether the track is visible in the Stage Window, its index number, an icon indicating the type of track and the track's name. You can drag a track to a different position in the list of tracks. This changes its index number but NOT its drawing order. Double clicking on the track header will bring up a property window specific to the type of track. Here you can change several attributes of the track. You can hide and show a track in the Stage by clicking on the “eye” icon. This does not affect the track in the final movie, it only affects it in the Stage.



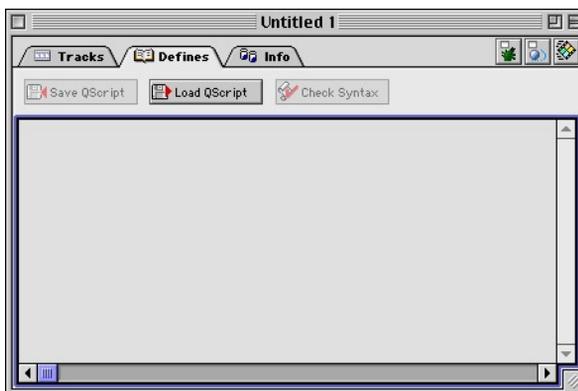
Track Sample

This is a colored bar that represents a media sample in a track. You can add more media samples to a track by choosing “Add Sample” (Command K) from the Tracks Menu. Double clicking on a media sample will bring up a properties dialog specific to that media sample. The media sample's start and end times can be changed by dragging the ends of the sample or by dragging its center. Holding down the option key when 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.



Zoom In/Out

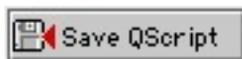
Zoom In and Zoom Out adjust the time scale used for displaying the timeline in the Tracks Tab. These buttons are located in the bottom left corner of the Tracks Tab. You may also use the Zoom In (Command **I**) or Zoom Out (Command **O**) menu items from the View Menu.



Defines Tab

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

Save QScript



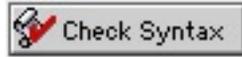
This button allows you to save the contents of the script editor field to a QScript file which may be retrieved at a later time.

Load QScript



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

Check Syntax Button



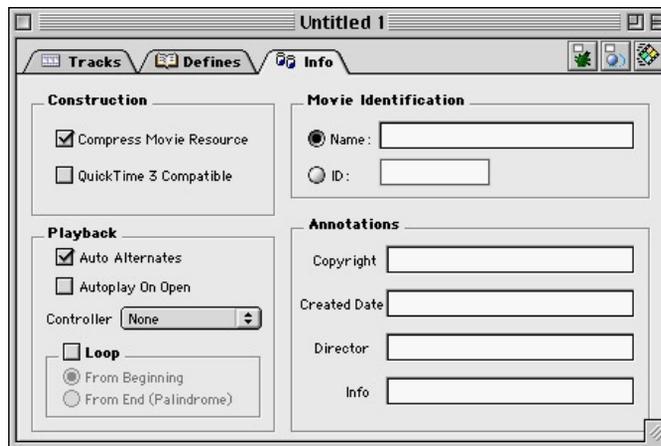
Checks the syntax of the script in the Defines tab. If an error is found an Errors window will appear to notify you of the error. This will list the error and the line number of your script where the error was detected.

Script Editor

This is a scrollable edit field where the script for your defines will be added or edited.

Info Tab

The Info Tab provides you with general information about your LiveStage Professional project. It is in this tab that you can setup copyright information and various movie playback properties. The properties displayed in this tab are placed into related groups. We will go through each group outlining the properties contained therein.



Construction

The construction section deals with all the properties associated with the actual creation of the movie. There are two properties in this group.

Compress Movie Resource	The movie resource will be compressed if this is checked. This will make a smaller movie but in some rare circumstances an application may have trouble opening it. Currently, the only application known to have problems reading this type of movie is Dumpster, a developer tool from Apple.
QuickTime 3 Compatible	This will guarantee the movie will be QuickTime 3 compatible when checked. Some QScript statements will cause syntax errors because they will only work in QuickTime 4.

Playback

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

Auto Alternates	Checking this will allow QuickTime to automatically select enabled 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.
AutoPlay on Open	If you check this the movie will automatically start playing when it is launched.
Controller	Select the type of movie controller to use from this popup menu.
Loop	Select this property if you want your movie to loop. Once looping is turned on you can select the type of looping.

From Beginning	Plays the movie indefinitely starting at the beginning of the movie after it reaches the end.
From End (Palindrome)	Plays the movie indefinitely, forward from the beginning and then backwards from the end.

Movie Identification Section

The Movie Identification Section allows you to give the movie a unique name or ID that will be stored inside the movie. This method of movie identification is used if you wish to have movies that ‘talk’ to each other while they are playing. These fields should be ignored if you are not creating movies that communicate with each other.

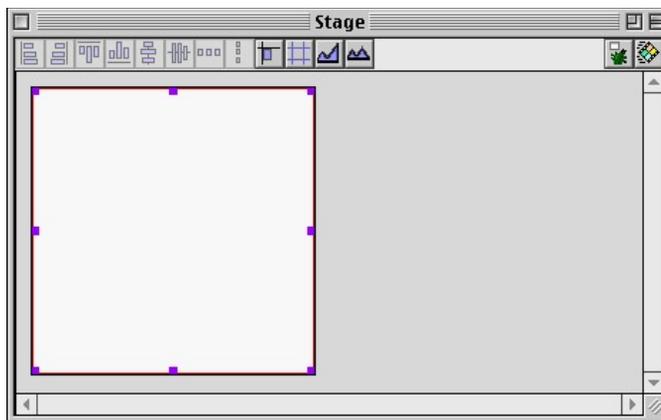
Name	Select this radio button in order to tag the movie with a name and then enter the name to be used as a tag in the field.
ID	Select this radio button in order to tag the movie with an numeric ID and then enter the ID to be used as a tag in the field.

Annotations Section

Enter information specific to your movie here. Some of this information is displayed in QuickTime player, including Copyright, Created Date, Director and Info.

Stage Window

The Stage Window shows a representation of all the visible elements in the movie. This window acts as a layout area for all of the movie's visual elements. In the Stage Window you will get an approximation of how the visual elements will look in the movie. You should note however that this is just an approximation, thus certain items like effects will not be shown.



Visual tracks can be moved or resized within the Stage Window. Other visual items like sprites can only be moved within the Stage Window.

Objects Displayed in the Stage Window

Only visual tracks like the Sprite Track, Picture Track and the Text Track are displayed in the Stage Window. The images representing the tracks are displayed for the entirety of the movie even though the sample may not be active during the time that you are viewing. The methods to add the tracks to the project are explained in more detail in the Tracks section.

Dragging and Dropping onto the Stage

Certain files can be dragged and dropped onto the Stage from the Finder and appropriate tracks will be created for them automatically. Dragging one or more images onto the Stage Window or the Tracks Tab will create a picture track that contains 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 premade button creator AppleScripts included in the Libraries Other folder. 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.

Stage Usability Issues

The Stage not only shows a representation of the output, but it also provides the capability to manipulate the size and position of a visible track. The position and size of a track will remain the same throughout the entire movie file. Select the track by clicking on it in the Stage Window. A highlight with handles on the corners and edges of the highlight points will appear. Dragging the track anywhere in the Stage Window will move the track, clicking and dragging the handles will resize the track.

The position of an image inside a Sprite Track can be set at a specified time. Therefore, the image position within a Sprite Track can be changed at different points in time. Click and drag the image within the Track to move it around within the Track object.

Groups of tracks can be selected and manipulated all at once. To select a group, hold down the shift key and select all the Tracks. 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 Stage Window.

Toolbar

The Stage has a toolbar across the top of its window. The toolbar provides features that aid in viewing and aligning objects on the stage at a given point in time. The buttons are as follows:

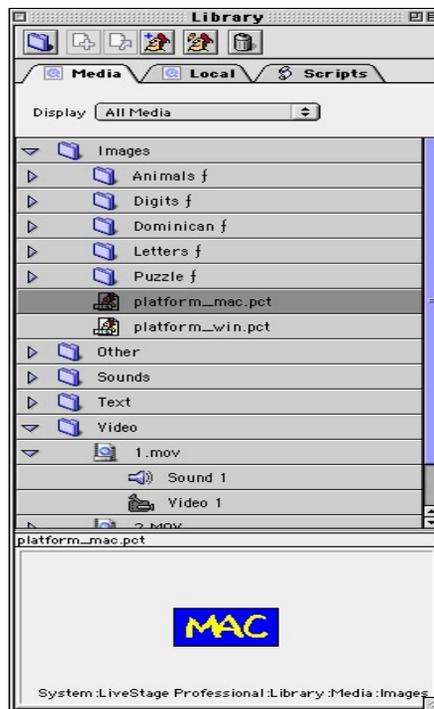
Align Left Edges	Aligns the left edges of the selected sprite(s).
Align Right Edges	Aligns the right edges of the selected sprite(s).
Align Top Edges	Aligns the top edges of the selected sprite(s).
Align Bottom Edges	Aligns the bottom edges of the selected sprite(s).
Align Horiz. Centers	Aligns the horizontal centers of the selected sprite(s).
Align Vert. Centers	Aligns the vertical centers of the selected sprite(s).
Spread Horizontally	Horizontally spaces the selected sprites evenly using the left and right most sprites as edges.
Spread Vertically	Vertically spaces the selected sprites evenly using the top and bottom most sprites 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.

Library

The Library is a convenient way to view and access media and to allow you to drag and drop media into the project. It is accessed from the Window Menu by choosing Library Window (Command L).

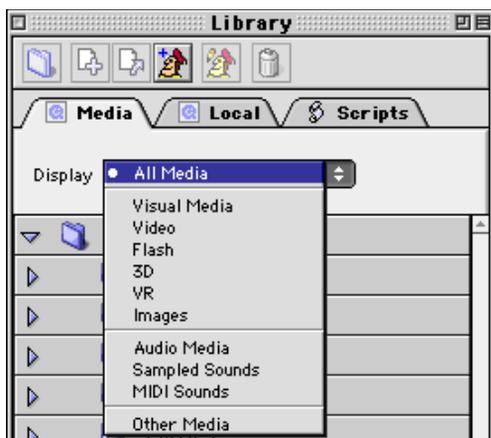
You use the Library to assist you in keeping the media for your projects organized. Each tab in the Library operates much like the Finder, you may create folders to store files or other folders. Items may be added or deleted from the Library and you can drag and drop these items from the Library into different parts of LiveStage Professional.

Each tab in the Library contains a few standard folders to help you manage your content.



Filters

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.



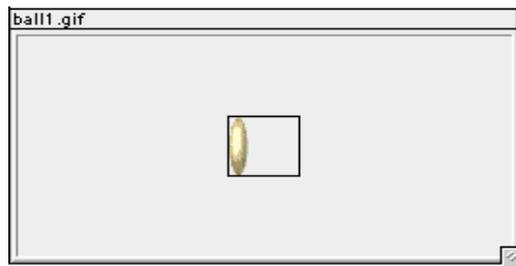
Filters ease the process of looking for content when you have a large number of files in your Library. The same set of filters are provided in both the Media and Local tabs. For example, you can filter out everything except for Visual Media or view only QuickTime VR media.

Filtering does not affect the actual contents of your Library, only the view that you are given in the current tab. To remove a filter 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. It gives you a preview or thumbnail view of certain types of content in your Library. When you select an item in any one of the tabs in the Library, the preview area will attempt to display the contents of that item.

The preview area displays scripts and any image format that is supported within QuickTime. With image types that have layers you may also view each layer individually by expanding the image in the library and clicking on the particular layer.

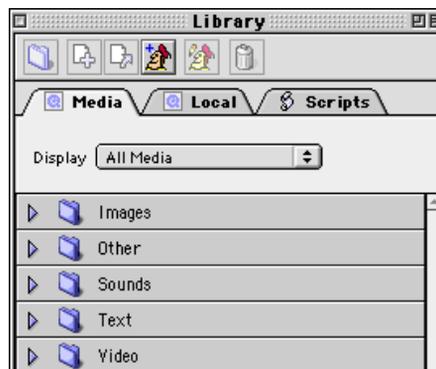


Media Tab

The Media Tab operates as a global media repository for LiveStage Professional. 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.

A series of folders provide default groupings for the media that ships with LiveStage Professional. These include Images, Sounds, Text, Video and Other as seen in the picture below.

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 is project specific. If you have created a Library folder where your project is stored you can have a local library that contains any content you want to use for just that project. By default the Local Tab will contain the same content as the Media Tab since all new projects do not yet have their own Library folder.

Scripts Tab

The Scripts Tab in the Library palette displays the list of scripts available to be used in the document. There are three folders created by default that may be opened or closed by either clicking on the arrow to the left of the folder icon, or by double-clicking the folder icon itself. The three Script folders are AppleScripts, Behaviors and QScripts. It is possible to create sub-folders within the two main folders so those custom script files can be placed in an organized manner.

The list box above the folders lists the filters to be used to filter out the specified scripts to display in the folders. Currently the filters are QScripts, AppleScript and All Scripts.

The area below the folder is a preview window that will display the script code for the selected script.



You can drag the icon representing the script from the library palette directly into the script editing field of a sprite or into the Defines Tab of the project window. This will cause the script to be copied into the script edit field.

Using Media in the Library

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 QScript it can be dragged into any Script Editor and will be added to the contents of that Script Editor.

There are a variety of visual media types that can appear in either Media or Local libraries that contain multiple elements or layers. Examples of these are QuickTime movies with more than one track and Photoshop files with multiple layers. These items appear in the library with a twist down triangle beside them. This button allows you to view the elements inside the file and extract only those items you require.

In the case of Photoshop files each layer in the file is displayed as a series of frames. An image composed of each layer is displayed in the first frame, the background is displayed as the second frame and all remaining layers are displayed after that.

The files in the Library are not stored within the document itself. 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 the flexibility in defining custom folders to contain media for individual projects.



Add Items

The assets in the Library do not get stored within the movie project. Instead, they are kept independent so they can be accessible to any movie project. In addition, media editors can gain direct access to the media 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 Items 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 Items, Add Aliases makes files accessible from the Library. In this case, however, it is working with an alias to a media file that is stored elsewhere other than in the Library folder. Ensure the alias is placed in 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 Items).



Create a New Image

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



Edit Current Image

If an image needs to be modified while a project is being developed, select it in the Image tab and click the Edit Current Image icon. LiveStage will launch 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.

QScript Reference

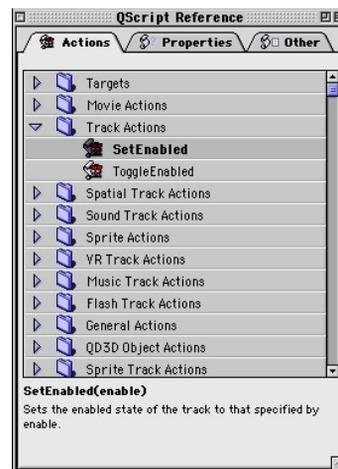
The QScript Reference provides you with an online reference for the QScript language that is built into LiveStage Professional. This palette contains all of the QScript keywords including a summary of what each does and which parameters (if any) are used.

The more than 120 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. These keywords can be dragged and dropped into any scripting pane to avoid typing in the QScript keyword.

The Actions Tab contains QScript statements that perform an action on an object or actions to get an object. These include actions like the SetVisible and SpriteNamed keywords.

The Properties tab lists the properties available in all of the different objects in your document. In here you will find items like the NumSprites and isVisible keywords. The properties are broken down into various groups which refer to the various items you will find in a movie. These include the Movie Properties, Sprite Track Properties and QuickTime Properties.

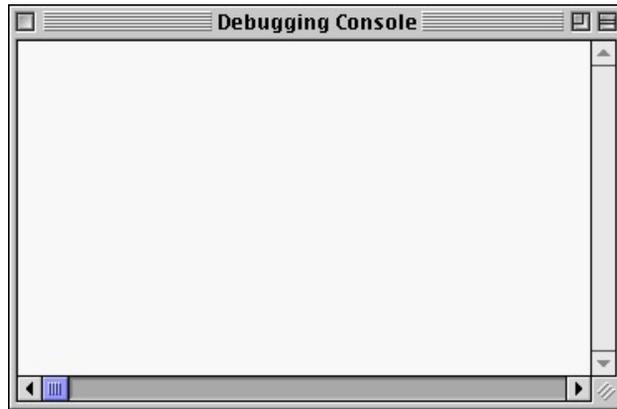
The Other Tab is a collection of unrelated elements that exist in the QScript scripting environment. This section includes keywords for defining variables, operators, control statements and various constants.



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 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 command.

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. You may also press Command Y as a shortcut to selecting the Debug Console menu item.



The Debug Console Window can display two types of debugging information. The first type is where the Debug Console Window displays the QScript commands that are being executed. To enable this functionality open the LiveStage Professional preferences and check the “Show source in debugger” item in the Compiler group. 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.

Alternatively you can place custom debug messages in your QScripts that notify you when parts of your script have been executed. You do this by calling the QScript DebugStr command passing in a string or numeric value. Any information passed into the DebugStr command will be displayed in the Debug Console Window.

For example, if you placed the following debug command into the Mouse Click event handler of a sprite:

```
DebugStr("Sprite was clicked on")
```

When you run the movie and click on that sprite, the “Sprite was clicked on” message will be displayed in the Debug Console Window. This can be used with or without the “Show source in debugger” option.

The above example provides a very simple sample of how you can use DebugStr to assist in your debugging efforts. Lets take a look at more of a real world example. In this case we will display a message and the value of a variable in a QScript. In this case we have a small project that contains a single sprite and in that sprite’s “Frame Loaded” handler we define a single variable which will be used as a counter.

```
GlobalVars inc
```

```
inc = 0
```

In the “Mouse Enter” handler we will increment our counter and then display a message “counter value: ” which will be follow by the value in the counter. The final result is that each time you move the mouse into the sprite the current value of the counter will be displayed in the Debug Console Window. Here is the script we put into the “Mouse Enter” handler.

```
GlobalVars inc
```

```
SpriteVars tempstring
```

```
// increment the counter
```

```
inc = inc + 1
```

```
// create the string and then append
```

```
// the counter value to it
```

```
SetString(tempstring, "counter value: ")
```

```
AppendString(tempstring, inc, tempstring)
```

```
// send the debug string to the debug window
```

```
DebugStr(tempstring)
```

The output displayed in the Debug Console Window when you run this movie and move the mouse over the sprite would be the following:

```
counter value: 1  
counter value: 2  
counter value: 3  
counter value: 4  
counter value: 5
```

```
.  
.
```

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

Menu Overview

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

In most windows in LiveStage Professional there is a small toolbar which provides a shortcut to a few different menu items. The toolbar 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 operate on various aspects of a file. Some of the items you will find here include the “New Project” menu item which creates a new empty project for you to work with. You will also find other items like “Run Wired Movie”, “Save” and “Close”.

New Project (Command N)

Creates a new Project Window with the Info Tab selected allowing specific settings for the new project to be defined.

Open... (Command O)

Brings up the Open dialog box displaying projects created with LiveStage. In the Open dialog box, select the project file, and click the open button, or double-click the movie project name. This opens LiveStage files, Movie files, and Flash Movie Files.

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

Open Recent

Brings up a popup menu listing the most recent project files opened by LiveStage. Selecting any of the files in the list will open that LiveStage Project file. If the document you are trying to open is already open in LiveStage Professional, the Project Window will be displayed.

Close (Command W)

Closes the current window. If 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.

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.

Revert

Revert 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.

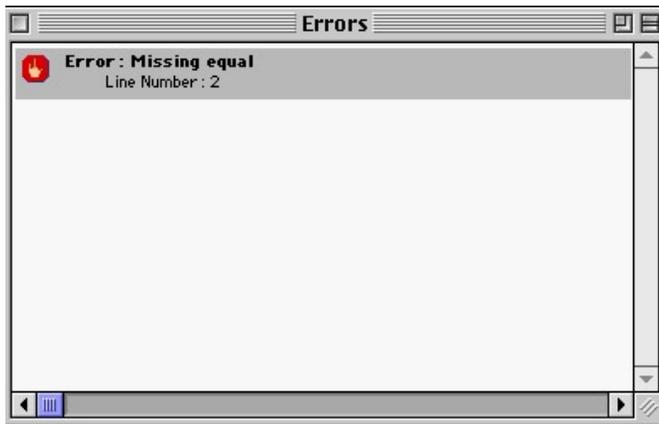
Run Wired Movie... (Command R)

Compiles the current project into a movie and plays it. If there are no QScript errors reported, the movie file will run in a separate window. If there are script errors in the project, the Errors Window will appear notifying the user of the scripts that have errors in them. Any script errors must be corrected before a project can compile properly and run.

The  Run Wired Movie button in the Objects window will perform the same action.

Error Window

All syntax errors are reported in this window. Each error will have the following information: the type of error and the line number of the script where the error is located. Double clicking on top of the error will bring up the script pane and the line containing the error.



Export Wired Movie... (Command E)

Exports the compiled project file to disk. After choosing Export Wired Movie..., the movie will compile into a movie file if there are no errors in the movie project's QScripts. The movie name will appear in the Save dialog and the .mov extension will be appended to the file name.

Quit (Command Q)

Quits the application. If the project(s) 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 in 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)

Copy 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)

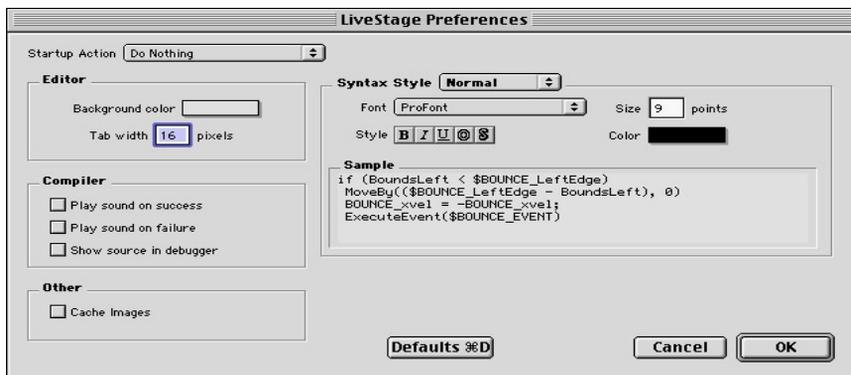
Select all of the objects, or all of the text in the QScript pane.

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. The LiveStage Preferences dialog will appear. To return to the standard settings click on the Defaults button at the bottom of the dialog.



Startup Action

This allows you to customize the appearance of the opening screen. The options are:

Ask What To Do	This option asks if you would like to open an existing project or create a new project. This is the default setting.
Create New Document	This choice will result in LiveStage Professional automatically opening a new untitled Project Window.
Open Existing Document	This option opens a standard file open dialog allowing the user to choose which file to open.
Reopen Last Document	This startup action will automatically open the last saved LiveStage document.
Do Nothing	This choice will tell LiveStage Professional to run but it will not open or create any document. This will provide you with the LiveStage Professional menu bar in which you can then decide what action to perform.

Editor

The editor group gives you a couple of settings that affect all QScript edit fields.

Background Color	A color chip indicating the background color for all QScript edit fields. Click on the color chip to change the background color.
Tab Width	Indicates the size of a tab within a QScript edit field.

Compiler

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.

The “Show source in debugger” checkbox if checked will tell LiveStage Professional to display all executed QScript commands in the Debug Console Window when you are running a movie.

Other

The other group contains a single item which is the Cache Images setting. Enabling the Cache Images setting tells LiveStage Professional to utilize available memory to store images that are being used into memory. This setting will make LiveStage Professional operate faster but will require a lot more memory.

If you disable this setting LiveStage Professional will use significantly less memory but will also run slower since it will have to access your hard disk for all image data.

Syntax Style

This indicates the formatting styles being used in the QScript editors syntax highlighting. From the popup menu select the Syntax Style to be modified: Normal, Reserved, Comments, Strings or Highlighted. After selecting the syntax style, customize it by changing the Font, Style, Size 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 each syntax style and when it is applied:

Normal	The default text settings for any text which is not covered by the other settings.
Reserved	Any keywords that are used within the QScript language.
Comments	Type style used for all comments in your QScripts.
Strings	Any strings in your QScripts will use these settings.
Highlighted	Type style 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.
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.
Size	Indicates the point size to use.
Color	Click on the box and a color picker will appear. Select the desired text color from the palette. The default color is black.

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, Instrument, Effect, Tween, Modifier, Picture and Text tracks. Two additional tracks which are not listed in the menu but can be put into your document by dragging the files on to the tracks tab are External Tracks and Flash Tracks.

Add Sample (Command K)

Creates a new sample in the selected track. The sample created will be added after the last sample in the track.

Path Menu

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

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

Go to End

Go to the end of the movie. This is the time at the end of the last sample.

Go to Previous Frame

Go to the previous frame in the current movie.

Go to Next Frame

Go to the next frame in the current movie.

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 this dialog. The fractions of a second component of the time is based on a range from 0 to the movies time scale which is fixed at 600 in LiveStage Professional. 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 of objects in the Stage Window as well as the display and settings of the Grid. This menu is only active when the Stage Window is active.

Snap to Grid

Align objects to the grid in the Stage when they are dragged around.

Show Grid

Displays / Hides the grid in the Stage.

Set Grid Size...

Brings up the Grid Settings dialog to allow setting the grid size and color.

Align Left Edges

Aligns the left edges of the selected sprites.

Align Right Edges

Aligns the right edges of the selected sprites.

Align Horizontal Centers

Aligns the horizontal centers of the selected sprites.

Align Top Edges

Aligns the top edges of the selected sprites.

Align Bottom Edges

Aligns the bottom edges of the selected sprites.

Align Vertical Centers

Aligns the vertical centers of the selected sprites.

Spread Horizontally

Horizontally spaces the selected sprites evenly.

Spread Vertically

Vertically spaces the selected sprites evenly.

View Menu

The View Menu handles the drawing modes used to display the objects in the Stage Window. It also sets the zoom factor of the Stage. This menu is only active when the Stage Window is active.

Draw Full

Draws the object(s) on the Stage 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

Draws the object(s) as fast as possible on the Stage, 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.

Draw Outline

Toggles between drawing the outline for everything or not. If this is set an outline is drawn around each object on the Stage. 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.

Draw None

This is the quickest draw setting for the Stage as the Stage doesn't draw anything. To make this useful you should turn on the Draw Outline mode which will draw the frames of all the objects in the stage.

50%, 100%, 200%

Sets a standard zoom level for the Stage.

Zoom In / Zoom Out

Sets the zoom factor for the Stage. The Stage can be set to one of the three preset zoom settings or the zoom settings can be incrementally set via the Zoom In (Command **J**) 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 Vertical, Tile, and Zoom.

Stack

Stack 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.

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

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.

Help Menu

Visit the Totally Hip Web Site

This will launch the default web browser and go to the Totally Hip web site.

Get LiveStage Online Support

This will launch the default web browser and go to the support area of the Totally Hip web site. Here information, FAQs and software updates can be found.

Register LiveStage Online

It is important to register LiveStage in order to receive the QuickTime Pro unlock code, 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.

Script Menu

Any AppleScript files you have in the “AppleScript” folder will show up here thus allowing you to execute your AppleScripts within the LiveStage Professional environment. Each script is listed below the *Open Scripts Folder* menu item. Selecting a script will automatically execute it.

Open Scripts Folder

Will open the AppleScript folder in the finder.

Search And Replace

A contributed Script that will do a search and replace of one string with another through all scripts in your project.

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.

CHAPTER 4

Working In The Tracks Tab

Overview	4-1	Track Order	4-8
Creating Tracks	4-1	The Playhead	4-8
Selecting Tracks and Media Samples	4-2	Selection Properties View	4-8
Track Properties	4-3	Media Samples	4-9
Editing Tracks and Media Samples	4-7		

Working In The Tracks Tab

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 in 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.

Overview

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 Professional presents the contents of your document 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 final movie's duration will be when the last sample finishes.

Creating Tracks

When you first start a project the Tracks Tab contains no tracks 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 Professional.

These are:

Sprite Track

A visual track in which sprite samples are stored. Sprites are graphics that may overlay other visual tracks and may have interactivity and animation.

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.
Effect Track	A visible track in which effect samples are stored. Effects are also known as transitions and filters.
Tween Track	A 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.
Modifier Track	A track that produces a predefined sequence of values, as specified by the user, over time. Modifier Tracks are used as a source of data for properties in other tracks or sprites.
Picture Track	A visual track in which picture samples are stored. A picture track contains a series of images that can be displayed over time as in a slide show.
Text Track	A visual track in which text samples are stored. Contains text that may be displayed within the movie or used as a Chapter or HREF (text that references a web page) track.

For additional information on a particular track type, please refer to the chapter that discusses that track type. Please note that the track types that are not mentioned here are supported in LiveStage Professional's external track or external movie feature covered elsewhere in the user manual.

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

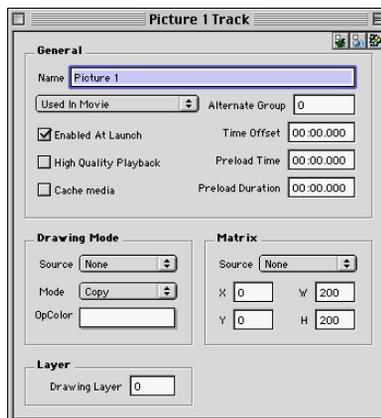
To select a track click on the track header in the Tracks Tab. To select a Media Sample click on the sample in the Tracks Tab timeline. Note that the last item you clicked will be the currently selected item.

This can be confusing since selecting a media sample will show both the track and the media sample as being selected. If you are unsure as to which item is currently selected, look at the Selection Properties Panel at the top left corner of the Tracks Tab. It will show either track information or sample information based on the current selection. If a track is selected the Sprite Track dimensions and its duration will be shown; if a sample has been selected the Start Time and Duration will be shown.

Track Properties

Once you have one or more Tracks in the Tracks Tab you will want to add content to those tracks and modify their various 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. This will display an information window. 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 common track properties are contained in the General group of the track information window. 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.

General Properties

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. This can be changed to a name more meaningful to your project.
Used In Movie	<p>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 those.</p> <p>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.</p> <p>A movie's poster is a still frame, which represents the contents of the movie.</p> <p>The movie setting indicates that the track will be available while the movie is playing. This is the usual setting.</p>
Enabled At Launch	This check box indicates if this 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.
High Quality Playback	This flag 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.
Cache Media	This flag 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.

Alternate Group	If you are using alternate tracks within your movie, this is where you specify the group that the selected track belongs to. Alternate groups are used to support different language types or different levels of quality from an audible or visual track. Specify a numeric value that represents the group to which this track belongs. Each track that belongs to an alternate group will use the same group id.
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.
Preload Time	The starting time of a segment of the track that is to be preloaded. Used in conjunction with the Preload Duration value. When you preload a segment of a track, that track segment is loaded into memory. This provides smoother playback when that part of the track is played. In most cases you would leave the preload time set to zero.
Preload Duration	The duration of a segment of the track that is to be preloaded. If the duration is set to -1, all of the track will be preloaded from that Preload Start Time to the end of the track. All media data in the segment of the track defined by the Preload Start Time and Preload Duration values will be loaded into memory when the movie is to be played.

Tracks that are of a visual type have an extended set of common properties, which are located in the Drawing Mode, Matrix and Layer groups. Some of the track types for which these apply to are Video, Sprite and Picture tracks.

Drawing Mode Properties

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.

Matrix Properties

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 for the track yourself using the properties that follow. Tweens and modifiers can rotate and distort the visual image of the track.
X,Y,W,H	These values specify the visual location of the track and its width and height. Specifying the location only makes sense if there is more than one track.

Layer Properties

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

There are a number of other properties that you may see for different tracks. These properties are covered on a track by track basis later in this manual.

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 Professional, 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 Professional 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 but 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. 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 may 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 does not affect their visual representation in any way, it is used strictly for organization and scripting.

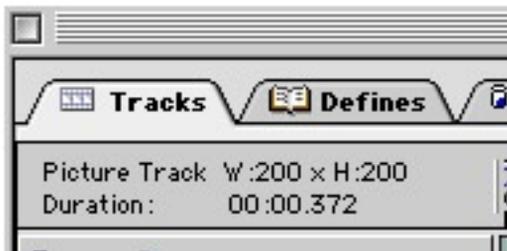
The Playhead

The playhead indicates the point in time in your movie 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.



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. This is fine if the movie you are creating is not too complex, but is inefficient if you want to create multiple effects or groups of unrelated sprites. In this case you will want to create a new sample in your track.

To create a sample, select the track you wish to add the sample to and then choose “Add Sample” from the Tracks menu. This will place a new empty and untitled sample in your track at the end of the last sample.

Note that External Tracks allow very little editing to their contents. One of the 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.

CHAPTER 5

Sprite Tracks

Sprite Track Overview	5-1
Sprite Samples	5-3
Images Tab	5-4
Sprites Tab	5-8

Sprite Tracks

Sprite Track Overview

In this chapter we will discuss sprites and how they can be used in your projects.

Sprite animation produced by LiveStage Professional 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 and where. 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

Sprites may be thought of as actors on a stage with the stage area specified by the boundaries of the Sprite Track.

A sprite contains properties that describe its location and appearance within the QuickTime movie at a given point in time. During the course of an interactive movie a sprite's properties are modified to cause it to change its appearance and location on the stage.

Each sprite in a QuickTime movie refers to an image stored in the 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 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.

Creating a Sprite Track

In order to start working with Sprites you will need to have at least one Sprite Track (see Chapter 4 Working in the Tracks Tab 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. 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 this window there are two new groups of properties that have not been discussed previously. These are the Display and Playback groups.

The Display group contains visual characteristics that are unique to Sprite Tracks. It contains two properties that are described below:

Bit Depth	Indicates what bit depth the Sprite Track will be in. This can be set to either 8 (256 colors) or 16 (thousands 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.
-----------	---

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.
------------------	--

The Playback group contains settings that affect how the Sprite Track will play back in your movie. Here is a description of the properties:

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 present 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 the actual idle frequency may be much less than specified.

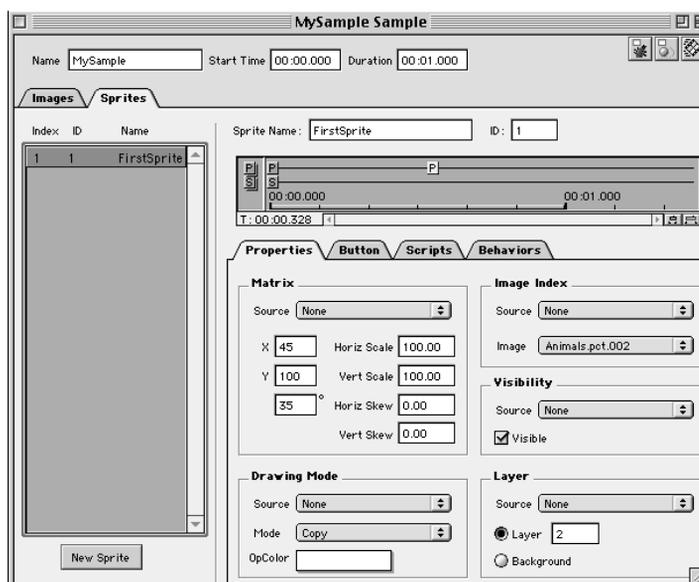
Scale Sprites With Movie

This setting 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.

This property only affects vector images and you must ensure that the vector images are not recompressed.

Sprite Samples

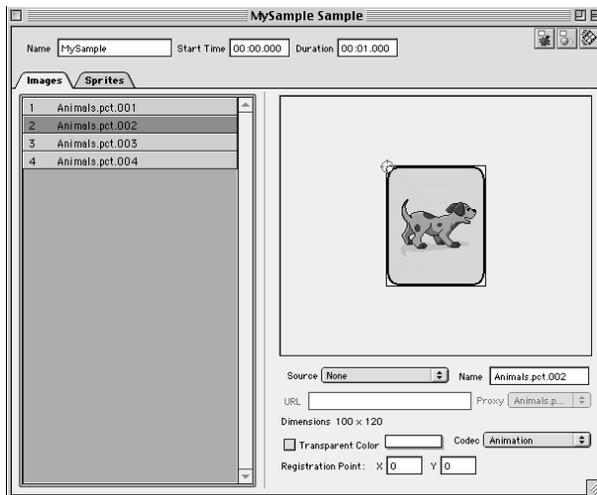
Double clicking on a sample within a Sprite Track will display the Sprite Sample Window.



The Sprite Sample Window contains some general sample information including the sample's name, start time and duration. This window includes two tabs, Images and Sprites. These tabs list the images and sprites that are contained within a Sprite Sample.

Images Tab

Before a sprite can be created, you need to provide one or more images that the sprite can use. All available images in the sprite sample are listed in the Images Tab.



If you are working with a new sprite sample the image list will be empty. To add images into the image list, drag and drop them from either the Library window or the Finder.

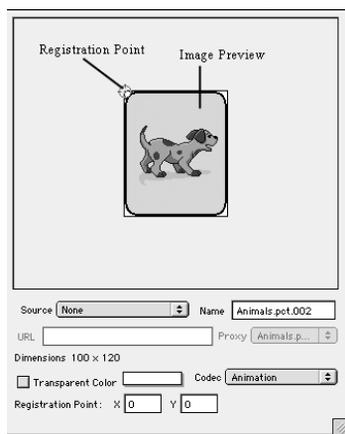
As you place images into the image list you will notice that the images are numbered from 1 to n . These numbers represent the indexes of the images and can be used when you are writing scripts to refer to the image. For example, you may use the following statement:

```
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: "\$ImageOne.qif".

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, its name, places to specify a URL and a proxy image, its dimensions, a check box indicating if a transparent color is being used and what color that is. There is also a specification for the image's registration point, which is shown as an X and Y location. You may also choose to recompress the image with another compressor by selecting an alternate Codec from the popup. To change the graphic used for an image you can drag and drop an image from the Library into the image preview area. This will replace the original image data.



Source Popup Menu

Typically you would use an image for display purposes, however you can replace the image data with that of another image source. The Source property specifies an alternate source for the image's data. This effectively allows you to map videos, effects, text, Flash and a number of other interesting outputs onto the surface of a sprite.

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. 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.

Transparent Color

A transparent color can be specified for an image. Click on the Transparent Color check box to enable transparency and then select the transparent color by clicking on the color swatch. From this point, any sprite that makes use of this image will draw the image transparently and mouse related events will be triggered only for the non transparent areas of the image. Image Transparency only works if you choose the Animation Codec for compressing your image since the other codecs do not work well with transparency.

Registration Point

Registration points are used to set the point in the image that is used to locate the image spatially. Typically an image's registration point is 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. This can be done by changing the X and Y points in the boxes or by option-clicking on the image in the preview area. If a sprite using the image is drawn at 100,100, then the registration point of the image will be placed there. This is useful when you are animating the sprite by changing its image over time so that the animation sequence stays aligned on one point. You will also want to use this if you are rotating the sprite since it will rotate about its registration point.

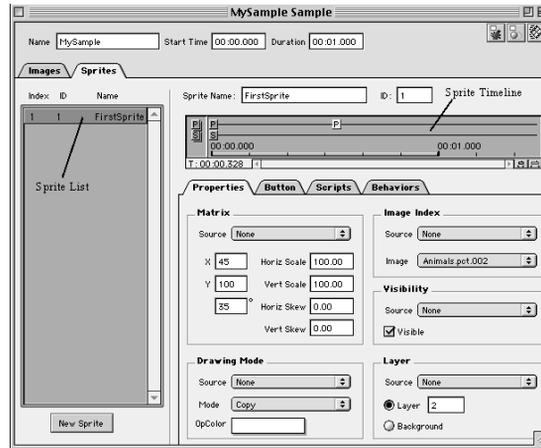
Codec

In order to offer the best possible compression for your QuickTime movies we have added a Codec (Compressor/Decompressor) popup as part of an image's properties. The Codec popup allows you to specify what Compressor/Decompressor you want to use on the image data. Following is a list of the different standard codecs and what they are tailored to.

Don't Recompress	No additional compression is done on the image data. It is used exactly as provided and incorporated into the movie. Use this if your image is already compressed well or if it is a vector graphic.
Graphics	Best for digital and other related images. This compressor provides very good compression with no loss in image quality.

Animation	Specifically designed to handle animated content like cartoon graphics, etc. This can provide very high levels of compression for this kind of content, but is not appropriate for photo type graphics like JPEG images. This codec is also used when you wish to use a transparent color for the image as the other codecs do not properly handle transparency.
JPEG	Standard JPEG style compression which is good for photographs or similar images. Can compress to very small images but there is a sacrifice in image quality.
Sorenson	High quality compressor similar to JPEG but creates smaller files. The Sorenson compressor is known to create unusual visual side effects when used for sprites whose matrices are being manipulated extensively.
There are other Codecs that you may also specify but they are not for image compression. These Codecs allow you to create certain effects like fire within the image.	
Ripple	Generates a ripple effect like that of a rock being thrown into a pool of water. It should be noted that sprites that utilize the ripple codec only affect sprites in the Sprite Track. You must also ensure that the sprite using the ripple codec is on a layer above the sprites that it is to ripple. Changing the OpColor used with the Ripple Codec will affect the amount of ripple used for each color component.
Fire	A fire effect which you can use with alpha blending to create some spectacular effects.
Cloud	Produces cloud effects for use in your movie.

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, Scripts, Button and Behaviors that contain a variety of property settings for each sprite.

Sprite List

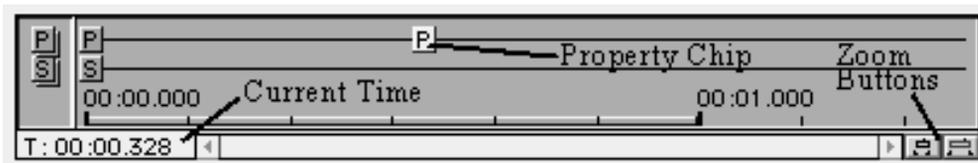
The sprite list gives you a listing of all sprites that are contained within the sprite sample. Clicking on a sprite in this list selects the sprite, at which point you can edit its properties and scripts using the various property sub-tabs in the Sprite Tab. You can also double click on a sprite to open it in its own editing window. This allows you to easily compare two sprites side by side.

On some occasions you may want to put your sprites into a particular sequence to improve organization or to make certain scripting operations easier. Reordering the sprites is a simple matter of dragging and dropping the sprites within the Sprite List.

Note: you should be careful when doing this since if you refer to sprites by index number in any of your scripts, those scripts will have to be updated or they will no longer operate correctly. To avoid this problem you can refer to your sprites by name or id instead of index.

To create new sprites click on the New Sprite button located at the bottom of the list. This will place a new sprite named “Untitled” into the list.

Sprite Timeline



The sprite timeline, at the top of the Sprites Tab, graphically represents all property and script actions that take effect on the sprite over the Sprite Sample’s time frame. The timeline displays the Sprite Sample’s time along with the Sprite Sample duration bar which indicates the sample’s start time and duration.

Along the timeline there are little markers labeled P and S, these are known as the Property Chip and Script Chip respectively. These chips represent changes to either the sprite’s properties or scripts at various points in time. Property chips will be automatically added if you manipulate a sprite in the stage window when the playhead in the Project Window is at a different point in time. Clicking on a Chip will change the time shown in the Project Window to the time for that Chip and will display the associated properties for that time. Command clicking in the 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 tabs 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, then Paste (or choose Duplicate). A copy of the Chip will appear just offset from the Chip originally selected.

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. By holding down the Command key while clicking on a part of the time will change the current time.

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 outlining what those properties do.

Matrix

A matrix specifies how and where a sprite is rendered, including the sprite's X and Y location, angle of rotation, horizontal and vertical scaling and horizontal and vertical skews.

The X and Y positions are specified as integer values that may be either positive or negative. You can also change these values by dragging the sprite in the Stage window.

The angle represents the number of degrees to rotate the sprite. This will affect all content displayed in the sprite.

Horizontal and Vertical Scale values will cause the sprite to grow or shrink. Specifying a value of 100 means that the sprite will display at 100% of its size. If you specify a value of 50 you will get a scale factor of 50% of the sprite's original size.

Horizontal and vertical skew values will cause the sprite to be slanted along either its vertical or horizontal axis.

As an alternate approach to specifying all of these values directly, you can use the Source popup to have the sprite utilize a Tween track to provide data for this setting. Tween tracks can be used to generate a range of Matrix values over time. This will also be discussed in the chapter covering Tween Tracks.

Drawing Mode

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.

You can change the drawing mode by clicking on the Mode popup and choosing one of the many options from the list. Some of these modes will utilize a color value to modify the drawing operation.

For more information on utilization of the Drawing Mode go to the “Drawing Modes Explained” section in Appendix II.

Image Index

The Image Index is where you specify the image data that is to be used by the sprite. Clicking on the Image popup menu shows all of the images available to you in this Sprite Sample. Any of those 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.

You should note that the images are referred to by index number. Because of this, the Image Index property may be controlled by a Tween or Modifier Track. 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.

Visibility

The Visibility property indicates if the sprite is visible. If checked the sprite is visible otherwise it is hidden. Like all of the other properties you may also specify a source track (a Modifier or Tween Track) which will provide values for the Visible property over time. Likewise, 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.

Layer

The layer is a numeric value that specifies a sprite's drawing order within the Sprite Track. 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.

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.

You may use 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.

Button Tab

The button tab contains settings that allow you to easily make the sprite into 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 are grouped into two main groups which handle when the mouse is over the sprite and when it is not. Within these groups are two sub groups, these handle the conditions of the mouse button (left button on Windows machines) being pressed or not.

Each individual grouping includes three properties that modify the behavior of the sprite for the given condition. An outline of these properties and how they effect 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.
-------	---

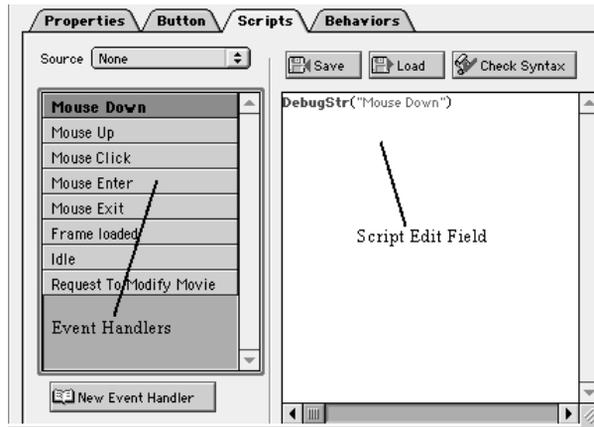
Cursor	Designates which cursor to display when the condition is met. These cursors are the standard cursors available in QuickTime. Choosing none means that the cursor will not be changed by this condition. Following is a list of the cursors:
Open Hand	None Closed Hand Pointing Hand Right Arrow Left Arrow Down Arrow Up Arrow
Variable	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.

Scripts Tab

The Scripts Tab is where all of the scripts for a sprite are located. From this tab you can see what scripts are attached to the current sprite, enter new scripts and create custom Event Handlers.

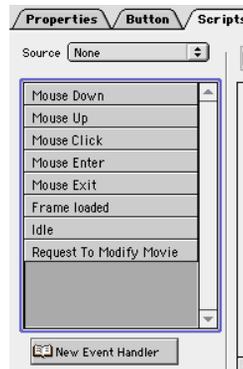
Overview

The layout of the Scripts Tab is divided into two views, the left side displays all of the Event Handlers for the sprite while the right side contains a script editor field that displays any script data for the selected Event Handler.

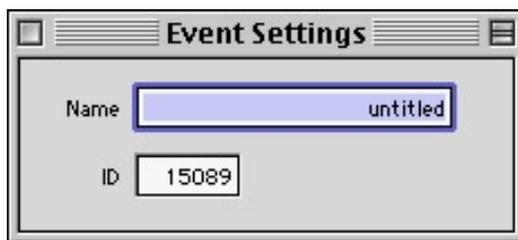


Event Handler List

The Event Handler List lists all Event Handlers for the selected Sprite. All Sprites will have a standard list of Event Handlers that include Mouse Down, Mouse Click, Idle, etc. Event Handlers can be displayed in either normal type or bold type. Bold type is used to show you that there is a script attached to that Event Handler.



In some cases a Sprite may have one or more Custom Event Handlers. These are Event Handlers that you can create using the “New Event Handler” button located below the Event Handler List. When you click on the “New Event Handler” button the following dialog will be displayed:



This dialog allows you to specify a Name and ID for the custom event, the default name is “untitled” and a unique ID is created for you. Once you close the Event Settings dialog a new Custom Event Handler will appear in the Events List. Custom 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 delete the script attached to it. LiveStage Professional will then remove the Custom Event Handler for you.

Event Handlers

Event Handlers are how sprites receive notification about particular actions that are performed on them. For example, a sprite can respond to mouse clicks. The action of clicking the mouse on a sprite triggers the Mouse Click Event Handler. Each Event Handler can have a QScript attached to it which will be executed when the Event Handler is triggered. This allows you to perform simple operations like changing the image of a sprite when the mouse moves over it, or more complex operations like controlling another movie.

To enter a script for an Event Handler, select the event in the Event Handler list. Click in the Script Edit field and enter the QScript commands that are to be associated with this Event Handler.

There are many standard Event Handlers attached to a sprite when it is created, below we list these Event Handlers and describe what triggers them.

Mouse Down	This Event Handler is triggered when the user presses the mouse button down while over the sprite.
Mouse Up	This Event Handler is triggered if the user releases the mouse button after it was first pressed on the sprite.
Mouse Click	The Mouse Click Event Handler is triggered when the user presses the mouse button down over the sprite and then releases the mouse button while still over the sprite.
Mouse Enter	Mouse Enter is triggered when the user moves the mouse over the sprite.
Mouse Exit	Mouse Exit is triggered when the user moves the mouse from inside the sprite to an area outside of the sprite.
Frame Loaded	The Frame Loaded Event Handler is triggered when the sample that contains the sprite is loaded from the movie. This event is unique in that it is not specifically a sprite event. When you are performing operations within this event that refer to the sprite you must identify the sprite using a QScript command. The two commands available to you for this purpose are SpritedNamed and SpriteOfID. You can use the special define “ThisSpriteID” to refer to the current sprite’s ID.
Idle	The Idle Event Handler is triggered whenever the movie gets idle time. The frequency of this event is affected by the value you specify in the Sprite Track’s, Idle Frequency setting.
Request To Modify Movie	This Event Handler is provided for completeness but currently there is little documentation available from Apple on what it does.

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. Now perhaps we want to trigger the same event in a different sprite. For that we would use one of the QScript commands that reference a sprite. Here is an example:

```
SpriteNamed(“Our Test”).ExecuteEvent(2000)
```

Here we have said 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 QScript Reference in Appendix I.

Script Edit Field

On the left side of the Scripts Tab is the Script Edit field and some related controls for the storing and retrieval of QScripts. The Script Edit field becomes active when you have selected an Event Handler from the Event Handler List.

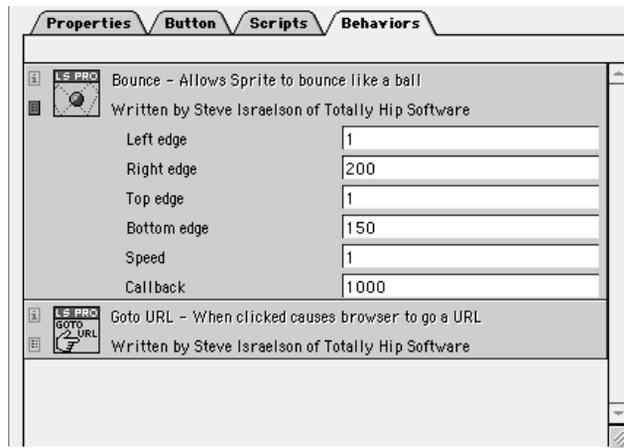
If the Event Handler contains a script, it will be displayed in the Script Edit field. In cases where there is no script attached to the selected Event Handler, the Script Edit field will remain empty. Once you have an Event Handler selected you may enter QScript commands into the Script Edit field.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

Loading, Saving and Verifying QScripts

Across the top of the Script Edit field are three buttons titled Save, Load and Check Syntax. The Save and Load buttons allow you to store and retrieve scripts from disk, thus allowing you to save useful scripts for future use. Check Syntax performs verification on the script in the script editor to ensure that it is syntactically valid and also re-formats the script into standard format.

Behaviors Tab

The Behaviors Tab is used to add Behavior objects to a sprite. Behaviors provide a simple way to add functionality to a sprite without creating your own QScripts. If you are unfamiliar with Behaviors you may want to read the “Behaviors” chapter.

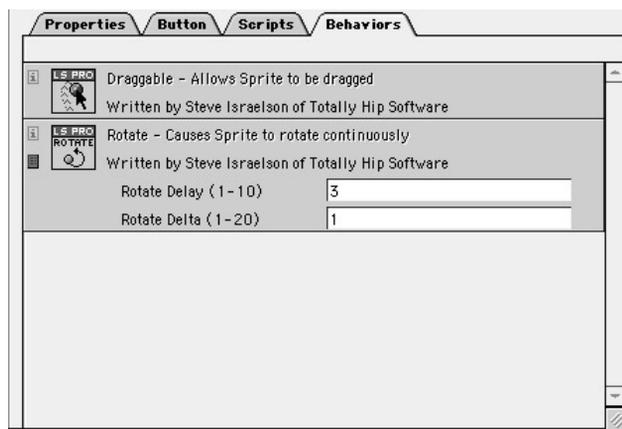


The Behaviors Tab contains a single list that contains any Behaviors that have been added to the current sprite. If no Behaviors have been added to the sprite, the list will contain the text ‘No Behaviors’. When you have one or more Behaviors attached to a sprite you will see the Behavior’s icon, name, short description and author information displayed in the list.

Adding Behaviors

LiveStage Professional comes with a variety of pre-made Behaviors that are located in the Scripts Tab of the Library Window.

Adding a Behavior to a sprite is a simple operation, just drag and drop the Behavior from the Library Window into the Behaviors Tab of the sprite. The new Behavior will be added to the end of the Behavior list. Note that LiveStage Professional will prevent you from adding the same Behavior to a sprite more than once.



The Behavior List

Each Behavior in the Behaviors Tab contains an icon representing the Behavior, a short description of what the Behavior does and some author information. Beside the Behavior’s icon there are two icons that may appear which give you extra information on that Behavior.

The first icon, called the information icon  displays additional information (if any) on the Behavior. This information will tell you what the Behavior does and any extra details you need to know about its operation.

The second icon which may appear beside the Behavior's icon is the parameters icon . The parameters icon expands the view of the Behavior to show any parameters that this Behavior takes. If the behavior does not take any parameters 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 6

Instrument Tracks

Adding Instruments	6-1	Listening to your Instruments	6-4
MIDI Instruments	6-3	Using Instruments	6-5
QuickTime Synthesizer Instruments	6-3	Exotica	6-6
Sampled Instruments	6-4		

Instrument Tracks

Instrument Tracks and Set Controller contain lists of instruments that may be accessed through the QScript command *PlayNote*. Instrument Tracks contain one or more Instrument Samples. These in turn contain instruments that you can use in your movie. These instruments can be standard MIDI, QuickTime Music Synthesizer or sampled sounds.

To create one or more instruments you first need an Instrument Track. After creating one of these tracks (using the Tracks – Create menu) 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

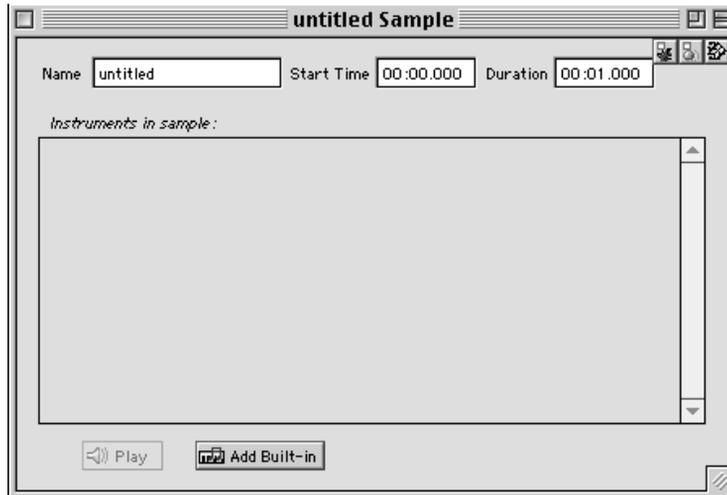
There are several different types of instruments that you can add to an Instrument Sample. These include MIDI instruments, QuickTime synthesizer instruments or sampled instruments.

MIDI 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.

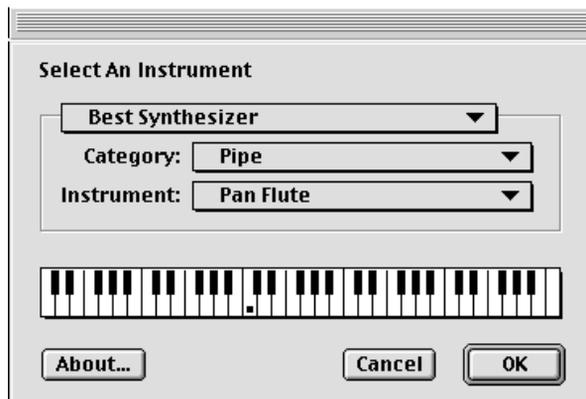
The QuickTime synthesizer instruments are an expanded set of instruments that provide more variety and higher quality playback. The instruments are built into QuickTime itself and use sampled sounds to generate instrument sounds and sound effects.

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 previous two types of instruments, sampled sounds must be stored in their entirety within the movie since they are not built into QuickTime.

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.

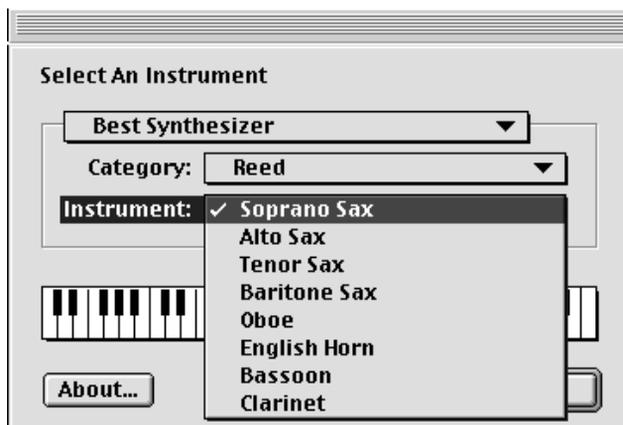


The Select An Instrument dialog is made up of several lists which are, for the most part, self explanatory. The first popup menu indicates the synthesizer you would like to use. By default this is set to Best Synthesizer. Using this setting provides you with all of the standard MIDI instruments. If you choose the second option, which is QuickTime Music Synthesizer, the dialog changes to present you with a number of extra options for selection of QuickTime Synthesizer Instruments.

MIDI Instruments

For standard MIDI instruments the dialog presents a very basic set of lists that allow you to choose the type of instrument you are interested in (the Category list), 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.



QuickTime Synthesizer Instruments

The QuickTime Music Synthesizer provides many more options to choose from compared to the MIDI Synthesizer. When the QuickTime Music Synthesizer is selected the Category list will be updated with the following categories:

No Instrument	No instrument is used.
Use Substitute Instrument	Use a standard MIDI instrument instead of a QuickTime Synthesizer instrument. The substitute instrument is selected in the Substitute Instrument group.
GS Piano & Chromatic Percussion	Piano and percussion style instruments.
GS Strings, Ensemble and Brass	String and brass instruments.
GS Synth	Several synthesizer style instruments.
GS Ethnic & Percussive	Various ethnic instruments.
GS Sound Effects	Many sound effects.

Each of these categories will present you with a different list of instruments in the Instrument list.

Substitute Instruments enable you to use the QuickTime Music Synthesizer instruments but still have a fall back in case that synthesizer is not available on the target system. QuickTime will take the substitute instrument and use it in place of the original instrument in cases such as these. Note that if you do not specify a substitute instrument and the QuickTime Music Synthesizer is unavailable then no sound will be played for that 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 sample 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 the instrument you want to listen to and then click on 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 play simply plays back the sound as it would in any other sound program.

Properties

An instrument track contains the standard General Properties which are discussed in “Chapter 4 - Working In The Tracks Tab”. Instrument Tracks also contain Audio properties.



The audio properties section contains two items, the first being a Source popup menu for the sound volume and the second a Volume edit field. The source popup lists any Tweens or Modifier Tracks that may be used to specify values for the volume property of this track (refer to Modifier Track chapter). 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. Details are given in the QScript section.

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 (refer to the External Movies chapter). 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, LiveStage Professional allows you to use the new inter-movie communication commands to control other movies in order to play a sound track in them.

CHAPTER 7

Effect Tracks

Setting the Effect Properties	7-1	Filter Effects	7-4
Transitions	7-2	Creating a Filter Effect	7-4
Creating a Transition Effect	7-2	Setting the Filter Effect Properties	7-5
Adjusting Transition Effect Timing	7-3	Special Effects	7-6
Viewing Transitions	7-4		

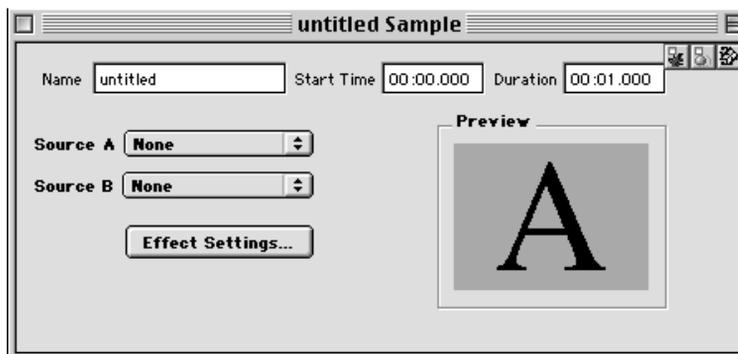
Effect Tracks

Effect Tracks allow you to add special visual effects to your movie. They can be simple effects, like fire, or more complex effects like transitions. The full suite of SMPTE (Society of Motion Picture and Television Engineers) transition effects is available for use as transitions.

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 and then add one Effect Sample to the Effect Track for each effect you want to add.

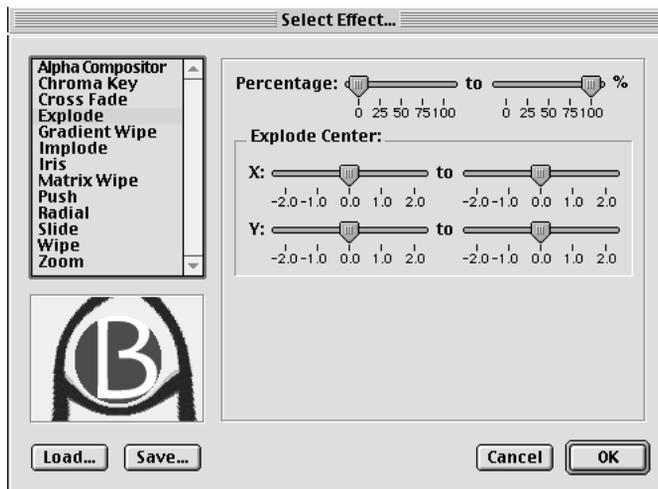
Setting the Effect Properties

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 are then ready to choose the 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.

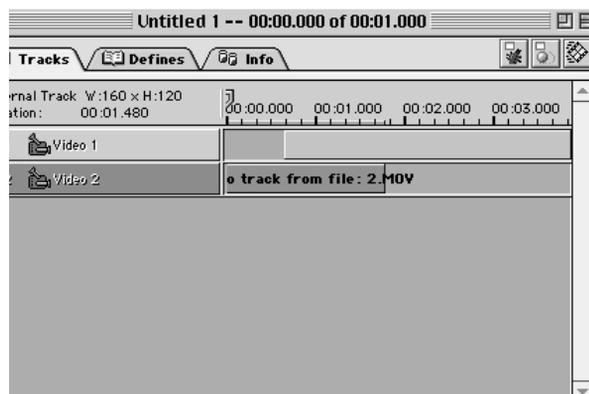
When two source tracks are chosen for an effect, the effect is known as a transition effect. When only one source track is chosen, the effect is known as a filter effect. When no source tracks are chosen the effect is known as a special effect.

Transitions

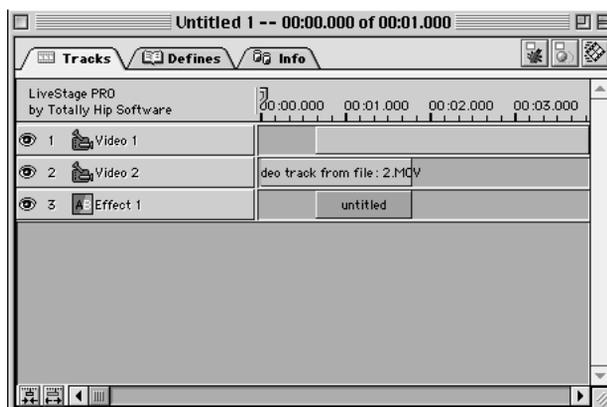
Transitions are most often used for linking two visual elements together smoothly. As an example you may want to fade out one piece of video while fading in a second. Transitions may also be used to create nice transitions between pictures in a Picture Track.

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 (see picture).



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



Now you can 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.

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 your QuickTime movie which you can then play.

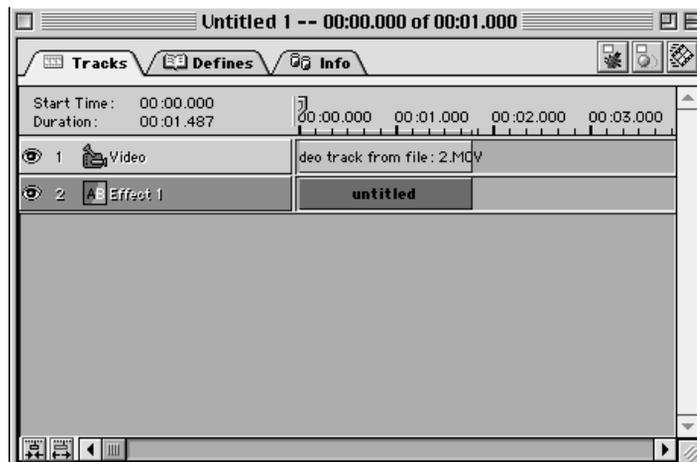
The stage does not provide any preview facility for transitions and effects.

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 provided two visual tracks as the sources, filter effects need only one visual source.



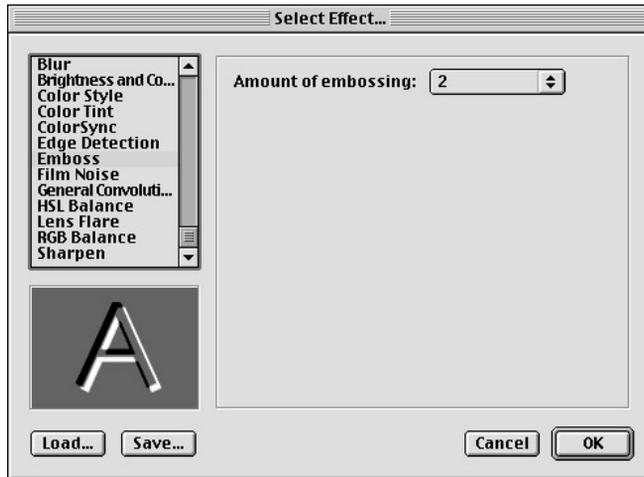
Filter effects are created just like Transition Effects. To create an effect, create an Effect Track in your movie and then move the default sample to the location in which you want the effect to take place. If you want to apply multiple effects over time you can either create multiple Effects Tracks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

Setting the Filter Effect Properties

To set up 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. This results in much smaller file sizes, which are ideal for web delivery.

CHAPTER 8

Tween Tracks

Introduction	8-1	Path to Matrix Translation	8-9
What is a Tween?	8-1	Path to Matrix Rotation	8-9
Tween User Interface	8-1	Path to Matrix Translation and Rotation	8-10
Tween Editor	8-2	Path X to Y	8-10
Name	8-3	Path Y to X	8-10
Start & End Time	8-3	Polygon	8-11
Interpolator	8-4	Spin	8-11
Type	8-6	Graphic Mode Tween	8-11
Tween Types	8-7	VR Angle	8-11
Index / Layer / Flag Tweens	8-7	MultiMatrix Tween	8-12
Time / Event Trigger Tweens	8-8	Tutorial	8-12
Matrix Tweens	8-9		

Tween Tracks

Introduction

Tween Tracks may be one of the most powerful features in LiveStage Professional 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.

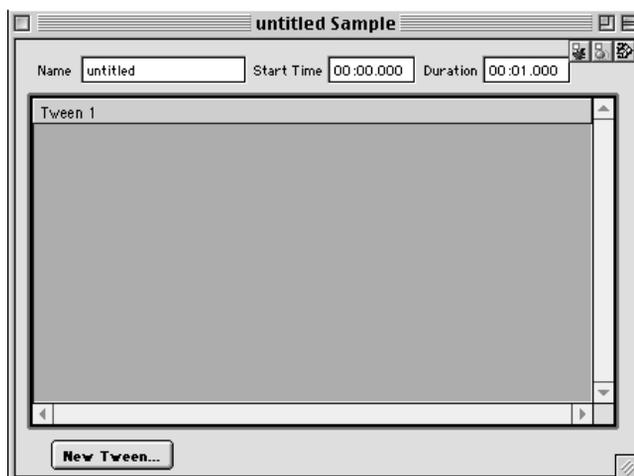
What is a Tween?

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 User Interface

Creating and working with Tweens is a simple process. 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.

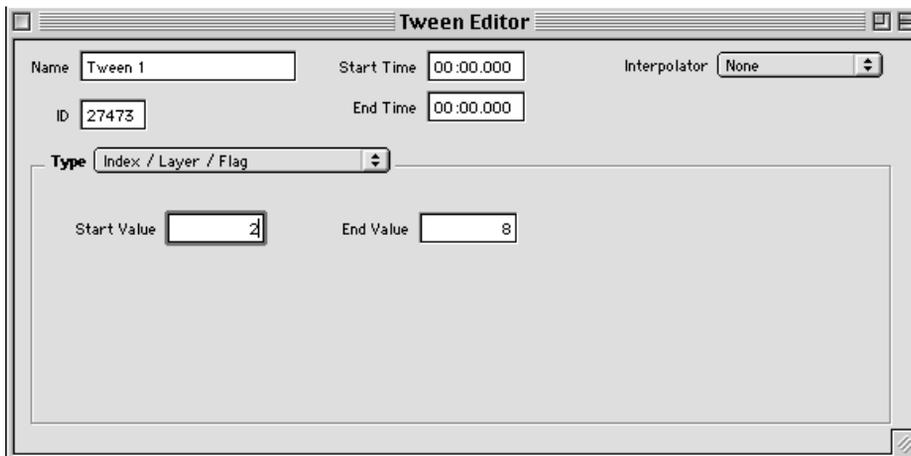
The name field defaults to “untitled” but you can change it to anything you like. The name you enter here will be displayed in the Track List of the Track Tab and is strictly for your own use.

Start Time is the starting time of the Tween Sample. By changing the time value in this field you will 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.

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.

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

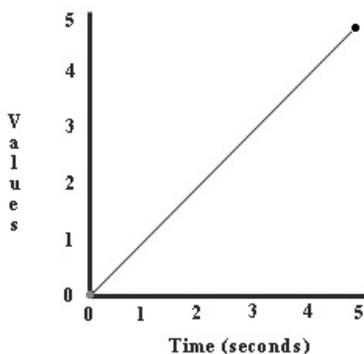
The Name field allows you to enter a meaningful name for this tween. Any name you enter here will be displayed in the Tween Sample's tween list and Source popup menus. Be sure to put in something useful as it will be used as the sole identifier of the tween.

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. Lets 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. This means that 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.

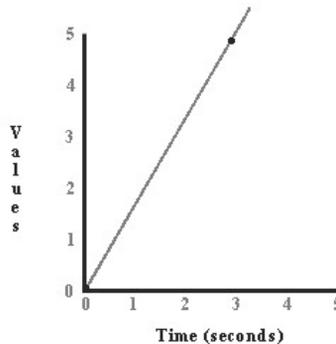
Let's view this tween graphically, the X axis is the elapsed time and the Y axis is the value the tween will generate.



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 3-second time in the tween a value of 3 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 following:

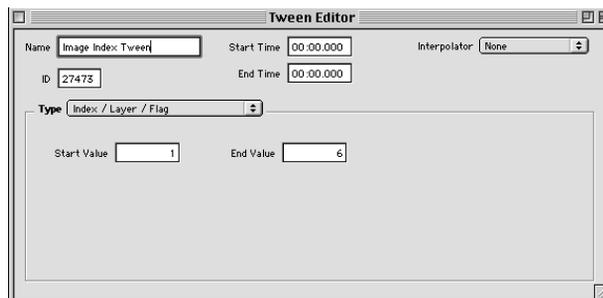


The values produced by this modified tween will be the following. For the start of the tween we will get a value of 0, as the time moves along to the 3 second mark we get a value of 5. Note that this is different than it was with the previous version where we would get a value of 3 at the same time. 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 higher 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".

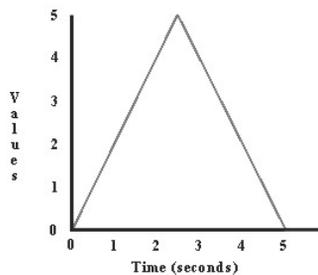
Now we can create a new tween that will be used as an interpolator for the Index tween. Create a new tween in the same tween sample and set its type to "Path X to Y". This tween operates using the passage of time along its X axis and the value range starting at the top of the Y axis and increasing as we go down the Y axis. Click in the path area of the tween to create three points in the shape of a 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 off, 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 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 below).



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.

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. 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 00:01.000 (1 second) then the output from the tween for a time of 0 seconds is 0. The output for a time of 00:00.25 is 1.25, the output for time 00:00.50 is 2.5 and the output for time 00:01.00 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 where you can edit the tween.

You will now be presented with the Tween Editor window. Next we need to set the tween's name; that we will call "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 as well: 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 specifying a matrix. These include stage coordinates, percentage of scaling, percentage of skewing and degrees of rotation.

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. 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.

If you option-click any point a smoothed curve will be drawn between that point and its neighbors; option-clicking again on that point will remove the curve smoothing feature. Finally, hold down the command key or the shift key to select more than one point: they can then be moved in unison.

This matrix data will perform a translation, or movement, such that the object using this Tween will appear to follow the path.

Path to Matrix Rotation

Generates a matrix with rotation information based on the path created in the tween editor window.

The path to matrix rotation tween may be used to rotate a tween. The registration point of the tween is the pivot point. Create the path in the same way as described under Path to Matrix Translation above. The rotation angle will be at a tangent to the current position on the path.

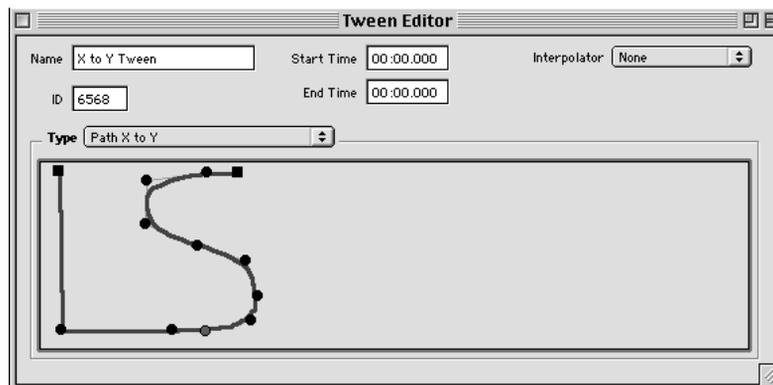
Path to Matrix Translation and Rotation

This tween is a combination of the path to matrix translation and the path to matrix rotation tweens. The matrix generated includes both translation data and rotation data that is derived from the path created in the tween editor window.

Use this tween to cause a sprite to rotate while moving across the stage. The sprite will appear to follow the contours of the path like a car driving on a road. Create the path in the same way as described under Path to Matrix Translation above.

Path X to Y

Creates values based on a path you generate in the tween. The path is treated like a graph with 0,0 being in the top left corner. Movie time is plotted along the x-axis and values returned by the tween are read along the y-axis.



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.

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 you have worked with.

To remove a tween from a MultiMatrix tween, select the tween and press the delete key.

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 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.

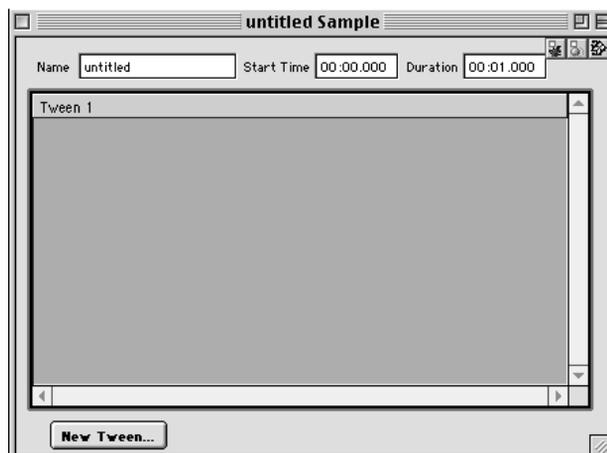
Click on the sample called “untitled”. For easier reference we’ll change the name of the sample to “Bird”. Use the small edit box in the upper left of the Sample Window. You’ll see the title of the window change to “Bird Sample”.

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”. (Of course you can use 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.

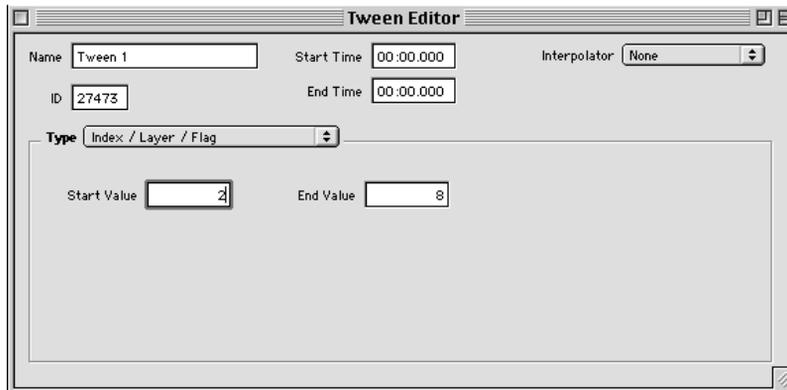
In the Project Window choose “Create/Tween Track” from the Tracks Menu. You will see a new track appear in the Tracks Tab. It’s called “Tween 1” and has a single sample named “untitled”. Double click on the sample and you will get a window titled “untitled Sample”. Change its name to “FirstTweenTrials” 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 “SpinThat Toucan”. 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 9

Modifier Tracks

Properties	9-1
Adding Sequences	9-1
Using Modifiers	9-3
Modifier Types	9-3

Modifier Tracks

Modifier Tracks are one of the non-visual tracks in LiveStage Professional. The sole purpose of a Modifier Track is to produce numeric values over time. Unlike Tween Tracks however, which generate numeric values through an algorithm, Modifier Tracks only provide values from a sequence of numbers specified when you create the Modifier Track.

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 wanted 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.

Properties

The properties for a Modifier Track are no different than the standard set of properties available in all tracks. To obtain more information on what these general properties are please go to the "Working with the Tracks Tab" section which describes these properties and how to use them.

Adding Sequences

To add a new sequence to the Modifier Track, double click on the Modifier Sample which will present you with the Modifier Sample window.



To create a sequence you must specify what type of sequence you want. Note that you may only have one type of sequence in a Modifier Sample (i.e., you cannot mix “Index / Layer / Flag” modifiers and “Event Trigger” modifiers in the same sample). Two other key pieces of information are required, the Rate indicates the frames per second that the sequence will play at. There is also a Repeat field which indicates how many times the sequence should replay.

To see how these two values work together, let's look at some numbers. If we created a sequence that played at a rate of 5 FPS and told it to repeat twice, the sequence would add 10 entries to the Modifier Track. If instead we changed the repeat count to be 10 the sequence would add 50 entries to the track.

One final piece of information is required before you can add the sequence to the track. You need to indicate what values the sequence should provide. The set of values the sequence will use is entered in the Sequence field, each value must be separated by a comma.

Once the data values that you require have been entered into the Sequence field be sure to press the Add button. This adds that sequence to the Modifier Sample.

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 cannot 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.

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.
Index / Layer / Flag	A whole number value which may be used in cases where an Index, Layer specification or Flag value is required.
VR Angle	Generates floating point numbers which can be used for the various panning angles, etc. used 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 10

Picture Tracks

Creating a Picture Track	10-1
Picture Track Properties	10-2
Picture Sample Properties	10-2

Picture Tracks

Picture tracks are a simple track type that make the creation of certain types of movies easier and more efficient. These tracks contain a sequence of one or more images which are displayed over time. This can be quite useful for creating a slide show or for specifying one or more background images to be used in your movie. A picture track may also be used as a source of image data for a sprite.

Creating a Picture Track

The creation of a picture track is no different than the other track types in LiveStage Professional. 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.

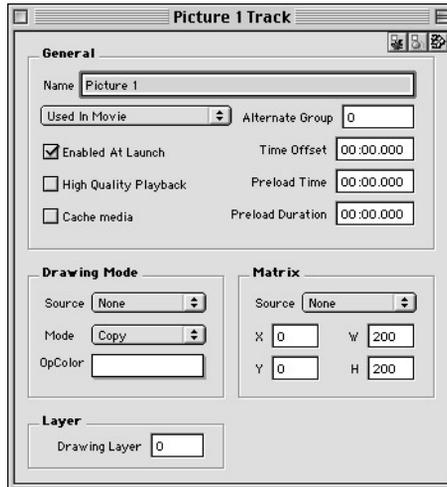
Dragging one or more images from the Library into the Tracks Tab will create a Picture Track. Be sure you drag the images into an empty area of the tab to create a new Picture Track that only contains those images. If you want to add one or more images to a Picture Track you can drag and drop those images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

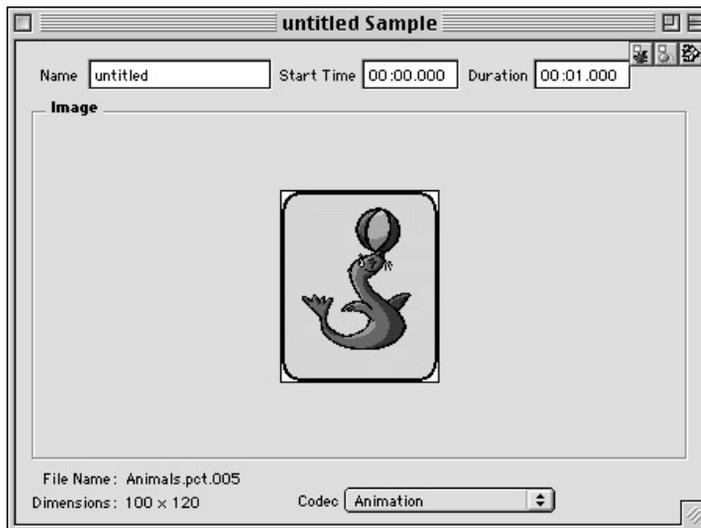


For more information on these properties please refer to “Getting Track Properties” section of Chapter 4.

Picture Sample Properties

Properties for a Picture Sample may be retrieved by double clicking on the Picture Sample in the Tracks Tab. The properties for each Picture Sample are displayed in its own window thus allowing you to compare two or more Picture Samples.

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 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 00:01.00, 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.
- Image** The Image group 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 into the sample by dragging an image from the Library into the Image group.

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 compress the image data. For a description of the compressor types refer to the Compression section in Chapter 5.

CHAPTER 11

Text Tracks

Text Track Properties	11-1	Font, Styles, and Sizes	11-4
Using Text Tracks	11-1	Interactive Text	11-5
Text Tab	11-3	Text Samples	11-5
Hot Spots	11-4	Properties Tab	11-7
Display raw Text	11-4	Chapter Tracks	11-14

A Text Track is a visual track type that displays text in different font and type styles within your QuickTime movie. Text Tracks provide many ways of incorporating text into your movies through various options and techniques.

If you want to put together a movie to act as a navigation bar on a web site, you can put together your button graphics and have these overlaid with text from a text track. The text acts as the label for the button and can be changed very easily. Interesting scrolling effects can also be achieved by turning on a few settings in the text track. You may be thinking, I can do all of this with graphics. While true, the benefit of using text tracks is that the file size is smaller and you can change the labels very easily since it is just like changing text in a word processor.

Text Tracks also work very well for creating information popup windows and scrolling credits.

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 visible track type there is also the standard visual track properties which include the Drawing Mode, Matrix and Layer of the track.

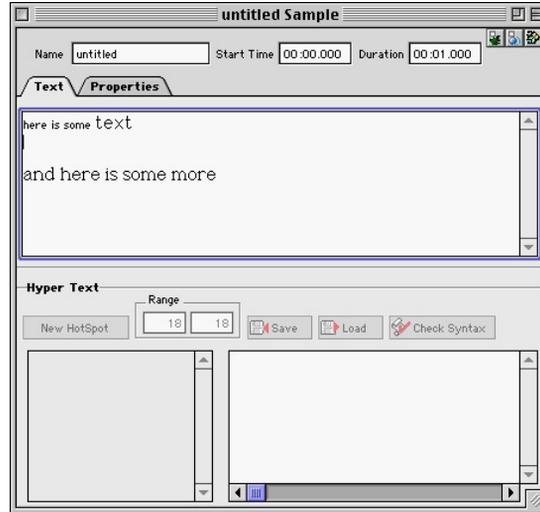
For more information on the standard track properties refer to the “Getting Track Properties” section in Chapter 4.

One property unique to Text Tracks is the Chapter Track property. This 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 in this chapter in the section titled “Chapter Tracks”.

Using Text Tracks

Like all other editable track types in LiveStage Professional, 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 of the settings available for the sample. In the sample window are two tabs used to organize the information in this window.

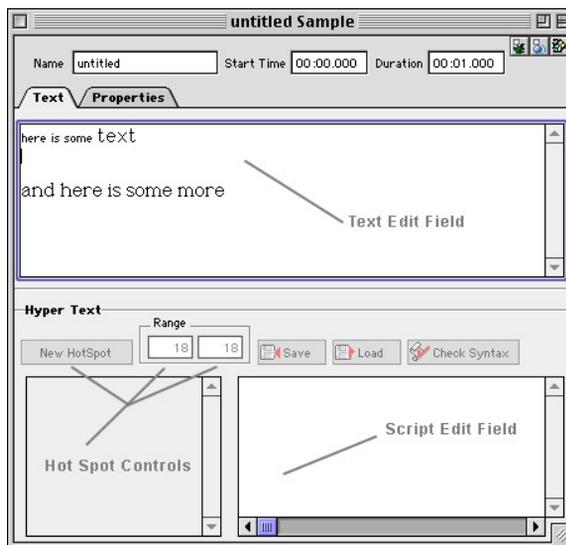


The first is the Text tab which incorporates text editing and hot spot creation and management. The Properties tab, the second tab in the sample window, contains all of the text properties. These primarily affect the display characteristics of the text.

Aside from the two tabs which handle the primary Text Sample data there are also a few properties displayed across the top of the Text Sample window that can be useful. 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 setup 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 controls that you use to enter the text for the Text Sample, set the text style (font, size, type style and color) and specify hot spots.



Any text you want to be displayed in the text sample is entered or copied into the text edit field. In the text edit field you will see the text much like it will appear in your movie, all font styles and hot spots are represented. To change the font settings for any text in the text edit field, select the text in question and then choose your settings from the Font menu. All changes are immediately reflected in the text edit field.

One item that should be mentioned is that any hot spots you have defined in the text edit field will not be displayed in any alternate font styles you have indicated. All hot spots are displayed in blue underlined text just as they are in the QuickTime movie.

The top portion of the tab contains a text edit field where all of the text for the sample is entered. All text displayed in this edit field is displayed using the type style information provided, thus giving you an good representation of how it will look in your movie.

Hot Spots

Below the text edit field is the Hyper Text group that incorporates a variety of components used in making hot spots within your text. Hot Spots are areas of the text that the viewer of the movie can interact with. Each hot spot you create can be scripted using QScript to perform almost any action.

To create a hot spot you simply select some text in the text edit field and click on the “New HotSpot” button. The selected text will now be shown in blue and underlined to indicate it is a hot spot. Note that once a piece of text is part of a hot spot you cannot select it as part of another hot spot.

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 at the end 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 in the track and enter some text into the Text editor window. Running your movie will now display this text into 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.

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 you have entered into a Text Sample, choose the different settings you want to use from the Font menu. The styles you have specified will be displayed for you in the Text Edit window in the Text Sample property window.

Interactive Text

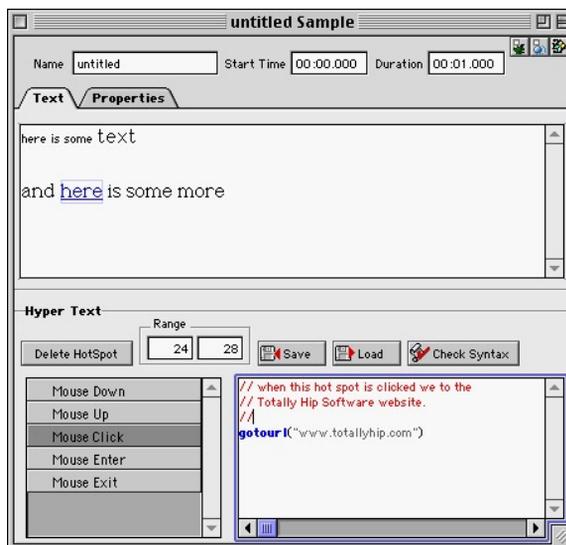
One of the most interesting pieces of functionality that is available through Text Tracks is the interactivity available to you when using hyper text. In a Text Sample you may create hotspots within your text that are associated with QScripts. Each hotspot may be scripted much like a sprite in a Sprite Track can be.

The representation of these hotspots is very similar to hyper links in a web browser window.

To create hot spots within your text select some of the text you have entered in the Text Edit window. In the Hyper Text group you will see a button become active that is titled “New HotSpot”. Clicking on that button creates a hot spot in your text and presents you with some standard Event Handlers with which you may attach QScripts. Just like you script sprites in a Sprite Track you may also enter QScripts into the Event Handlers of a hot spot in a Text Sample.

This hot spot 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.

Text Samples



Text Edit Window

The Text Edit field in the Text Tab contains all of the text information included in the Text Sample. Any font, size, style and color information that is set in the text will be displayed in the Text Edit Field. Also, hotspots that have been created in the text are shown in this window using rectangles around the associated text.

“Hyper Text” Section

The “Hyper Text” section of the Text Tab in the Text Sample window contains all of the elements that are used to manage the hotspots in your text. When no text or hotspots are selected all the items in this section are disabled.

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.

Range

The Range group contains two fields that indicate the starting and ending characters for the current hotspot. If no hotspot is currently selected these fields are disabled.

Event List

The event list displays the Event Handlers available for the currently selected hotspot. 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 Event Handler selects that handler at which point you may enter QScript statements.

Script Edit Window

The Script Edit Window becomes active when you have an Event Handler selected. In the Script Edit Window you enter QScript statements that will be executed when the event occurs.

Save

The Save button becomes active when there is a script being displayed in the Script Edit Window. Clicking on the Save button allows you to save the script to disk for later retrieval.

Load

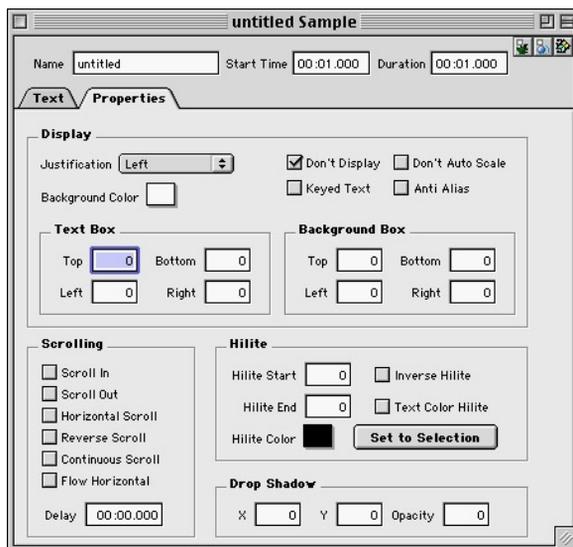
Whenever the Script Edit Window is active the Load button is active. Clicking on the Load button allows you to retrieve a saved QScript from disk and have it loaded into the current Event Handler.

Check Syntax

The Check Syntax button is active whenever there is a script being displayed in the Script Edit Window. Clicking on the Check Syntax button will verify that the currently displayed script is syntactically correct. If there are one or more errors the Error Window is displayed.

Properties Tab

The Properties Tab contains all of the properties for the Text Sample. These consist primarily of display related properties.



Display Group

The Display Group contains all the elements for a text sample that affects the display of the text.

Justification

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

Background 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.

Don't Display

Checking this property will cause the text in the Text Sample to be hidden until either another sample is reached that does not have this property checked or the property is reset during the samples duration.

Keyed Text

The Keyed Text property ensures that the background of the Text Track is rendered transparently.

Don't Auto Scale

Selecting the Don't Auto Scale property will prevent the text sample from being scaled if the movie size is changed. By default this property is unchecked.

Anti Alias

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

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.

Top

The Top field allows you to enter a numeric value that specifies the top edge of the text bounding box. This value is specified in pixels. If all fields in the Text Box group are set to zero then the text is only constrained to the track's dimensions.

Left

The Left field specifies the left edge in pixels of the text bounding box. All text content within the sample will be limited to this left edge. If all fields in the Text Box group are set to zero then the text is only constrained to the track's dimensions.

Bottom

The Bottom field specifies the bottom edge in pixels of the text bounding box. All text content within the sample will be limited to this edge. If all fields in the Text Box group are set to zero then the text is only constrained to the track's dimensions.

Right

The Right field specifies the bottom edge in pixels of the text bounding box. All text content within the sample will be limited to this edge. If all fields in the Text Box group are set to zero then the text is only constrained to the track's dimensions.

Background Box

The Background Box group contains four fields that are used to designate a bounding box for the text and a background color if one is specified. Any text you enter for the sample will be constrained to the area specified by the bounding box coordinates. This rectangle will work as the clipping rectangle unless the rectangle specified in the Text Box property is smaller at which point that rectangle will be used as the clipping rectangle.

Top

The top field allows you to enter a numeric value that specifies the top edge of the text bounding box. This value is specified in pixels. If all fields in the Background Box group are set to zero then the text is only constrained to the track's dimensions.

Left

The Left field specifies the left edge in pixels of the text bounding box. All text content within the sample will be limited to this left edge. If all fields in the Background Box group are set to zero then the text is only constrained to the track's dimensions.

Bottom

The Bottom field specifies the bottom edge in pixels of the text bounding box. All text content within the sample will be limited to this edge. If all fields in the Background Box group are set to zero then the text is only constrained to the track's dimensions.

Right

The Right field specifies the bottom edge in pixels of the text bounding box. All text content within the sample will be limited to this edge. If all fields in the Background Box group are set to zero then the text is only constrained to the track's dimensions.

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 when the sample becomes active.

Scroll Out

The text will scroll out of view when the text sample ends if this property is selected.

Horizontal Scroll

This property alters the default scrolling action to make the text scroll horizontally instead of vertically. In order for this property to have any effect you must have enabled scrolling by selecting either the Scroll In or Scroll Out properties.

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. Selecting 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.

Flow Horizontal

If you are using horizontal scrolling, the text in the sample will by default scroll in one long line. Any line breaks you have entered will be ignored. If you wish to preserve your line formatting when the text is being scrolled check this property.

Delay

The delay field allows you to specify a time value that affects the timing of any scroll operations you have specified. The delay time is a forced duration 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.

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.

Hilite

The Hilite 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.

Hilite Start & End

These two fields indicate the starting and ending positions for the highlight. Values for these fields can be entered manually or you can select the text to be highlighted on the Text tab and then press the “Set to Selection” button. The values for these fields are zero based and increment by one for each character.

Hilite Color

This color chip indicates what color will be used to highlight the text when you are using the highlight options. If no highlight options are selected, this property has no effect.

Inverse Hilite

Setting this property will cause all highlighted text to be displayed inverse. This means that black text is displayed in white, etc.

Text Color Hilite

Setting this will cause the color of the text to change as a highlight.

Drop Shadow

The Drop Shadow group contains a few properties that are used to enable and alter the drop shadow effect. A drop shadow gives the effect that a light source is being directed towards the text and is then creating a shadow. The x and y values specify how many pixels to offset the drop shadow. The opacity specifies how dark the drop shadow is (0 - 255).

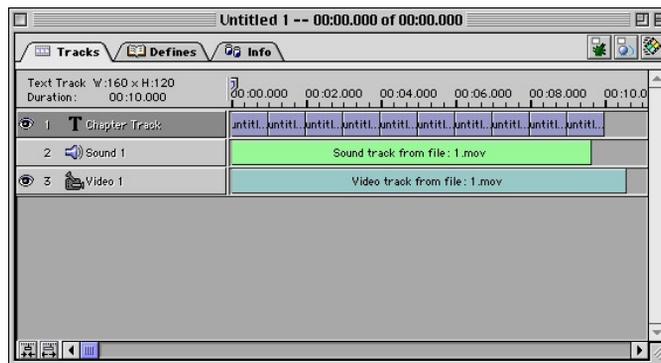
Chapter Tracks

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.

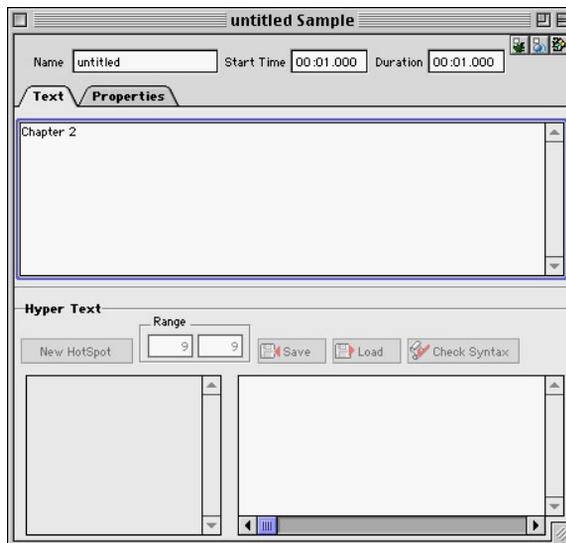


Creating Chapter Tracks

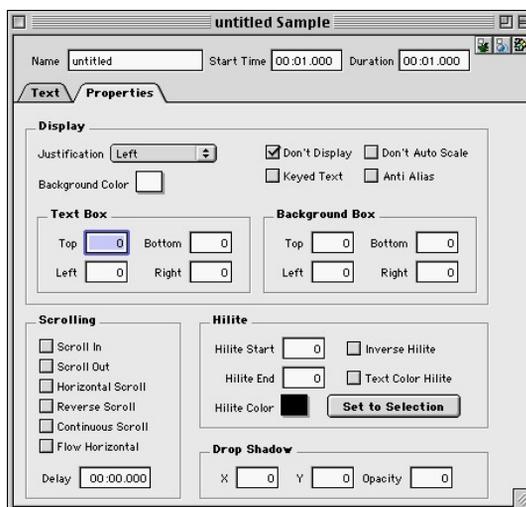
You create a Chapter Track by creating a new Text Track. Each sample in the Text Track represents a single chapter and the start time of the chapter is determined by the sample's start time.



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 we need to modify the properties of the Text Sample. In the Text Sample properties there is a “Don’t Display” option which tells the Text Track to not display the text in this sample. Enable this option for each sample in your Chapter Track, or you can just disable the entire track.



One last change must be made to the Text Track before it can be used as a Chapter Track. By default the track will be displayed in the movie even though we have told it to ignore the contents of each sample. You can address this by setting the Drawing Mode of the Text Track to Transparent, or you can just disable the entire track.

CHAPTER 12

External Movies

Movies in the Library	12-1	Flash Movies	12-3
Using External Movies	12-2	Flash Samples	12-4
External Track Properties	12-2	Script Management	12-5
Copying External Tracks	12-3	QuickTime VR Movies	12-6
Removal of External Tracks	12-3	QuickTime 3D Movies	12-6

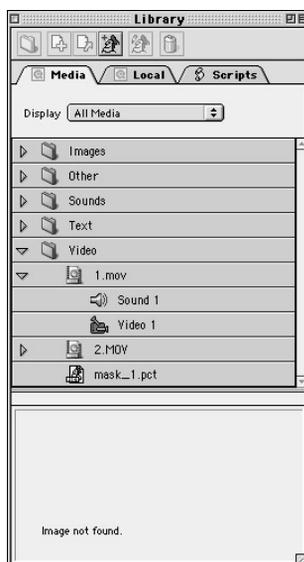
External Movies

External movies are movie files that QuickTime supports which may or may not be created within the LiveStage environment. Just a few of the external movie types that cannot be created in LiveStage are QuickDraw 3D files, MP3 audio, QuickTime VR and Flash files. Even though these tracks may not be created in LiveStage you can still make full use of them within your LiveStage projects. It is also possible to use QuickTime Movie files that were created by LiveStage as external files. LiveStage Professional gives you a project type work environment where you can bring together external movie files into one project and produce a single QuickTime movie.

Movies in the Library

To add an external movie to the Library, drag and drop it from the Finder onto any folder in the Library's Media Tab. The movie you have placed into the Library will then appear in that folder as long as QuickTime can handle that file type. Another way to add External movies into the Library is by clicking on the "Add Items" button to bring up a standard Mac "Choose a File" dialog.

Movie files that are brought into the Library may be expanded to display all tracks in that movie which are found and recognized by QuickTime. To view the tracks contained in an external movie you may either double click on the movie's icon or click on the arrow beside the movie icon.



If an external movie is going to be used as the basis for a project, the movie file can be opened via the Open File menu. This will create a new file and display all of the tracks for that movie in the Tracks Tab. The dragging and dropping of a movie file on top of the LiveStage application icon performs the same functions as well.

Using External Movies

External movies may be incorporated into your project in several ways. In cases where you are going to use an external movie as the basis for your project, you may use the Open item in the File menu. Opening an external movie in LiveStage Professional will add all tracks that QuickTime can process into a new project.

If you wish to add external movies into a project you have created already, you can do so by doing a drag and drop from either the Finder or the Library. Adding an external movie in this way will add all tracks that QuickTime can process into your project. The new tracks will be added to the end of your current list of tracks.

The last approach that may be used when adding External Movies to your project is to extract individual tracks from these movies and add only those parts to your project. This can only be achieved by having the external movie in the Library. When the movie is in the Library, view the tracks in the movie, select the track you want and drag and drop it into the Tracks Tab.

External Track Properties

External Track Samples cannot be manipulated and new Track Samples cannot be added to the track. However, the Track information can be modified by double clicking the track header to bring up the supporting window. In the window it is possible to modify the data for the General Information, Drawing mode, Matrix, and the Layer.

Modifying the time offset in the Track Information window will change the position of the samples for an External Track.

External Tracks that are visible can have their drawing attributes modified in the Drawing mode, Matrix and Layer sections.

Copying External Tracks

All External Tracks can be duplicated by selecting the specified track in the Tracks Tab and either selecting the “Duplicate” menu item from the Edit menu or performing a “Copy” and “Paste” from the Edit menu. Samples can not be duplicated or copied.

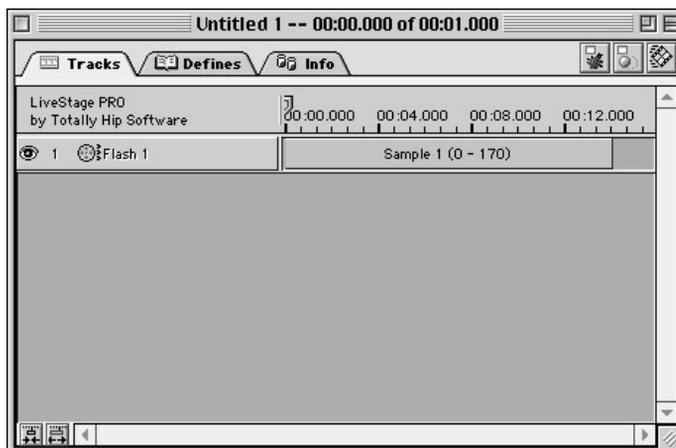
Removal of External Tracks

An External Track may be removed by selecting the track you want to remove and then choosing the “Clear” menu item from the Edit menu or press the delete key after the track has been selected.

Flash Movies

Flash movies are external movie files that are created in Macromedia Flash and imported into LiveStage. A Flash movie that is imported into LiveStage is placed into the Tracks Tab as a single track called a Flash Track.

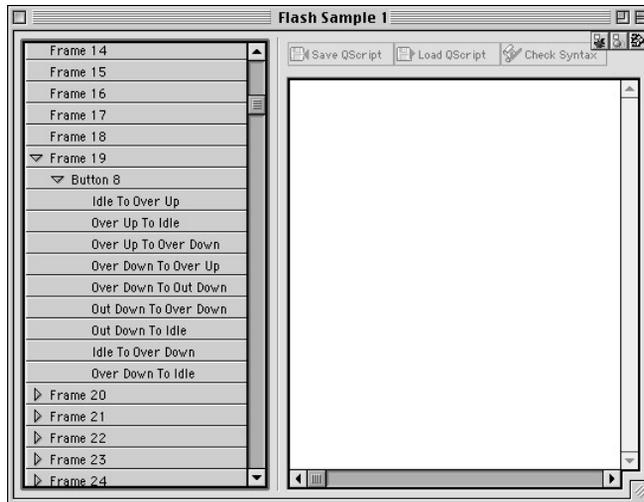
Both the Track Information and the Track Sample for the Flash Track can be edited. However no new samples can be added to the Flash Track.



To modify the Track Information, just double click the Track Header and a window for the Flash Track Information will appear. From that window it is possible to modify the General Information section to change the offset and duration of the track as well as the general track attribute settings. Modifying the Drawing mode, Matrix and the layer sections will change the display attributes of the Flash Track.

Flash Samples

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 sprite tracks except that there is no support for the GlobalVars, LocalVars and SpriteVars keywords. You can only use the MovieVars keyword.



A sample in a Flash Track represents a group of frames from the Flash Track. The contents of the Flash Track are broken down into samples so that you can manage the various elements easier.

By double clicking on one of these samples, you will be presented with the Flash Track Info window which contains a list of the frames contained in that sample. Frames that contain buttons will have the expansion icon beside them, allowing you to expand the view. Expanding a frame's view will present you with a list of one or more buttons, which in turn may also be expanded.

Frames and buttons enable you to enter QScript commands into the script edit window, thus allowing you to script 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.

Buttons are a little different, if you expand a button's view you will see that there are several different handlers that may be scripted. Like sprite handlers, you just click on the handler you want to script, then enter the script into the script edit window.

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

Script Management

There are a couple of script management features you may want to be aware of while working with Flash Tracks. Across the top of the Flash Info window you will see two buttons labeled 'Save QScript' and 'Load QScript'. 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 QScript' 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 Script' button and use the file requestor to locate and open that file. The new script will appear in the current script editor window.

QuickTime VR Movies

QuickTime VR is an extension of the QuickTime technology that allows users to interactively explore and examine three-dimensional virtual worlds. Users can navigate in a virtual world using standard input devices (such as the mouse or keyboard) to change the view displayed by the QuickTime VR movie controller. The images displayed in QuickTime VR movies can be either captured photographically or rendered on a computer using a three-dimensional (3D) graphics package.

A QuickTime VR movie that is added into LiveStage is placed in the Tracks Tab and contains a Video Track, a VR Track and a VR Panorama Track. The Track Sample cannot be manipulated and a new Track Sample cannot be added to any of the tracks.

However, the Track information can be modified by double clicking the Track Header for the specific track to bring up the supporting VR Track window. The Track Information Window for all three tracks contains the General Information section, which allows you to modify starting and duration times for each track as well as various other settings as explained in Chapter 4.

The Video and VR Panorama Tracks have visual sample data and will contain three additional sections in which attributes for the track can be modified. The sections are Drawing Mode, Matrix and Layer.

The Track Samples display a label with the name of the VR movie to indicate which external VR movie the track is related to.

QuickDraw 3D Movies

QuickDraw 3D is a graphics library that is used to create, configure, and render three-dimensional objects. LiveStage does not create QuickDraw 3D files but will display 3D movies.

A QuickDraw 3D movie that is added into LiveStage is placed in the Tracks Tab as a 3D Track. The Track Sample cannot be manipulated and a new track sample cannot be added to any of the 3D Tracks.

Track information can be modified by double clicking the Track Header for the specific 3D Track to bring up the supporting window. In this window it is possible to modify the track attributes in the General Information for the track to modify the offset and duration of the track as well as various other track settings.

CHAPTER 13

Behaviors

Using Behaviors	13-1	Name Section	13-10
Parameters	13-2	Description Section	13-11
Pre-Defined Behaviors	13-3	Parameters Section	13-11
Writing Behaviors	13-9	Event Handlers	13-12
Sections	13-10	Some Guidelines	13-13

Behaviors

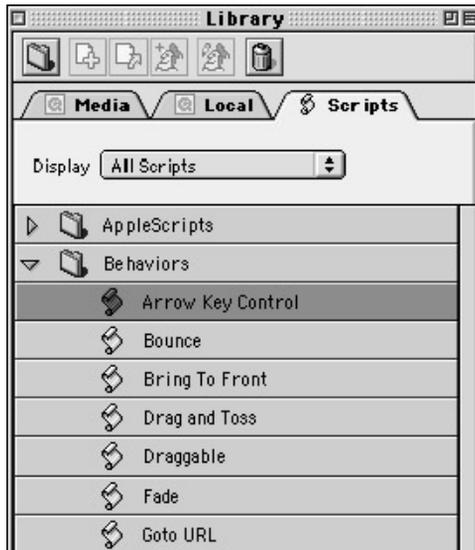
Behaviors are a powerful addition to LiveStage Professional. They allow you to create very rich interactive movies without the need to learn QScript. Behaviors are pre-made scripts that you use by dragging and dropping them into a sprite. A sprite can have one or more behaviors thus giving you a great deal of capability without ever writing a line of QScript.

LiveStage Professional comes with a library of standard behaviors you can use in your own projects. There is a great deal more power available to you however. 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 sprite to be draggable and changes the mouse to a hand cursor and allows you to position the sprite with arrow keys too. 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 the 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, then 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 want to 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 speed. 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 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.

Writing 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 is included in one file.

Take the following behavior as an example:

```
[Name]
Rotate – Causes the Sprite to rotate continuously
Written by Steve Israelson of Totally Hip Software

[Description]
Specify the number of idle calls to skip in Rotate Delay
Specify the angle to rotate by per step in Rotate Delta

[Parameters]
Rotate Delay (1-10), THW_RotateDelay
Rotate Delta (1-20), THW_RotateAngle

[Frame Loaded]                Frame Loaded Handler
SpriteVars THW_rotateTimer
THW_rotateTimer = 0

[Idle]                          Idle Handler
SpriteVars THW_rotateTimer

if (THW_rotateTimer = 0)
    THW_rotateTimer = $THW_RotateDelay
    Rotate($THW_RotateAngle)
endif
THW_rotateTimer = THW_rotateTimer – 1
```

This example is taken from the Rotate Continuous behavior included with LiveStage Professional. You will notice there are many familiar elements to a traditional QScript including the creation of variables and standard QScript keywords. There are a few areas that are different however, you may notice that there are sections defined by [*name*] which outline script groups. These sections direct LiveStage Professional as to where the various QScripts should be placed within the sprite.

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 will put the name of the behavior on the first line followed by the author of the behavior. 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 resize the window to view all of the text.

[Name]

Rotate – Causes Sprite to rotate continuously

Written by Steve Israelson of Totally Hip Software

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 outlines usage of the behavior, parameter descriptions or other special characteristics.

[Description]

Specify the number of idle calls to skip in Rotate Delay

Specify the angle to rotate by per step in Rotate Delta

Parameters Section

This section is used to list the parameters for the behavior. The parameters allow the user 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 behavior only 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.

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 to 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
- Frame Loaded
- Idle
- Request to Modify Movie

You may also include custom events which are labeled 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 like so: `$“Fade In”`

Some Guidelines

Here are a few items and guidelines you need to be aware of when writing your own behaviors. These items are important and will give you a better understanding of how you can use and write better behaviors.

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. Because of this you should make sure you use a unique name for any variables or defines in your Behaviors. Totally Hip Software reserves all variable and define names using “THW_” as a prefix.

You also need to keep in mind that 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.

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.

For Behaviors that use parameters we recommend that you put the parameter limits within parenthesis. For example, in the Fade Behavior 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.

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.

CHAPTER 14

Introduction to QScript

QScript Basics	14-1	Control Statements	14-8
Data	14-1	Target Objects	14-12
Constants	14-2	Putting It All Together	14-14
Variables	14-2	Events	14-14
Properties	14-5	QScript Actions	14-16
Statements	14-6	Continuous Lines	14-16
Assignment Statements	14-6	Line Continuations	14-17

Introduction to QScript

LiveStage Professional provides an extremely powerful user interface for the assembly and basic control of LiveStage movie elements. However, the addition of scripting opens up a whole new dimension (i.e. interactivity) for LiveStage wired movies. QScript is the scripting language used within LiveStage Professional.

In short, scripting is a simplified form of programming. QScript within LiveStage Professional allows elements of a LiveStage 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 wired movies.

QScript is a simple object oriented scripting language created by Totally Hip. 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 can be defined as one or more statements that are executed in sequence.

QScript scripts can be divided into two basic parts:

- 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.

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 with the name
“Total”
LocalVars Count //Declare a variable with the name
“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.

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 LiveStage 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 to a Movie Variable with an ID you simple add the ID you want to the end of the variable declaration. 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 the either use the variable by name or through the SetVariable and GetVariable QScript statements.

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
```

Properties

Properties are data values associated with various movie objects (i.e. sprites, video tracks, sound tracks, etc...). Properties return the value of a particular aspect of a movie object. The following are examples of properties of movie objects:

<code>ThisTrack.TrackEnabled</code>	Returns a TRUE or FALSE value indicating the enabled state of the current track.
<code>SpriteOfID(1).IsVisible</code>	Returns a TRUE or FALSE value indicating TRUE if the sprite with the ID of 1 is visible.
<code>TrackOfIndex(2).TrackVolume</code>	Returns a number between -256 and +256 indicating the volume level of the second track in the movie.

Properties of movie objects are read only. The following example is not a legal statement:

```
ThisTrack.TrackEnabled = TRUE
```

In the preceding example, the `TrackEnabled` property cannot be altered by assignment. To change the `TrackEnabled` property, use the `SetEnabled` action (see QScript Actions).

Properties of LiveStage objects are discussed individually in the QScript Reference. Also, refer to the general discussion of the properties of various objects in earlier chapters.

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 an 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:

```
GlobalVars    Total, CurrentImage
LocalVars    count

count = count + 1    // Increments the local
                    // variable count

Total = Total + count    // Adds the value of the local
                        // variable "count" to the
                        // global variable Total

CurrentImage = ThisSprite.ImageIndex
                    // Assign the current sprite's
                    // image index to the global
                    // variable CurrentImage
```

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
AND

Binary OR
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

Control Statements

Control statements add decision-making capability to scripts. Control statements fall into two groups:

Conditional Statements

Switch Statements

Loop Statements

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.

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
    // script statements executed when neither
boolean_expression
    // is TRUE
ENDIF
```

In each of the above conditional statement formats, different groups of statements are executed depending on the result of the expression controlling the conditional statement.

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. The 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. The following script executes the script statements in the loop statement 5 times:

```
LocalVars count
```

```
count = 0
WHILE (count < 5)
    count = count + 1
ENDWHILE
```

In the second loop statement format (FOR loop), “expression_from” and “expression_to” are numeric expressions (expression 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.

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. Appendix I lists the available Actions available 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.

```
LocalVars count // declare a local counter variable
```

In the preceding example, all text following “//” are 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.

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)// Show 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_Object_Descriptor>.<QScript_Action>
```

The Target Object Descriptor 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)
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)	Specifies the track by supplying the type of the track
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.
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

In the above list, Target Object Descriptors 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 Object Descriptor, a script statement can target any object in any Movie.

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. The 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

When an event is triggered, the script associated with that event is executed. These scripts are called Event Handlers. Each sprite in a wired movie can have event handlers to respond to the events triggered for that sprite. Events are triggered on an individual basis.

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 problem 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 behavior which does all the work for you.

The following is a list of the events that can be triggered in a wired movie:

Frame Loaded	Triggered when the frame of the movie that the event handler is associated with has been loaded. A frame is located anywhere that there is a property or script chip in a Sprite Sample.
Mouse Enter	Triggered when the mouse pointer enters the non-transparent area of the image the sprite is currently displaying.
Mouse Exit	Triggered when the mouse pointer exits the non-transparent area of the image the sprite is currently displaying.
Mouse Down	Triggered when the mouse button is clicked down in the non-transparent area of the image the sprite is currently displaying.
Mouse Up	Triggered when the mouse button is released.
Mouse Click	Triggered when the mouse button is released and the mouse pointer is in the non-transparent area of the image the sprite is currently displaying.
Idle	Triggered at time intervals specified in the Sprite Track properties dialog.
Request To Modify Movie	This event handler is provided for completeness, however no documentation has been provided by Apple.

In addition to the above Standard Event Handlers, Custom Event Handlers can be created. Custom Event Handlers provide the ability to modularize your 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.

CHAPTER 15

QuickTime VR Movies

Introduction	15-1	VR Track Samples	15-3
What is a QuickTime VR Movie?	15-1	General Properties	15-4
Using VR Movies	15-2	Scripting VR Tracks	15-7
VR Track Properties	15-2	Custom Cursors	15-7

Introduction

This chapter covers QuickTime VR movies, what they are and how they are incorporated into LiveStage Professional 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 3D 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.

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.

Using VR Movies

Now that you know what a QuickTime VR movie is and what it can do, you need to understand how it is incorporated into the LiveStage Professional environment and projects.

Utilizing QuickTime VR Movies in LiveStage Professional can be achieved using a drag and drop operation from either the Library Window or the Finder into the Tracks Tab. Adding a QuickTime VR movie into a LiveStage Professional project creates a single VR track in the Tracks Tab labeled with the movie name. This point is significant since a QuickTime VR movie is always made up of several tracks. LiveStage Professional displays the VR movie as one track so working with a VR movie is simpler and less prone to creating movies that don't work properly.

VR Track Properties

VR Tracks have a basic set of properties that are similar to most visual track types. A VR Track contains properties for its Drawing Mode and Matrix. There are however a few properties that are specific to VR Tracks.

The VR Track group located at the top of the VR Track properties window contains fields that indicate the Name of the VR Track, the Idle Frequency, the Time Offset of the track and the Duration of the track.

Following is a description of each property in the VR Track group.

Name	This is the name of the VR Track that is displayed in the Tracks Tab.
Idle Frequency	Indicates how often the VR Track is to receive idle time. This value is specified in ticks, which are 1/60 second. The value you specify here is used as a recommendation to QuickTime, if QuickTime is busy performing other tasks you are not guaranteed to receive idle time this frequently. Specifying a 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.

Time Offset	The Time Offset property allows you to offset all samples in the VR Track by a specified amount of time. Since VR Track samples cannot be moved within the Tracks Tab you may use this field to align the contents of a VR Track with other content in your project.
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.

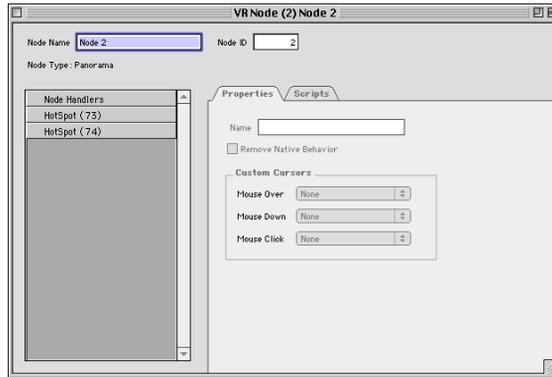
VR Tracks also have a Matrix group just like all other visual track types in LiveStage Professional. What makes the Matrix group special in VR Tracks is that you can specify negative values for the width and height fields. Using negative values in these fields will invert the VR Track either horizontally, 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 hot spots along the left-hand side and the properties associated with the current hot spot on the right hand side.

A VR Sample window contains a number of elements that allow you to setup various properties and script event handlers. Lets take a look at the VR Sample window and the components in it.



General Properties

There are a couple of properties presented across the top of the VR Sample Window, which allow you to set/view standard sample information. The following are the lists of fields described in detail.

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)

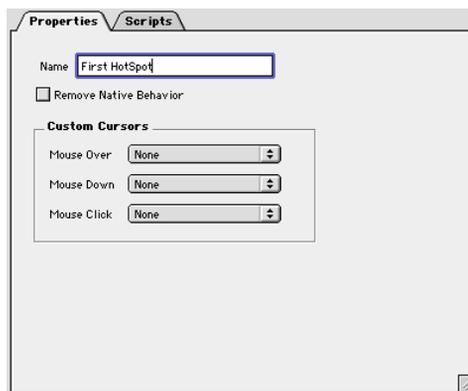
Hotspot list/Node handlers

What are hot spots? Hot spots are areas in a QuickTime VR movie that enable the user to interact with the movie. LiveStage Professional does not create hot spots in a QuickTime VR movie, your VR creation software must create these. Traditionally hot spots are used for navigating a QuickTime VR movie or jumping to web locations by linking to a URL. LiveStage Professional however offers you a lot more freedom in the type of functionality you can associate with a hot spot.

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

Properties Tab

The Properties tab is used to set the values for various properties of a hot spot. This tab becomes active whenever there is a hot spot selected in the hot spot 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 Hot Spot you are currently editing. This name is displayed in the QuickTime Player when the mouse is over the hot spot. The name is also displayed in the VR Sample window's hot spot list.
Remove Native Behavior	When this property is checked any actions defined for the hot spot in the VR movie are disabled. This allows you to reprogram the hot spot 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 hot spot's functionality then check this setting.

Custom Cursors

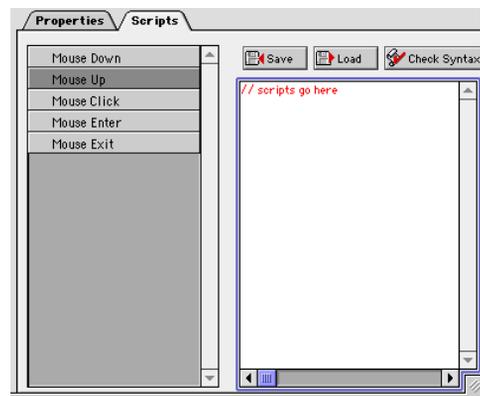
This is a group of properties that indicate what cursors are to be displayed when the mouse is in various states while over a hot spot. Each state (listed below) contains a popup menu containing all of the available custom cursors. To find out how you create custom cursors in LiveStage Professional,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

The available mouse states for custom cursors are:

- Mouse Over The mouse is currently over the hot spot but the mouse button is not being pressed.
- Mouse Down The mouse is currently over the hot spot and the mouse button is being pressed.
- Mouse Click The mouse is currently over the hot spot and has been clicked.

Scripts Tab

The Scripts Tab is the location where you enter any QScript statements that are to be associated with the currently selected Hot Spot 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.



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 Hot Spot selected will present you with a longer list of event handlers that include Mouse Down, Mouse Click, etc.

Once an event handler has been selected you may enter your script statements into the script editing window. Like all other script editing windows in LiveStage Professional, there are a couple buttons above the window that allow you to load and save your QScripts.

Scripting VR Tracks

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

In VR Tracks the use of variables other than MovieVars is not supported. This includes the use of the LocalVars, SpriteVars and GlobalVars QScript statements. Using the other variable declaration statements will give you undefined results and thus should be avoided.

Custom Cursors

A Custom Cursor is an image used to replace the default mouse pointer graphic. QuickTime uses a standard set of mouse pointer graphics for representing various actions within a VR movie. LiveStage Professional lets you provide your own custom cursors for some of these situations thus allowing you to customize the experience.

Adding custom cursors into LiveStage Professional 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 Professional 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 on, otherwise LiveStage Professional will not recognize them.

Glossary

Term	Description
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. This Sprite media sample is imported from an external Video 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 the specified 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 the 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. So, given the values 0 and 1, all values between 0 and 1 are interpolated values.

Interpolator	<p>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 type volume envelope for sounds, or to make sprites move in more realistic ways by making them start slowly, accelerate, and then stop slowly.</p>
Layer	<p>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.</p>
Linear	<p>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. To non-linearly 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.</p>
Matrix	<p>See transformation matrix or the sprite's matrix (using a path tween).</p>
Media	<p>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.</p>
Media Sample	<p>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.</p>

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 UI element displayed at the bottom of the movie that provides a set 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.
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 will 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.
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.
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 they are derived mathematically.
Wired Movie	Wired movies are movies that have tracks that contain scripts that run in response to a users actions.
Zoom in/out button	These buttons allow the user to change the scale that the timeline is drawn at. Pressing zoom in will show more detail, allowing precise placements of samples.