Customizing Final Cut Studio Blu-ray Disc™ Templates

October 5, 2009

Introduction

This document describes how to customize Final Cut Studio disc publishing templates in order to create a unique Blu-ray Disc™ experience beyond what is provided by the templates included with Final Cut Studio.

Audience

This document is intended for Final Cut Studio users, developers or support personnel that have some experience with development (e.g. HTML editing), are comfortable with editing Property List (XML) files, and have some understanding of the concept of video encoding, e.g. that MPEG-2 and H.264 are different video compression standards.

An Overview of Disc Publishing

A new feature in Final Cut Studio is the “Share” feature. Users can now burn DVDs and Blu-ray discs from within Final Cut Pro or Motion. Compressor’s ‘Create Blu-ray’ batch template also provides that feature in Compressor. When using this feature, the user chooses a disc template and provides a title and optional graphic files such as logo, title, and background images. Compressor then encodes the video to a Blu-ray Disc™ compliant format. After the encode is finished, the disc structure is created and can be burned to disc using a Blu-ray disc burner attached to the computer. For more information on Compressor or the Share feature, go to http://www.documentation.apple.com/ and see the Compressor User Manual and the Final Cut Pro User Manual about the Share feature.

Caution, Before You Begin

Please note that modifications to the settings described herein could result in a disc that is not Blu-ray Disc™ compliant. Successful playback on a particular Blu-ray disc player does not guarantee that your disc complies with the standard or will play on all players.

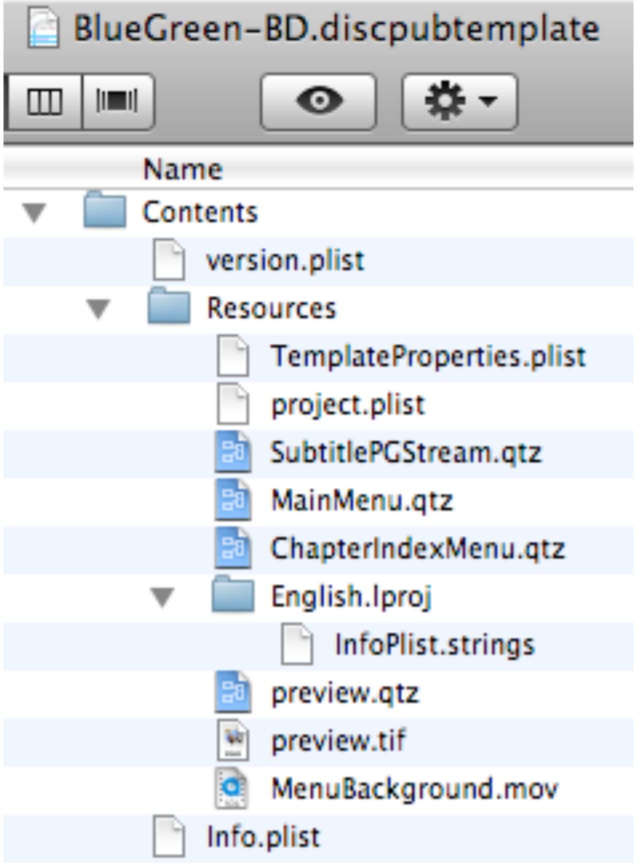
Template Bundle Contents

Overview

The templates for Disc Publishing are located on your hard drive at
/Library/Application Support/Final Cut Studio/DiscPublishing/Templates

Each template is a bundle with file extension ".discpubtemplate". The Blu-ray templates have names ending with "-BD", for example, "BlueGreen-BD.discpubtemplate".
In Finder, you can view the contents of the bundle by Control-Clicking the bundle name and choose "Show package contents" from the context menu.

Inside a Blu-ray disc template bundle, you will see the standard bundle folders such as Contents, Resources, and (for example) English.lproj.



BlueGreen-BD.discpubtemplate

Name

- Contents
  - version.plist
  - Resources
    - TemplateProperties.plist
    - project.plist
    - SubtitlePGStream.qtz
    - MainMenu.qtz
    - ChapterIndexMenu.qtz
    - English.lproj
      - InfoPlist.strings
    - preview.qtz
    - preview.tif
    - MenuBackground.mov
  - Info.plist

Files in the Template Bundle

Info.plist and version.plist

These are standard bundle files that define various generic properties about the bundle such as name and version. For more information, see the developer documentation on bundles at http://developer.apple.com/documentation/CoreFoundation/Conceptual/CFBundles/CFBundles.html

TemplateProperties.plist

This is the "top level" file that describes the overall template properties. It is a standard property list (plist) file. Compressor uses this file to present the template to the user, even before the template is applied to create a disc. It determines such things as the template name and disc format. It also contains a reference to the project file (project.plist). For more information on property lists, see http://developer.apple.com/documentation/Cocoa/Conceptual/PropertyLists/Introduction/Introduction.html

project.plist

This file defines the structure of the discs created by this template. It defines user navigation, graphical, audio and video content. It is also a standard property list file. Some project elements refer to other files in the bundle by filename. See the Appendix for more information about project.plist.

preview.tif

This is a thumbnail image in TIFF format that is presented to the user as a preview when choosing a template.

preview.qtz

This file contains a recipe for rendering a video preview of the template. It is a Quartz Composition file. It can be configured to render the main menu or the chapter index menu. In fact, it references the other Quartz Composition files used when creating a disc and merely changes some properties specifically for preview. For more about Quartz Compositions, see the developer documentation on Quartz Composer at http://developer.apple.com/graphicsimaging/quartzcomposer/.

MainMenu.qtz

This file is used to render the main menu. It is a Quartz Composition file. This file is referenced by name in the project.plist file as well as by preview.qtz.

ChapterIndexMenu.qtz

This file is used to render the chapter index menu. It is a Quartz Composition file. This file is also referenced by name in the project.plist file and preview.qtz.

MenuBackground.mov or MenuBackground.png

These are the menu background assets. Menu templates with video background (Blue Green, Chamber, On Location, and Street) use the QuickTime movie (.mov) while the still background templates use the PNG file. These files are referenced by name in the asset section of the project.plist file.

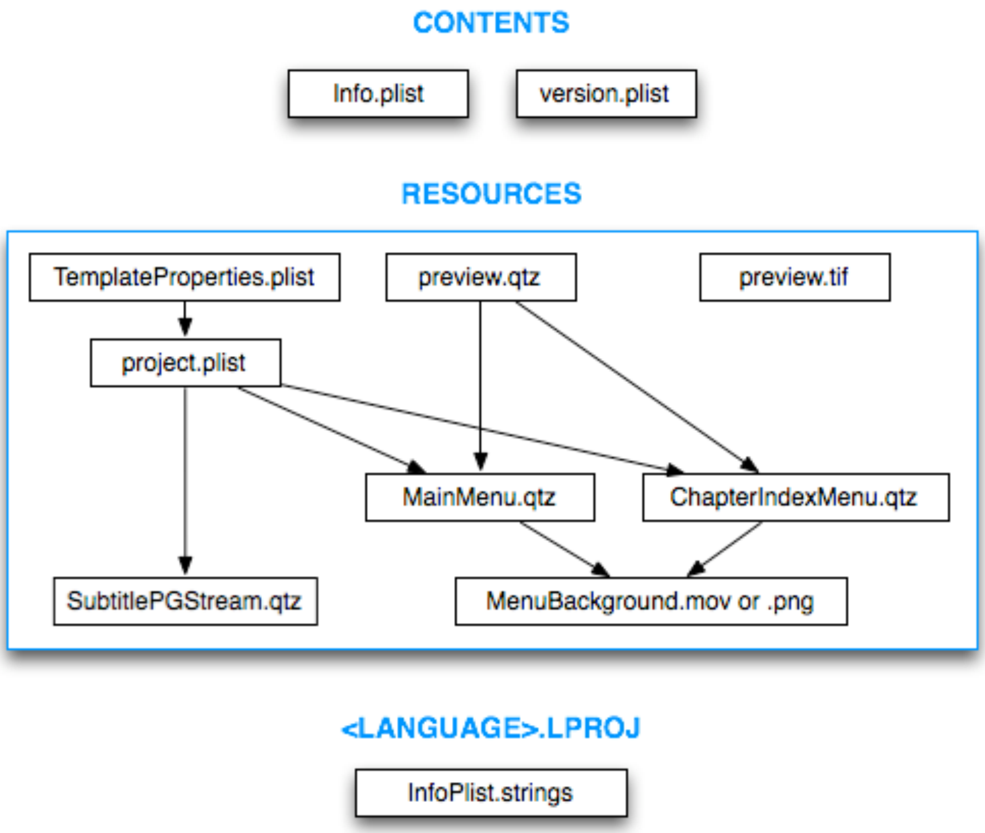
SubtitlePGStream.qtz

This file is used by the Blu-ray disc player to render subtitles (when "Use Chapter Marker Text as Subtitles" is selected by the user). It is a Quartz Composition file. It is also referenced by name in the project.plist file.

InfoPlist.strings

This is also a standard bundle file containing localized strings. The only noteworthy string in this file is the template name, indicated by "DSPPTemplateDescription". It is the string displayed to the user when selecting a template in the Compressor UI or Share feature.

The following is a diagram of the folders (in blue) and files, with arrows indicating file references (where the contents of one file refers to another):



Examples

This section describes specific modifications you can make based on the provided templates to create more custom templates. You can duplicate existing templates and make the necessary modifications to the duplicates. Duplicating an Apple-provided template and then adjusting the duplicate to suit your needs is the most convenient way to create custom templates. Don't change the original templates provided with Final Cut Studio!

Making a copy of an existing template is more than just duplicating via Finder. You have to differentiate the template from other templates in the same folder otherwise you won't be able to tell which one is the one you want in the Compressor or Share UI. To make these modifications, it is best to use Property List Editor, which is available on your hard drive at /Developer/Applications/Utilities if you have the Xcode developer tools installed. Otherwise, you can use TextEdit or your favorite text editor.

Copy an Existing Template

1. Duplicate one of the existing templates. You'll find the templates in /Library/Application Support/Final Cut Studio/DiscPublishing/Templates. In Finder, select an existing template and press Command-D. You'll get a new bundle named "xxx copy.discpubtemplate".
2. Rename the new bundle to a new name of your choice, for example "nice.discpubtemplate". Make sure the file extension is ".discpubtemplate".
3. Show the package contents of the bundle so you can edit the properties. In Finder, Control-Click the new bundle (nice.discpubtemplate) and choose "Show Package Contents".
4. Navigate to the Contents folder of the new bundle. You'll see three things: Info.plist, Resources, and version.plist. You'll need to edit the Info.plist file. If you have Xcode developer tools installed, you can just double-click it, otherwise open it with TextEdit.
5. Open Info.plist in your choice of editor. Change the <string> value for CFBundleIdentifier so that it has a new name, e.g. "com.company.whatever.nice". Change the <string> value for CFBundleName to a new name, e.g. "Nice". You can also change the CFBundleGetInfoString as well. Save the Info.plist file.
6. In Finder, navigate to the Resources folder of the bundle. Edit the TemplateProperties.plist file in your choice of editor. Change the value of AbstractName to your new name, e.g. "Nice". Save the TemplateProperties.plist file.
7. In Finder, navigate to the English.lproj (or whatever your native language folder is). Edit the InfoPlist.strings file in Xcode or your favorite editor. This is a UTF-16 file, but TextEdit should work as well. Change the string value (right-hand side) of DSPPTemplateDescription to your new name, e.g. "Nice". Save the InfoPlist.strings file.
8. You're done. Now you have a copy of an existing template and the new name is "Nice". You should now be able to see it in the Compressor Job Action tab when you choose your disc template. The next step is to customize your new template.

Change the Menu Background Movie

Compressor allows you to replace the background movie with a still image, but you may want to replace the background with a movie of your own instead. To have your new template show a different movie in the background of the main menu and chapter index menu, you'll need to start with one of the templates that uses a movie as the background, such as Blue Green, Chamber, On Location, or Street. Copy one of those templates as described in the previous section.

Let's say your new background movie file is called "mybackground.mov". This movie must have dimensions of 1920x1080 and a frame rate of 24.0 fps (not 23.98) to match the template. Take note of the duration of the movie. Your menu loop duration cannot be longer than the duration of the movie.

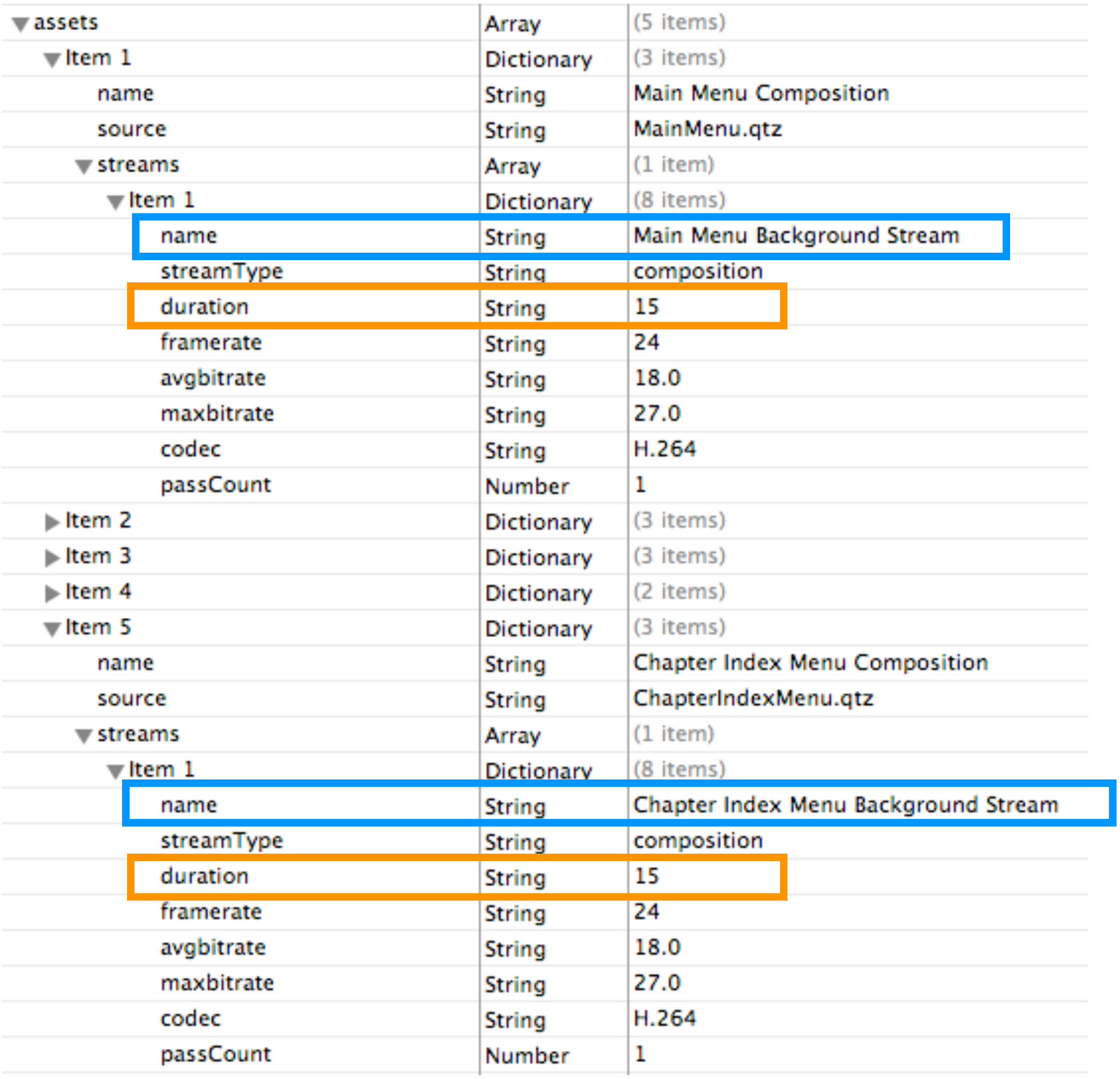
In Finder, view the package contents of your new disc template bundle, if it's not already open. Navigate to the Contents/Resources folder. In that folder you should see a movie file called "MenuBackground.mov". Again, don't modify the original stock templates... this is only going to work if this is a copy. Remove MenuBackground.mov (move it to the trash) and copy your new movie file into the Resources folder, and rename it to "MenuBackground.mov". Basically you've just replaced the original background movie. If the duration is different than the original, you'll need to adjust some values in the project.plist file, as described below.

Adjusting Menu Background Loop Duration

There are two menus in the template: main menu and chapter index menu. For each of those, you'll have to adjust the stream duration and the clip duration. The stream duration is the overall duration of the movie file. The clip duration is how long you want the menu background to play before looping. The clip duration cannot exceed the stream duration.

To change the stream duration, open the project.plist file (in your .discpubtemplate bundle under Contents/Resources) with Property List Editor, TextEdit or your favorite editor.

Find the item named "assets", as shown below:

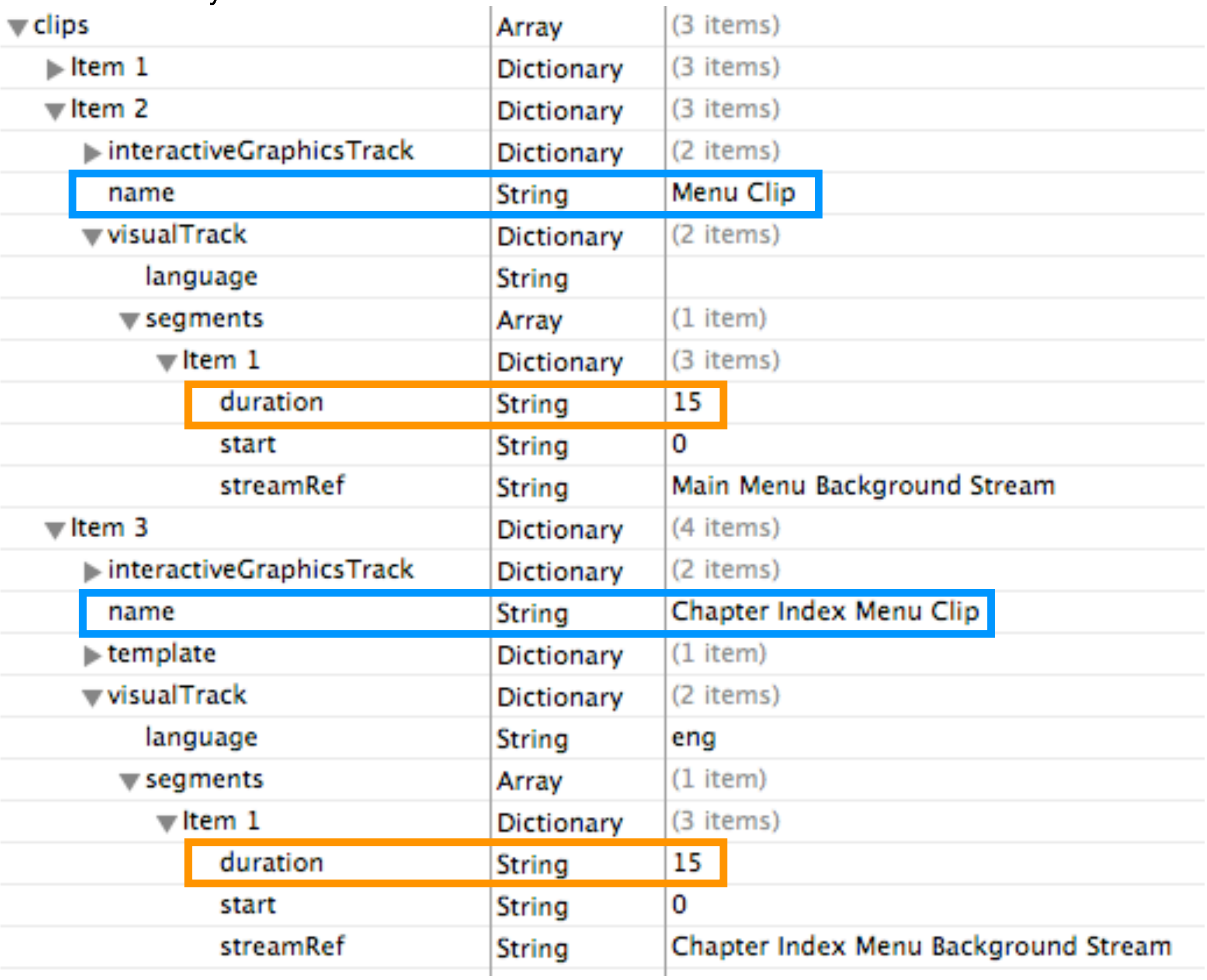


▼ assets	Array	(5 items)
▼ Item 1	Dictionary	(3 items)
name	String	Main Menu Composition
source	String	MainMenu.qtz
▼ streams	Array	(1 item)
▼ Item 1	Dictionary	(8 items)
name	String	Main Menu Background Stream
streamType	String	composition
duration	String	15
framerate	String	24
avgbitrate	String	18.0
maxbitrate	String	27.0
codec	String	H.264
passCount	Number	1
▶ Item 2	Dictionary	(3 items)
▶ Item 3	Dictionary	(3 items)
▶ Item 4	Dictionary	(2 items)
▼ Item 5	Dictionary	(3 items)
name	String	Chapter Index Menu Composition
source	String	ChapterIndexMenu.qtz
▼ streams	Array	(1 item)
▼ Item 1	Dictionary	(8 items)
name	String	Chapter Index Menu Background Stream
streamType	String	composition
duration	String	15
framerate	String	24
avgbitrate	String	18.0
maxbitrate	String	27.0
codec	String	H.264
passCount	Number	1

Under assets, find the asset whose name is “Main Menu Composition” or the one that refers to MainMenu.qtz (its source value is “MainMenu.qtz”). Under that asset, there should be a streams array. The first item is the composition stream (streamType = “composition”). There should be a duration item, with a String value. Change the duration to match your background movie’s duration. If you want to use units of seconds, just type that in, including the decimal point, if fractional seconds are needed. You can also use 27MHz units (at 24fps, each frame is 1125000, so multiply the number of frames by 1125000).

Apply the same change to the chapter index menu asset’s composition stream. To find the chapter index menu asset, look for the asset whose name is “Chapter Index Menu Composition”. Set the composition stream’s duration to the same value as above, the duration of your background movie.

Now you'll need to update the clip duration for each menu background. The clip duration must be equal to or less than the stream (movie file) duration. You can use the same value as you set for the streams above.



▼ clips	Array	(3 items)
▶ Item 1	Dictionary	(3 items)
▼ Item 2	Dictionary	(3 items)
▶ interactiveGraphicsTrack	Dictionary	(2 items)
name	String	Menu Clip
▼ visualTrack	Dictionary	(2 items)
language	String	
▼ segments	Array	(1 item)
▼ Item 1	Dictionary	(3 items)
duration	String	15
start	String	0
streamRef	String	Main Menu Background Stream
▼ Item 3	Dictionary	(4 items)
▶ interactiveGraphicsTrack	Dictionary	(2 items)
name	String	Chapter Index Menu Clip
▶ template	Dictionary	(1 item)
▼ visualTrack	Dictionary	(2 items)
language	String	eng
▼ segments	Array	(1 item)
▼ Item 1	Dictionary	(3 items)
duration	String	15
start	String	0
streamRef	String	Chapter Index Menu Background Stream

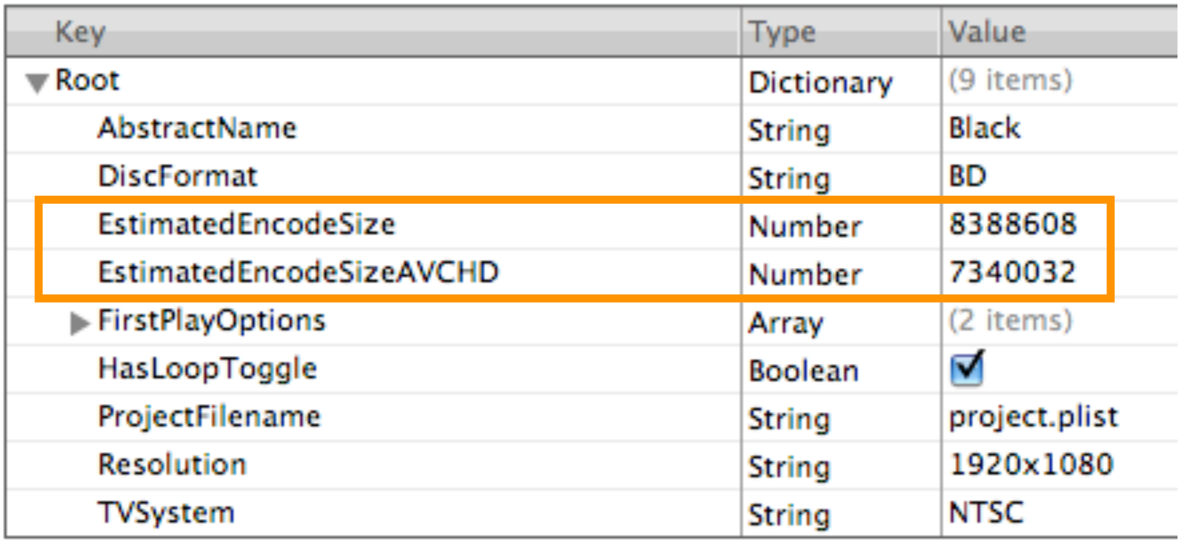
Navigate to the "clips" section and find the clip whose name is "Menu Clip". Under that item, look under "visualTrack", and then "segments" under that. There should be one item under segments with a duration, start, and streamRef value. Change the duration to match the duration of your movie (the same value used above).

Apply the same change to the chapter index menu clip. Under "clips", find the clip whose name is "Chapter Index Menu Clip". Under visualTrack/segments, the first item's duration needs to be set to the background movie duration.

Save the project.plist file (Command-S or File->Save) and you are ready to use the template! If you performed the "Copy existing template" procedure (above) properly, your template should appear in the Share UI or Compressor Job Action tab as one of the available templates.

Adjusting Estimated Size

Each template contains an estimated size value that allows Compressor or the Share feature to estimate how much space will be left on the disc for the main video and audio files. These values are located in the TemplateProperties.plist file in the template bundle alongside the project.plist file. Use the Property List Editor or your favorite text editor to edit TemplateProperties.plist in your template bundle. Here is an example of its contents:



Key	Type	Value
▼ Root	Dictionary	(9 items)
AbstractName	String	Black
DiscFormat	String	BD
EstimatedEncodeSize	Number	8388608
EstimatedEncodeSizeAVCHD	Number	7340032
▶ FirstPlayOptions	Array	(2 items)
HasLoopToggle	Boolean	☑
ProjectFilename	String	project.plist
Resolution	String	1920x1080
TVSystem	String	NTSC

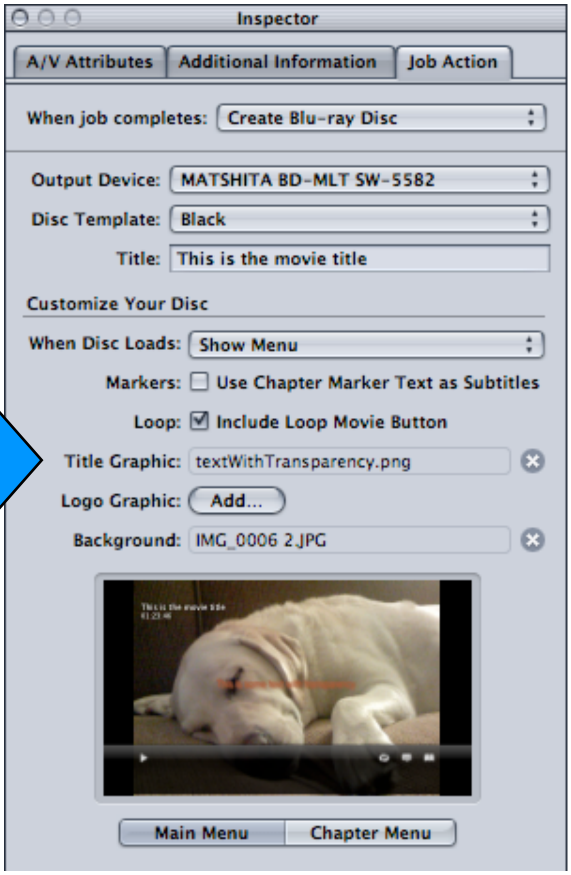
The two values, **EstimatedEncodeSize** and **EstimatedEncodeSizeAVCHD**, represent the number of bytes that the template will require on the disc for Blu-ray and AVCHD, respectively. If your background movie is longer or shorter in duration than the one provided, you will need to adjust these values. To obtain the values for your template, you will have to burn a disc or create an image file twice: once using the provided template, and a second time using your template. The difference in disc size is the number you will add to or subtract from the provided values.

Note that adjusting these values is only important if your background movie is significantly longer or shorter than the provided background movie.

For example, if you burn a Blu-ray disc or image with the provided template and it uses 245,000,000 bytes, and you burn a disc with the same source assets but with your template and it uses 350,000,000 bytes, then you need to take the difference (105,000,000 bytes) and add it to the EstimatedEncodeSize for BD. You will need to do the same for EstimatedEncodeSizeAVCHD if you want accurate information for AVCHD. Note that you will have to burn 4 discs, 1 per template and format combination, in order to get the most accurate results.

Using Transparency in Graphic Images

When specifying background, title, and logo graphics for the menu in the Compressor Job Action tab, keep in mind that if your graphic images contain transparency (alpha), they will be blended with the background. This works with the title and logo graphic images (they appear on top of the background).



When job completes: Create Blu-ray Disc

Output Device: MATSHITA BD-MLT SW-5582

Disc Template: Black

Title: This is the movie title

When Disc Loads: Show Menu

Markers: □ Use Chapter Marker Text as Subtitles

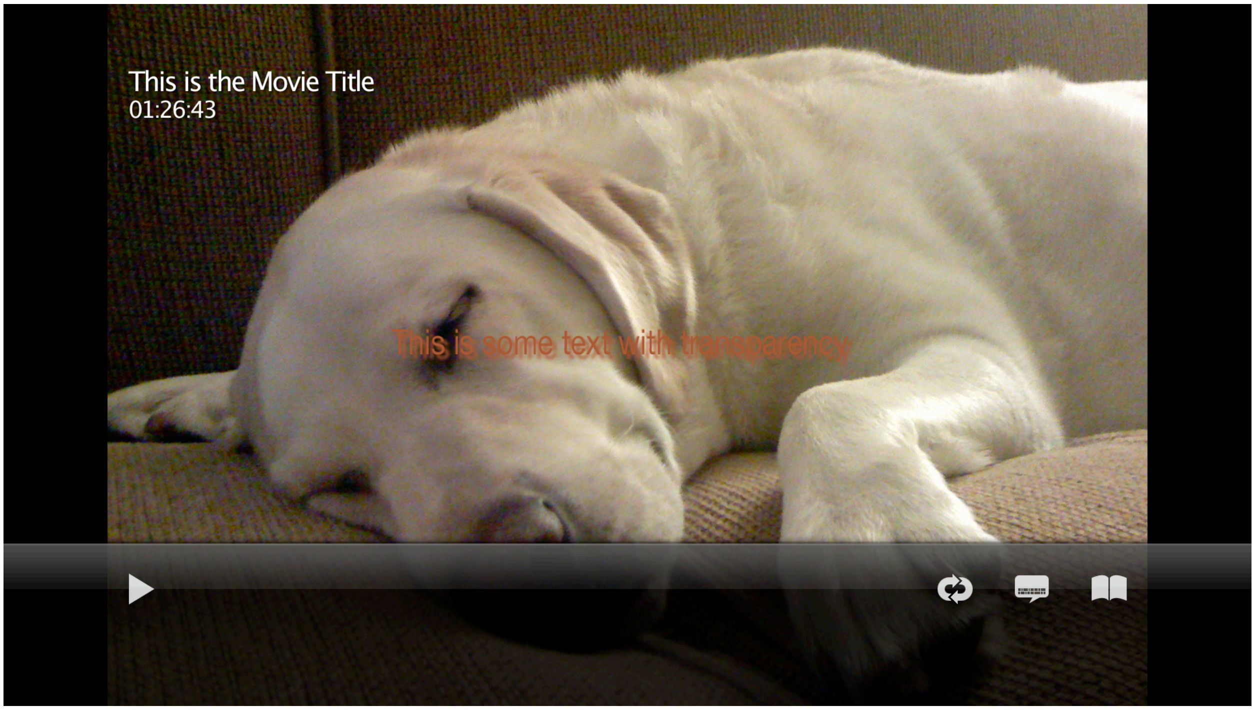
Loop: ☑ Include Loop Movie Button

Title Graphic: textWithTransparency.png

Logo Graphic: Add...

Background: IMG_0006 2.JPG

For example, if you wanted to create a drop shadow effect, modify your title graphic file to include the drop shadow effect, then use the resulting image as the “Title Graphic”. Notice how it blends with the background:



Change Button Appearance

The templates provided with Final Cut Studio have button graphics that should work well with many graphic styles, therefore we encourage you to use them as-is. However, if you really must change the way the buttons look, this section describes that procedure.

For this customization, you'll need to have Xcode (Developer) Tools installed on your computer. If you do not have Xcode Tools installed, you can install it from the Mac OS X Install Disc that came with your computer.

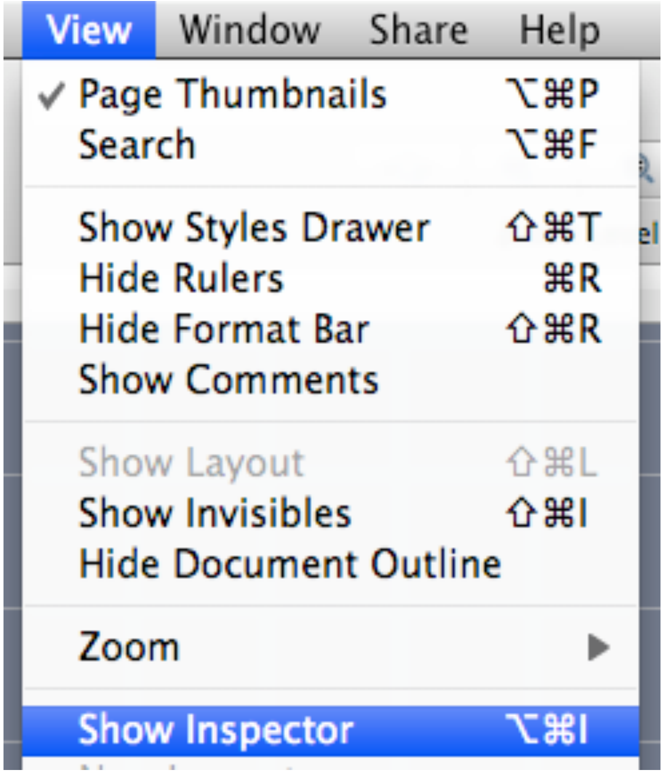
Button appearance (the image representing the button) is controlled by the Quartz Composer file for a particular menu. Let's say you want to change the appearance of the Play button on the main menu. That means you need to open up MainMenu.qtz in the Quartz Composer application.

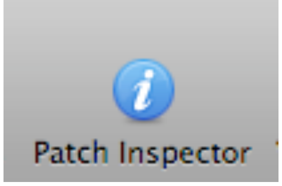
First, make sure you copy an existing template, as described above.

Navigate to the Contents/Resources folder inside your new template bundle. If you can't see inside the bundle, Control-Click the disc template bundle and select "Show Package Contents".

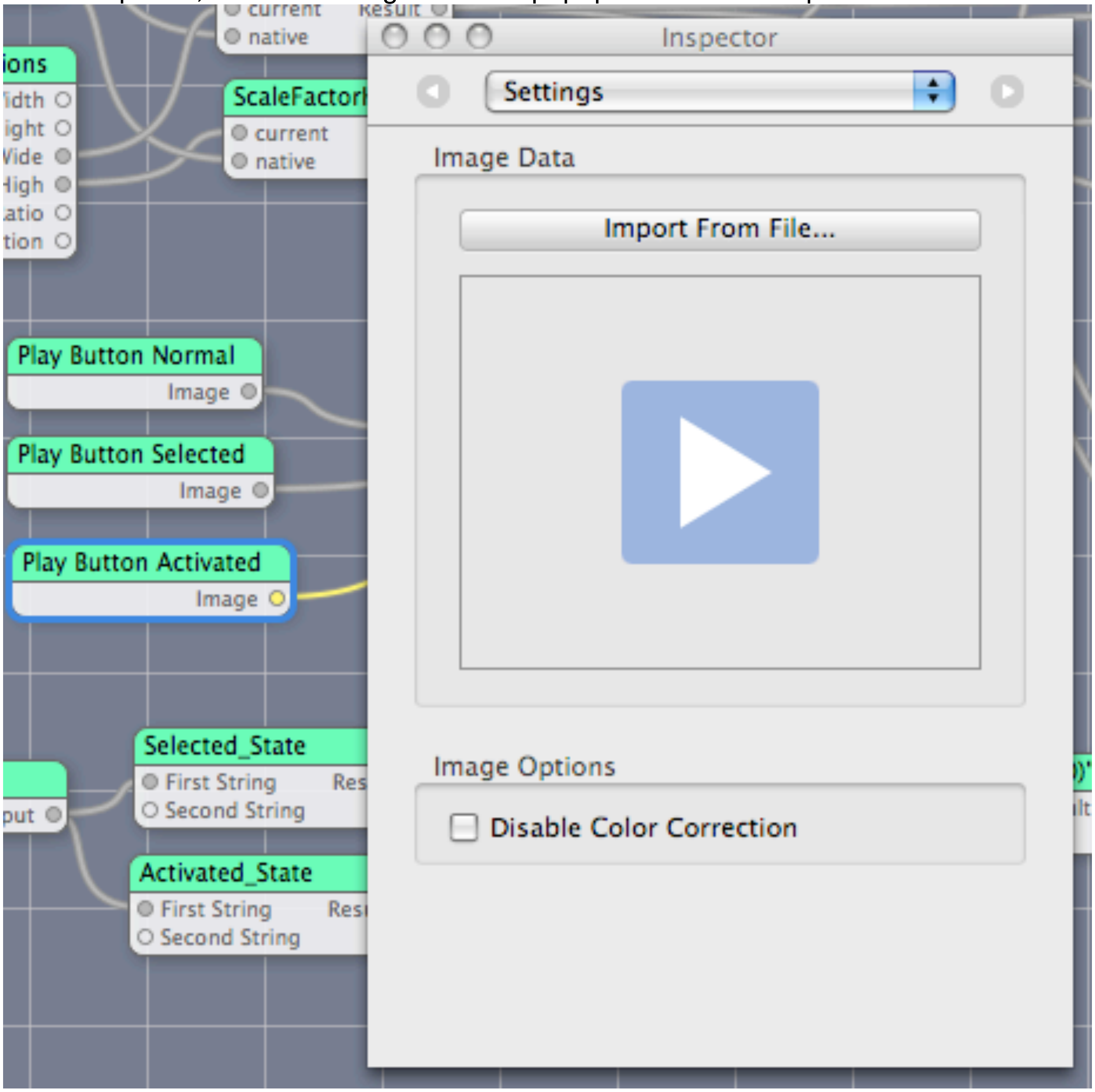
Once you're in the Resources folder, double-click MainMenu.qtz. This should launch Quartz Composer. If it does not, you can launch Quartz Composer using Finder; its location is /Developer/Applications/Quartz Composer.app.

There are two main windows in Quartz Composer for any particular composition file (MainMenu.qtz is a composition) — the Editor and the Viewer. The Viewer window shows the composition being rendered in real-time, while the Editor window allows you to edit the composition. When you make changes in the Editor window, they are immediately reflected in the Viewer window (as long as the viewer is running, not stopped). Because these templates are 1920x1080, they are natively rendered at that resolution. Any other size may not look correct, so for best results, you'll need a display with at least that resolution, if not two displays, one for viewer and the other for editor. The Editor window shows all the patches that make up the composition. A patch is a rectangle with a title and some inputs and/or outputs. The button images are each stored in "Image Importer" patches. In the templates, the patches have been renamed so that you can find them more easily. Each button has three states: Normal, Selected, and Activated. The Play button patches are named "Play Button Normal", "Play Button Selected" and "Play Button Activated". If you are looking in the editor, you can find them near the left side of the editor window. You may have to scroll the view to the left to find them. Click on the patch corresponding to the button state you want to change: Normal, Selected, or Activated. Note that "Selected" is the state when the button is "highlighted", or moused-over on a computer-based player. "Activated" is the state that appears when the user clicks or "activates" that button using the "Enter" button on the remote. The "Activated" state is usually only briefly visible during disc playback. Once you've clicked on an image importer patch of your choice, bring up the inspector panel. If it is not visible, click the "Patch Inspector" button in the Editor window toolbar, choose "Show Inspector" from the View menu, or press Control-Command-I.





In the Inspector, choose Settings from the popup button at the top.

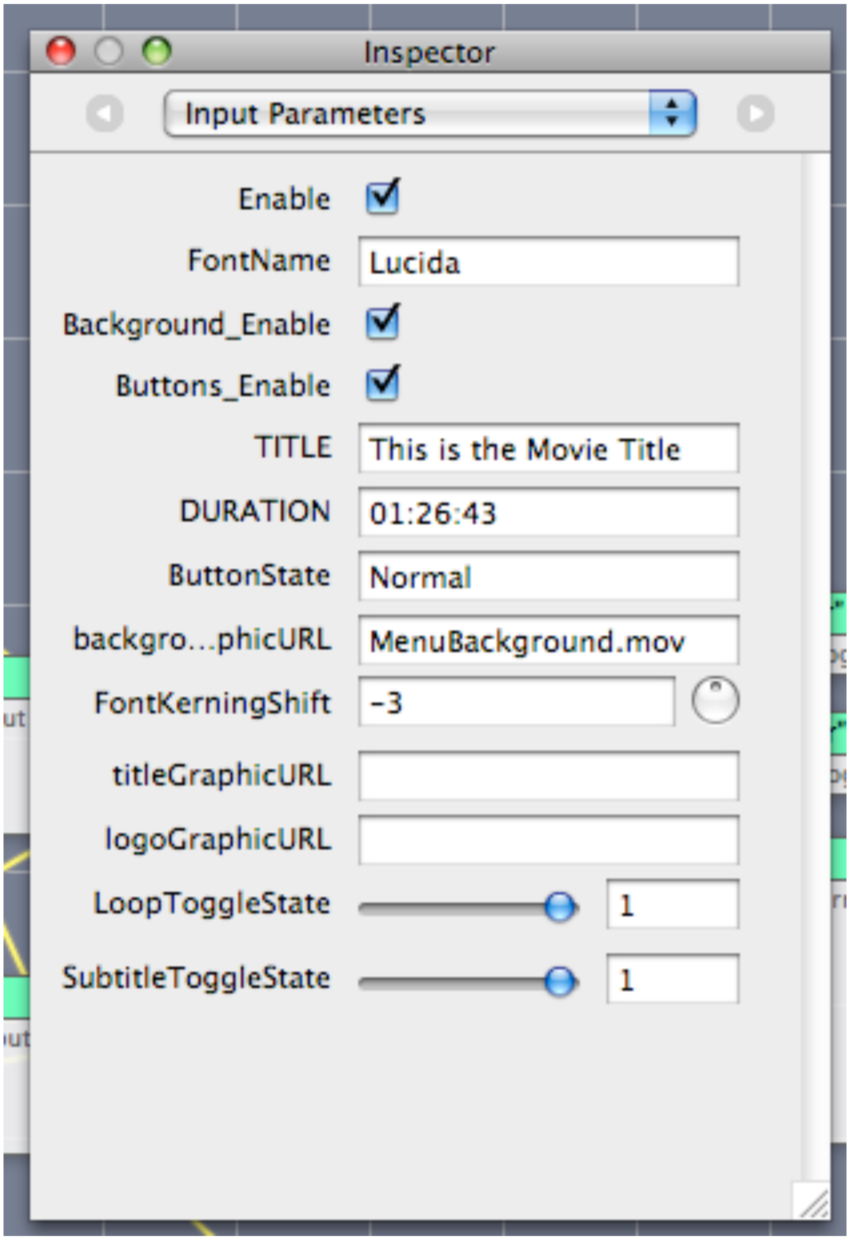


☐ Disable Color Correction

To replace the image, click the “Import From File...” button, or drag an image file from a Finder window into the image well (the button image itself).

You can see the results in the Viewer window immediately for the Normal state image, but if you want to view the Selected or Activated states, you’ll have to change one of the composition input values. If you are satisfied, just save the composition file (File->Save, or Command-S), otherwise, click the background in the Editor window (the gray grid).

In the Inspector, choose “Input Parameters” from the popup button at the top.



Enable	☑
FontName	Lucida
Background_Enable	☑
Buttons_Enable	☑
TITLE	This is the Movie Title
DURATION	01:26:43
ButtonState	Normal
backgro...phicURL	MenuBackground.mov
FontKerningShift	-3
titleGraphicURL	
logoGraphicURL	
LoopToggleState	1
SubtitleToggleState	1

Change the ButtonState value to “Selected” or “Activated”, depending on the state you want to preview. Once that is set, the viewer will show all buttons in that state. Change it back to “Normal” when you are done. Save the composition (File->Save or Command-S) and you are done!

You may also want to customize the preview thumbnail image that appears in the Share UI or Compressor Job Action template selection control. Simply replace the preview.tif file in the Contents/Resources folder of the template bundle with your own file, but make sure it is named “preview.tif” and has the same pixel dimensions (149x84).

Other button images are stored in the following image importer patches:

- Loop Toggle Off Normal
- Loop Toggle Off Selected
- Loop Toggle Off Activated
- Loop Toggle On Normal
- Loop Toggle On Selected
- Loop Toggle On Activated
- Subtitle Toggle Off Normal
- Subtitle Toggle Off Selected
- Subtitle Toggle Off Activated
- Chapter Button Normal
- Chapter Button Selected
- Chapter Button Activated

The same technique can be used for the chapter index menu. For that one, just edit ChapterIndexMenu.qtz. The button images for the chapter index menu are in the following image importer patches:

- ChapterPlayButtonNormal
- ChapterPlayButtonSelected
- ChapterPlayButtonActivated
- PrevButtonNormal
- PrevButtonSelected
- PrevButtonActivated
- HomeButtonNormal
- HomeButtonSelected
- HomeButtonActivated
- NextButtonNormal
- NextButtonSelected
- NextButtonActivated

Note that the button image when viewed from the Blu-ray disc player may not look exactly the same as the image you provided. This can happen as a result of the quantization step that is needed for all Blu-ray buttons. Button images will be quantized to a 256-color (with alpha) palette, with one entry being fully transparent. You can create images that have varying levels of alpha (transparency/opacity), but those will also be quantized so you may end up with fewer levels of alpha than in the original image.

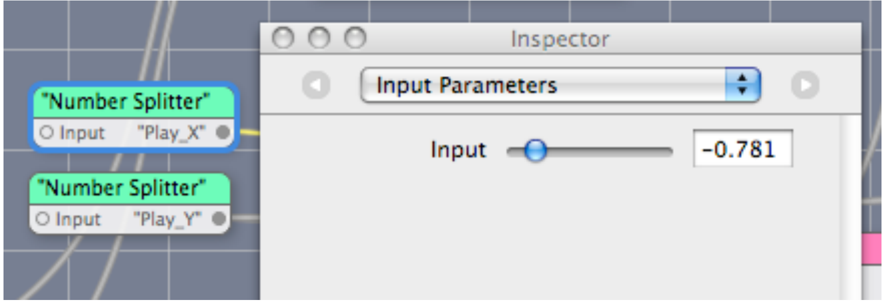
Change Button and Text Position

To change the on-screen position of a button, you'll need to edit the Quartz Composition file for that menu. Follow the instructions for copying an existing template, then navigate to the Contents/Resources folder of your new template bundle. Let's take the main menu, for example, therefore you need to double-click on MainMenu.qtz or open it from Quartz Composer.app. See the previous example if you need to find Quartz Composer.app.

Button positions are controlled by inputs of type "Number". Different buttons get their position from different patches, and some buttons get their horizontal and/or vertical positions all from the same patch.

For the Play button on the main menu, its horizontal position is controlled by a Number Splitter patch with an output named "Play_X".

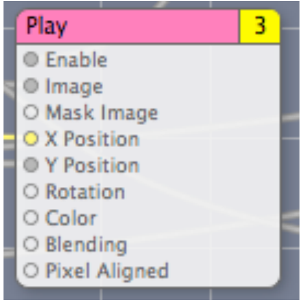
Click on that patch, bring up the Inspector, and select Input Parameters.



Input -0.781

You can grab the slider and move it. As you do, notice the button move in the Viewer window in real time. You can also type a number value into the box. Negative numbers are to the left of center, and positive values are to the right. Position values are in Quartz Composer coordinates. For more information on the Quartz Composer coordinate system, see Introduction to Quartz Composer User Guide at http://developer.apple.com/documentation/GraphicsImaging/Conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html

You can also change the vertical position of the Play button (the Number Splitter with the Play_Y output), but notice that when you do that, all of the buttons on the menu move up or down together. This is because all of the buttons have their Y (vertical) position connected to the Play_Y output, so that they remain vertically aligned. If you are feeling adventurous, you can disconnect the inputs for the Y position and set them independently in the inspector (Input Parameters) for each renderer patch. Renderer patches have pink title bars.



The renderer patch names are named after the buttons: Play, Loop Toggle, Subtitle Toggle, and Chapters.

Other buttons in the page get their horizontal dimensions from various Number Splitter patches. Look for patches with output values named:

- LoopToggle_X
- SubtitleToggle_X
- Chapters_X

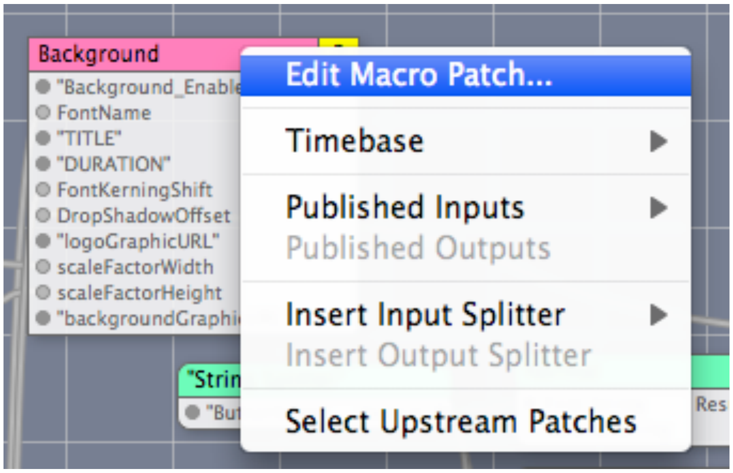
Note that these follow the naming convention described earlier: button name followed by X or Y, separated by an underscore.

Note that you need to modify the Input Parameters via the inspector, not the output parameters.

In the chapter index menu composition file (ChapterIndexMenu.qtz), you'll find the positions for those buttons as well. Look for patches named:

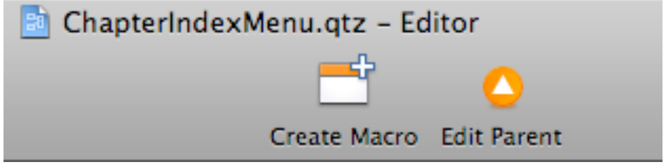
- ChapterButtonYPos
- ChapterButtonYOffset (this controls the vertical spacing between chapters)
- ChapterButtonXPos (controls horizontal position of all chapters)
- ChapterButtonXOffset (controls spacing between button and chapter name)
- Page Indicator X
- Page Indicator Y
- NavYPos (controls prev/next/home button vertical position)
- patches with outputs named Prev_X, Home_X, and Next_X

To edit the position of the text in the background you can follow the same procedure, but to find the text position in ChapterIndexMenu.qtz, you'll have to edit the background macro patch. Just find the macro patch named "Background", Control-Click and select "Edit Macro Patch..."



The title position is controlled by the "Title X" patch. Likewise, the duration vertical position is controlled by the "Duration Y" patch. Other positions are set directly on the inputs of each renderer.

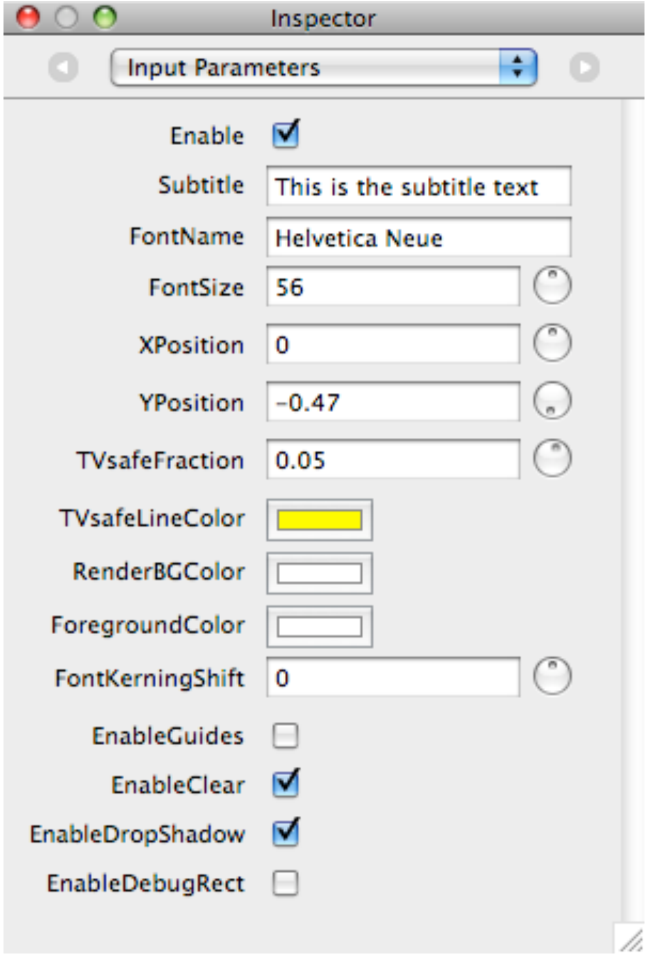
To get back to the root patch, click the Edit Parent button in the Editor toolbar:



When you're done, save the composition using File->Save or Command-S.

Change Subtitle Appearance

To change the appearance of your subtitles (the ones generated automatically based on marker names in the video), you'll need to edit the Quartz Composition file used for subtitle rendering, namely SubtitlePGStream.qtz located in your template bundle. Follow the instructions for editing Quartz Composer files under the previous section, Change Button and Text Position. The properties you can edit are available in the inspector window for the root patch. Click on the background of the Quartz Composer editor window, then click the Patch Inspector button in the toolbar.
This will display the inspector for the root patch. Select Input Parameters at the top.



Enable ☑
Subtitle: This is the subtitle text
FontName: Helvetica Neue
FontSize: 56
XPosition: 0
YPosition: -0.47
TVsafeFraction: 0.05
TVsafeLineColor: 🟨
RenderBGColor: ⬜
ForegroundColor: ⬜
FontKerningShift: 0
EnableGuides ☐
EnableClear ☑
EnableDropShadow ☑
EnableDebugRect ☐

The inputs you can change are:
- •FontName - must match a font name installed on the system, and optionally a font weight, e.g. "Helvetica Neue Bold". If no match is found, it defaults to Lucida Grande.
- •FontSize
- •XPosition - 0 means centered horizontally.
- •YPosition - 0 means centered vertically.
- •FontKerningShift - 0 means natural spacing based on the font specified above.

Other values should be left alone, otherwise you may experience undesired results.

Caveats

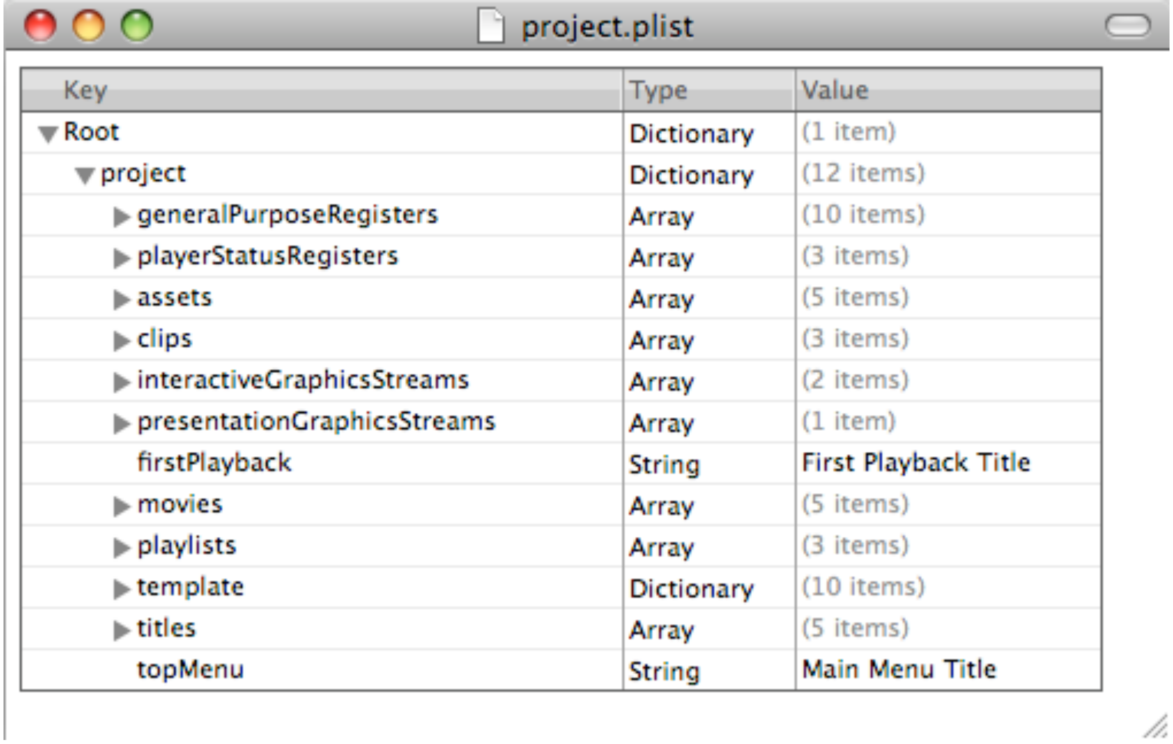
One thing to note about changing the Apple-provided menu background movie: if you replace it with a movie that is longer in duration than the stock movie, Compressor may not be able to accurately determine the remaining size on the Blu-ray disc, so you may run out of space when you try to burn. Try to use background movies that are the same duration or shorter than those in the Apple-provided templates. Or, if your main feature is not very long (less than a single Blu-ray disc side), you will probably not run into that problem.

Appendix

Project File Schema

The project.plist file represents the project, which defines the disc structure. In a template, the project.plist file does not necessarily define the complete disc structure; it also contains placeholders and other items that direct the template processor in creating the final disc structure. To edit a property list file, it is best to use the Property List Editor application, located on your computer's filesystem in /Developer/Applications/Utilities. Alternatively you can use any text editor to edit project.plist because it is in standard XML format.

Top level: project



project.plist

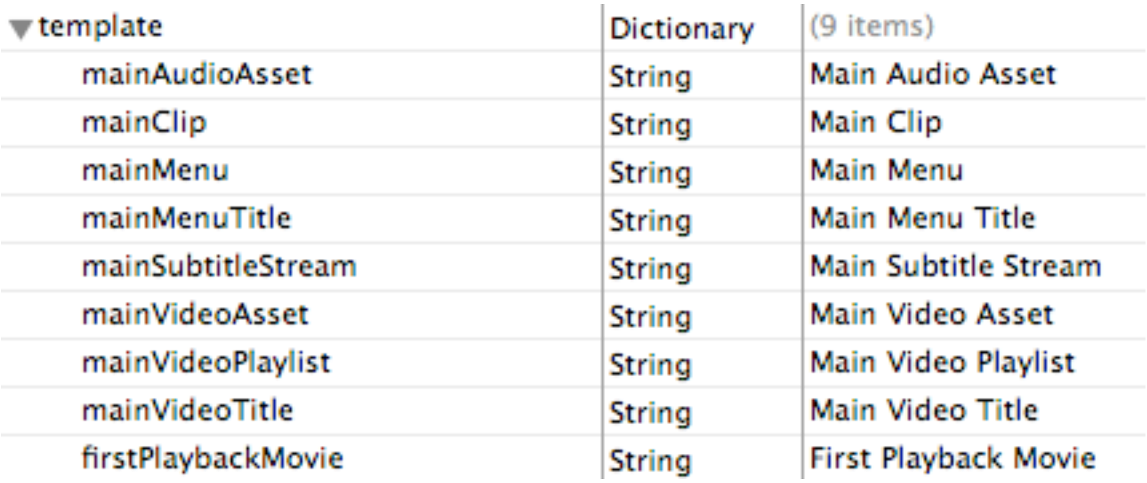
Key	Type	Value
▼ Root	Dictionary	(1 item)
▼ project	Dictionary	(12 items)
▶ generalPurposeRegisters	Array	(10 items)
▶ playerStatusRegisters	Array	(3 items)
▶ assets	Array	(5 items)
▶ clips	Array	(3 items)
▶ interactiveGraphicsStreams	Array	(2 items)
▶ presentationGraphicsStreams	Array	(1 item)
firstPlayback	String	First Playback Title
▶ movies	Array	(5 items)
▶ playlists	Array	(3 items)
▶ template	Dictionary	(10 items)
▶ titles	Array	(5 items)
topMenu	String	Main Menu Title

The "project" level contains everything in the project. This section describes parts of the project structure that may be useful for changing the look of your custom template, namely:

- template
- assets
- clips
- movies
- titles
- firstPlayback
- topMenu

template

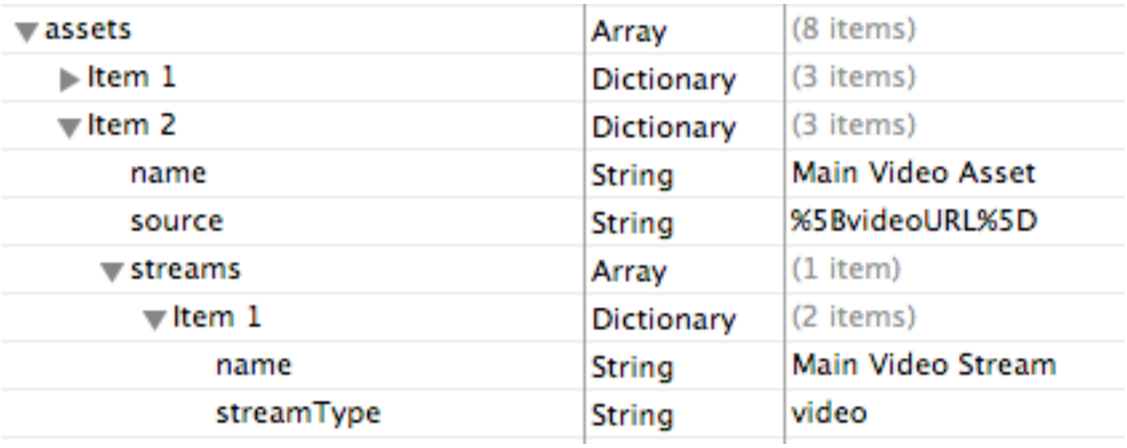
The template dictionary directly under “project” contains pointers for the template processor so it knows how to find specific objects that it cares about. Each item refers to an asset, clip, menu, title, playlist, or movie by name. The value must match an object’s name unique to its location in the project hierarchy.



▼ template	Dictionary	(9 items)
mainAudioAsset	String	Main Audio Asset
mainClip	String	Main Clip
mainMenu	String	Main Menu
mainMenuTitle	String	Main Menu Title
mainSubtitleStream	String	Main Subtitle Stream
mainVideoAsset	String	Main Video Asset
mainVideoPlaylist	String	Main Video Playlist
mainVideoTitle	String	Main Video Title
firstPlaybackMovie	String	First Playback Movie

assets

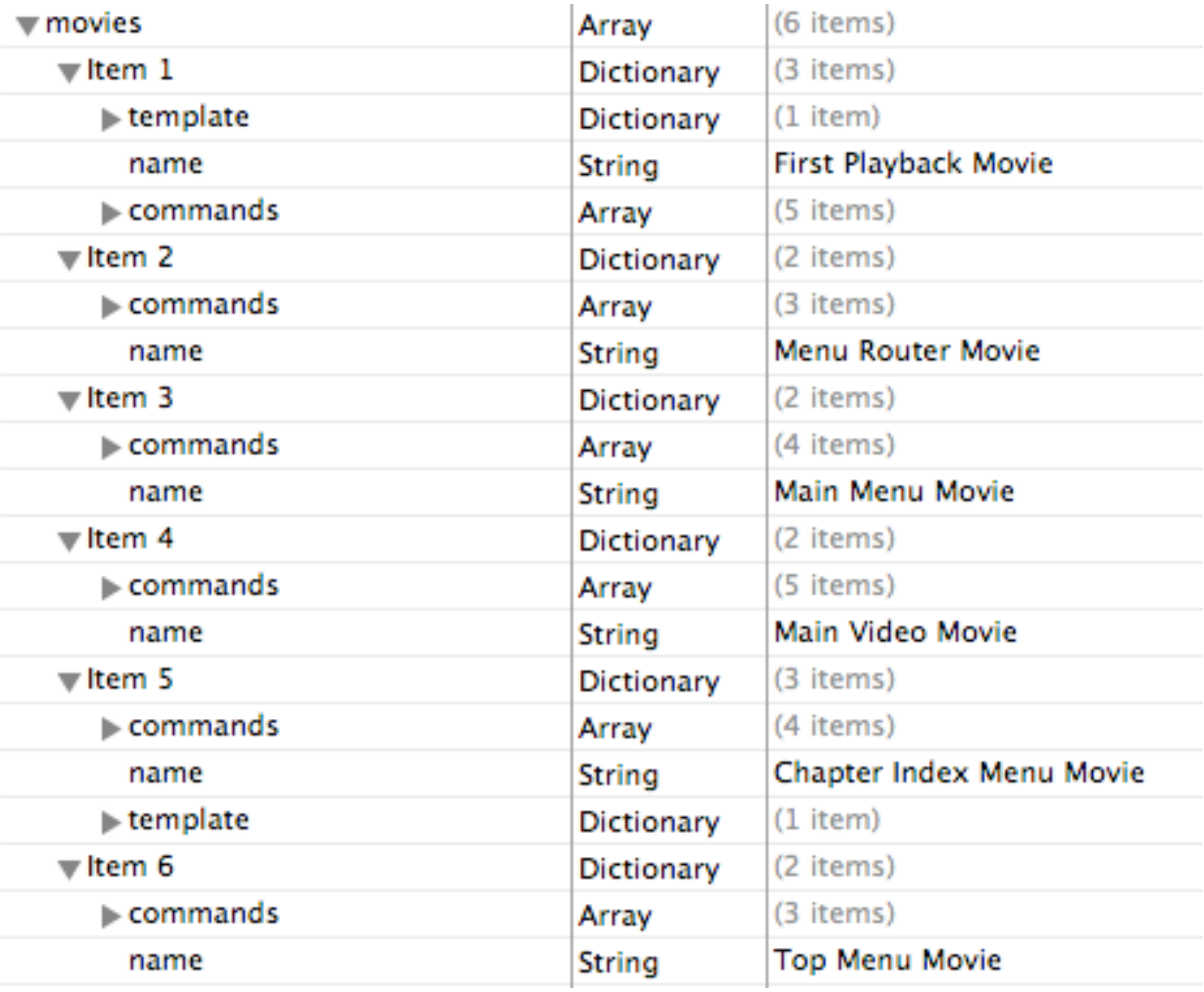
Assets are external file references. They can refer to media files such as video, audio, or graphics, subtitles, or even Quartz Compositions for rendering menus or subtitles. Each asset has a name and a source. The “source” value is a filename whose path is relative to the location of the project.plist file on disk, or an absolute URL. Some assets have a “streams” array, which lists streams from the asset that are referenced by other elements of the project.



▼ assets	Array	(8 items)
▶ Item 1	Dictionary	(3 items)
▼ Item 2	Dictionary	(3 items)
name	String	Main Video Asset
source	String	%5BvideoURL%5D
▼ streams	Array	(1 item)
▼ Item 1	Dictionary	(2 items)
name	String	Main Video Stream
streamType	String	video

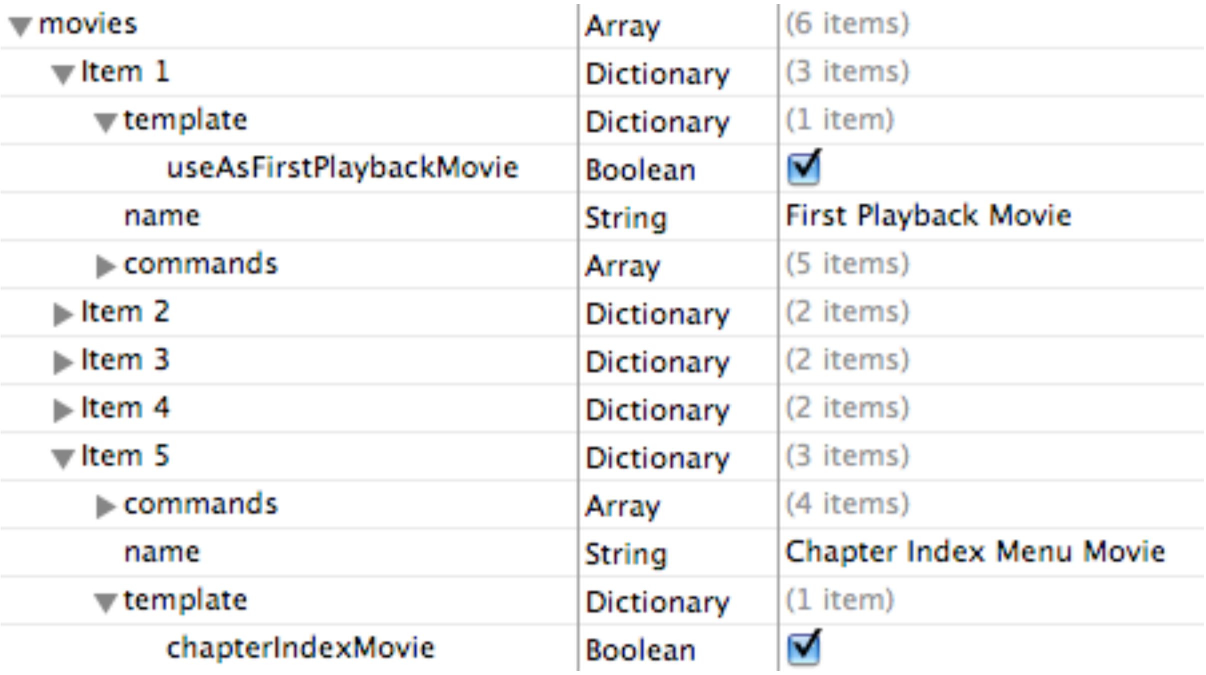
movies

Movies, as specified in the Blu-ray Disc™ format, are not really “movies” as we know them, but rather just a list of commands that are executed by the player. They are called “movies” because they contain all the commands to play back an entire (real) movie. Each movie has a name, a list of commands, and an optional “template” dictionary.



▼ movies	Array	(6 items)
▼ Item 1	Dictionary	(3 items)
▶ template	Dictionary	(1 item)
name	String	First Playback Movie
▶ commands	Array	(5 items)
▼ Item 2	Dictionary	(2 items)
▶ commands	Array	(3 items)
name	String	Menu Router Movie
▼ Item 3	Dictionary	(2 items)
▶ commands	Array	(4 items)
name	String	Main Menu Movie
▼ Item 4	Dictionary	(2 items)
▶ commands	Array	(5 items)
name	String	Main Video Movie
▼ Item 5	Dictionary	(3 items)
▶ commands	Array	(4 items)
name	String	Chapter Index Menu Movie
▶ template	Dictionary	(1 item)
▼ Item 6	Dictionary	(2 items)
▶ commands	Array	(3 items)
name	String	Top Menu Movie

When a template dictionary exists under a movie item, a boolean value called "useAsFirstPlaybackMovie" set to true indicates that the template processor will perform special processing and always sets this movie as the first to play back when the Blu-ray disc is inserted into the player. If this value is false or nonexistent, the template processor follows the firstPlayback setting in the project. The special processing is needed to set up the initial register values when the disc is inserted.

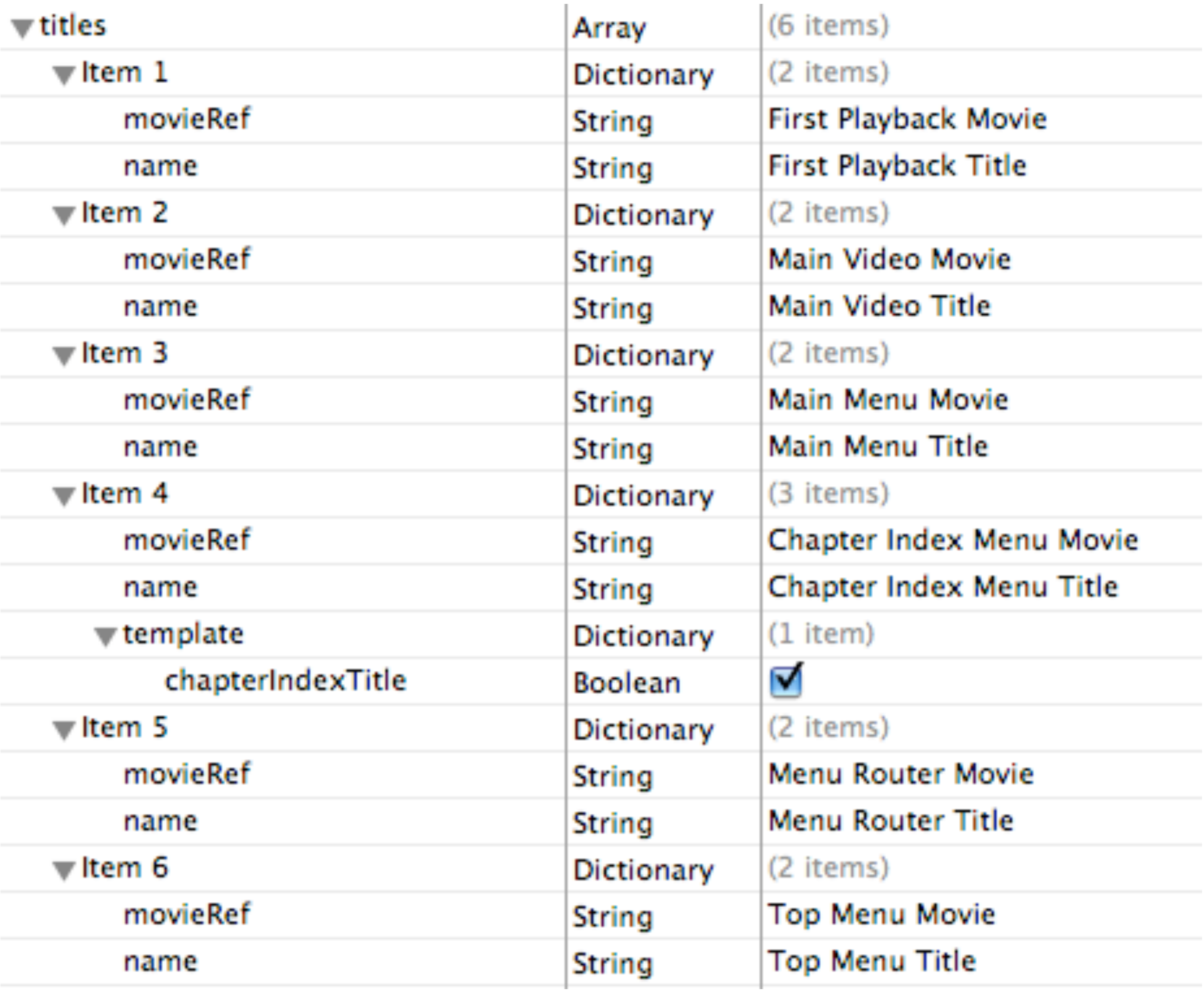


▼ movies	Array	(6 items)
▼ Item 1	Dictionary	(3 items)
▼ template	Dictionary	(1 item)
useAsFirstPlaybackMovie	Boolean	☑
name	String	First Playback Movie
▶ commands	Array	(5 items)
▶ Item 2	Dictionary	(2 items)
▶ Item 3	Dictionary	(2 items)
▶ Item 4	Dictionary	(2 items)
▼ Item 5	Dictionary	(3 items)
▶ commands	Array	(4 items)
name	String	Chapter Index Menu Movie
▼ template	Dictionary	(1 item)
chapterIndexMovie	Boolean	☑

Another template dictionary boolean value called "chapterIndexMovie" can be present, and when set to true, indicates to the template processor that this movie belongs to the chapter index menu, and will be removed if no chapter index menu is generated.

titles

A title is just a way to reference a particular movie so that it can be referenced by other elements such as commands. For example, the Jump Title command takes a title as an operand, which is the title to which to jump. First Playback and Top Menu also reference titles. A title has a name and a movieRef. The movieRef value matches the name of a movie in the project.

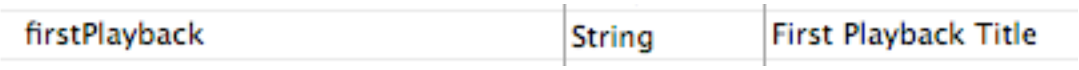


▼titles	Array	(6 items)
▼Item 1	Dictionary	(2 items)
movieRef	String	First Playback Movie
name	String	First Playback Title
▼Item 2	Dictionary	(2 items)
movieRef	String	Main Video Movie
name	String	Main Video Title
▼Item 3	Dictionary	(2 items)
movieRef	String	Main Menu Movie
name	String	Main Menu Title
▼Item 4	Dictionary	(3 items)
movieRef	String	Chapter Index Menu Movie
name	String	Chapter Index Menu Title
▼template	Dictionary	(1 item)
chapterIndexTitle	Boolean	☑
▼Item 5	Dictionary	(2 items)
movieRef	String	Menu Router Movie
name	String	Menu Router Title
▼Item 6	Dictionary	(2 items)
movieRef	String	Top Menu Movie
name	String	Top Menu Title

If a title has a template dictionary with a boolean value called “chapterIndextitle”, and that value is true, then the title is treated as part of the chapter index menu. If the chapter index menu is not generated, then the chapter index menu title is removed from the project.

firstPlayback

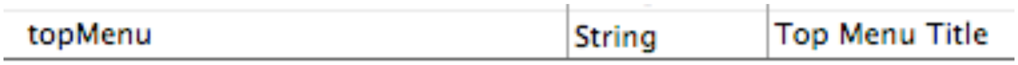
First Playback defines the title that will be the first to be played when the Blu-ray disc is inserted into the player. It refers to a title by name:



firstPlayback	String	First Playback Title

topMenu

Top Menu defines the title that will be played when the user presses the “Top Menu” button on the Blu-ray remote. It refers to a title by name:



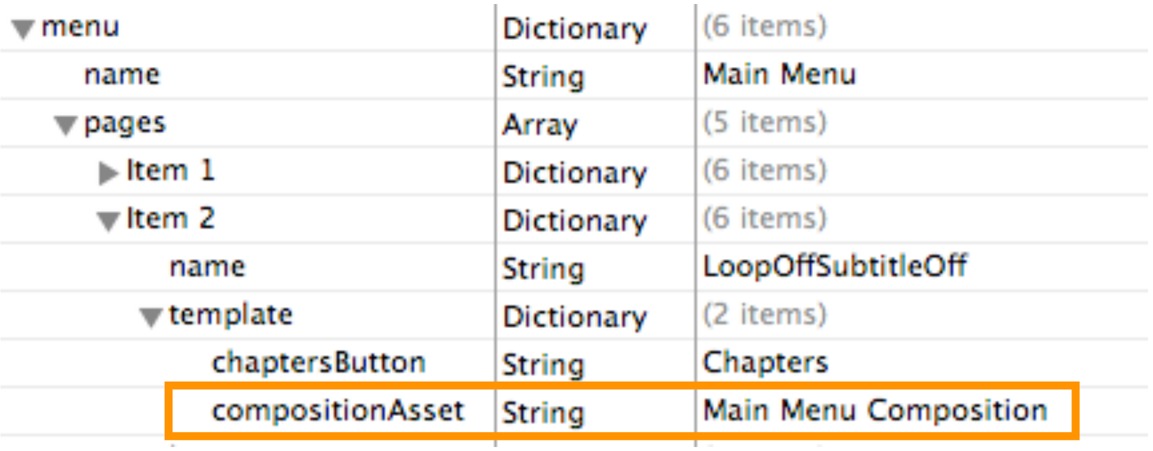
topMenu	String	Top Menu Title

Quartz Composer Files and Structure

The QC files provided in the Blu-ray templates for menu rendering are structured with two logical layers: a background layer and a button layer. The background layer contains a background image or movie, optional title image and logo image, and some text such as the title and movie duration. The button layer contains all of the buttons. The files also have "inputs" that are used for controlling text substitution such as the disc title, and controlling parameters such as which buttons are visible or where the user-provided images reside. During the disc publishing process, these compositions are rendered to create 2-D images, which are either encoded to video or used as menu buttons on the Blu-ray disc. The two composition files used during the disc publishing process generate the visual elements of menus (main menu and chapter index menu).

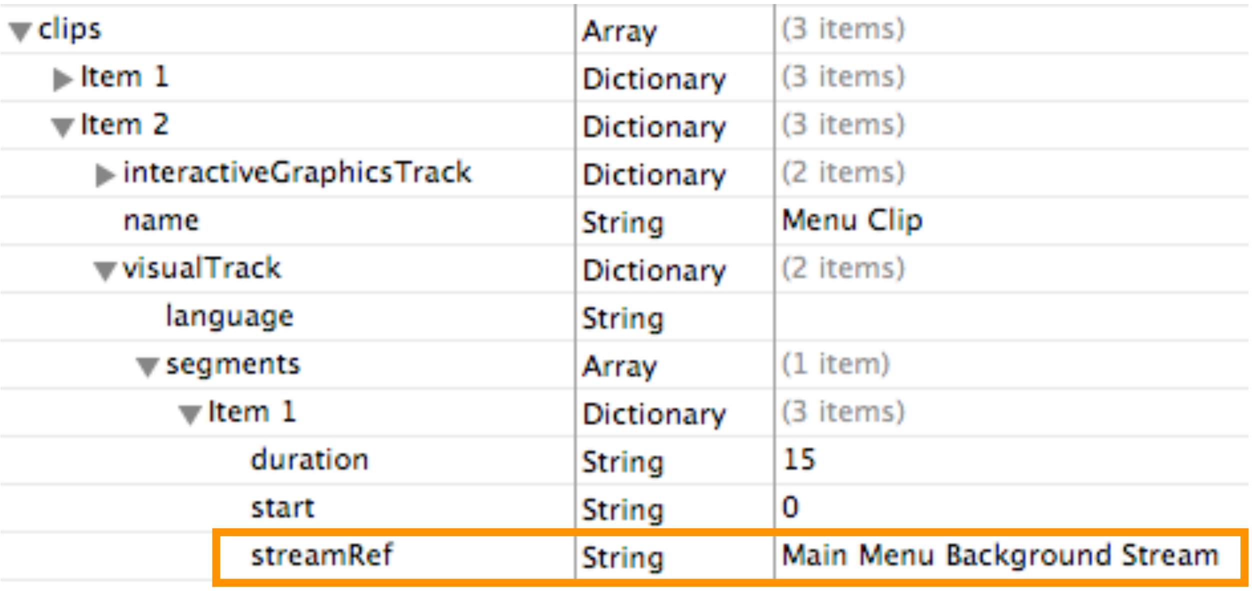
Referencing Quartz Composition (.qtz) Files

As mentioned earlier, the project.plist file contains references to the composition files (MainMenu.qtz and ChapterIndexMenu.qtz). For more information on the project.plist format, see the Appendix. Certain assets in the project.plist file refer to compositions. The "source" for those assets are .qtz files. Menus in the project contain pages that refer to composition assets via their "compositionAsset" item, as shown in the following portion of the project.plist file, as displayed by the Property List editor:



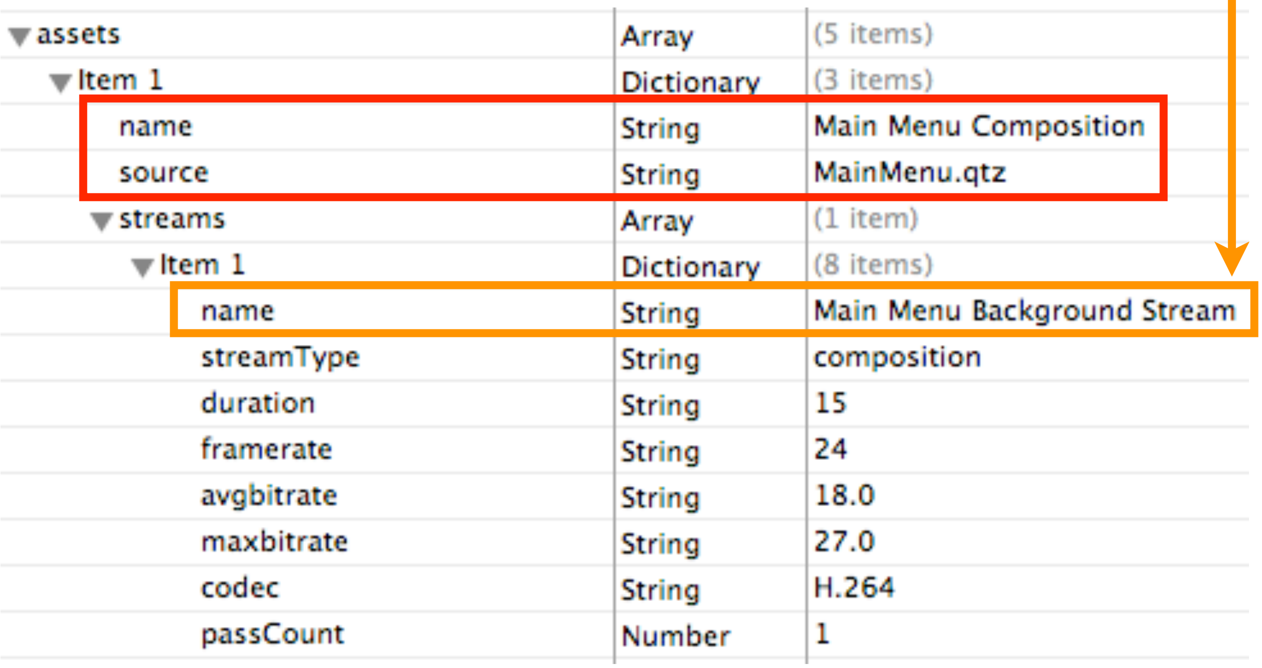
▼ menu	Dictionary	(6 items)
name	String	Main Menu
▼ pages	Array	(5 items)
▶ Item 1	Dictionary	(6 items)
▼ Item 2	Dictionary	(6 items)
name	String	LoopOffSubtitleOff
▼ template	Dictionary	(2 items)
chaptersButton	String	Chapters
compositionAsset	String	Main Menu Composition

A menu background in the Blu-ray Disc™ format is just another “clip”. In project.plist, a clip’s visual track can contain a segment whose streamRef points to a stream that is a composition (streamType = composition).



▼ clips	Array	(3 items)
▶ Item 1	Dictionary	(3 items)
▼ Item 2	Dictionary	(3 items)
▶ interactiveGraphicsTrack	Dictionary	(2 items)
name	String	Menu Clip
▼ visualTrack	Dictionary	(2 items)
language	String	
▼ segments	Array	(1 item)
▼ Item 1	Dictionary	(3 items)
duration	String	15
start	String	0
streamRef	String	Main Menu Background Stream

The Main Menu Background Stream is a stream under the asset pointing to MainMenu.qtz:



▼ assets	Array	(5 items)
▼ Item 1	Dictionary	(3 items)
name	String	Main Menu Composition
source	String	MainMenu.qtz
▼ streams	Array	(1 item)
▼ Item 1	Dictionary	(8 items)
name	String	Main Menu Background Stream
streamType	String	composition
duration	String	15
framerate	String	24
avgbitrate	String	18.0
maxbitrate	String	27.0
codec	String	H.264
passCount	Number	1

There are additional stream parameters for a composition that define duration, frame rate, average bit rate, maximum bit rate, codec, and passCount.

Time values

> In the project.plist file, time values such as “duration” and “start” are string values that indicate spans of time or specific points in time. The disc publisher is flexible in that it interprets these values as either seconds (easy to read) or 27MHz units (harder to read but more precise). It differentiates between the two units by the value. If the value is less than 100000, it is interpreted as seconds, otherwise it is interpreted as 27MHz units. When using seconds, one can specify fractional seconds as well, for example, “10.5”. In the 27MHz case, the values must be integral (nothing to the right of the decimal point).

duration is a string (time value, see above), and it defines the total duration of the media file.

framerate is also a string. Leave this as “24” for compatibility with all Blu-ray disc players.

codec is either “MPEG-2” or “H.264”.

avgbitrate and **maxbitrate** are average and maximum bit rate values of the encoded video stream in megabits per second.

passCount is the number of encoding passes, which can be 1 or 2.

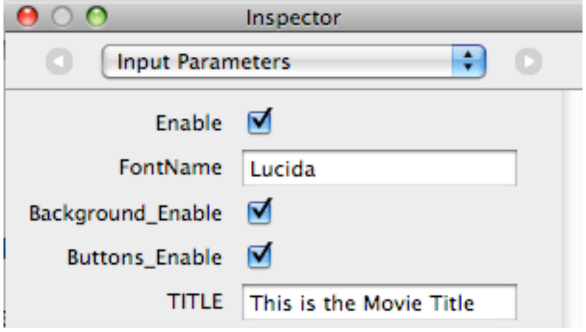
To summarize, the template processor uses compositions to create buttons as well as background (video or still) for menus.

Rendering the Menu Background

This section describes how the background is created. The disc publisher uses the “_Enable” inputs to turn layers on and off during rendering. To render the background, it enables the background layer (Background_Enable = YES) and disables the button layer (Buttons_Enable = NO).

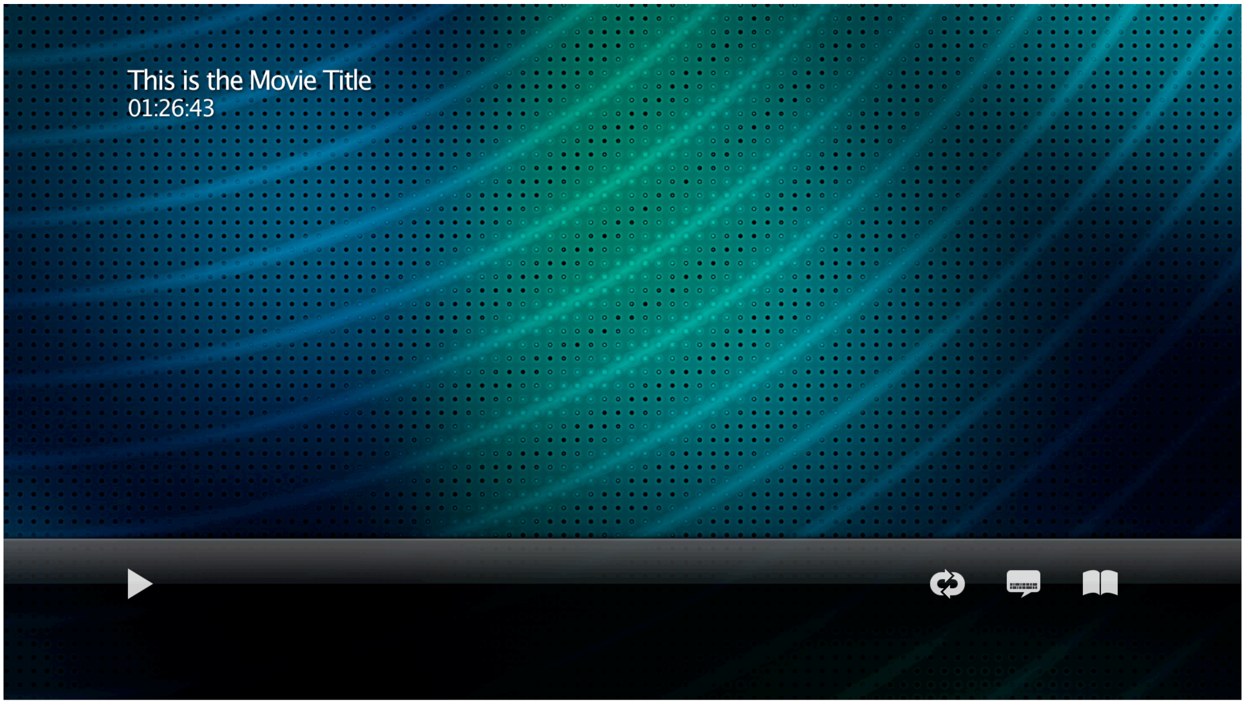
To see the “_Enable” inputs in action, open MainMenu.qtz in the Quartz Composer editor by double-clicking it in Finder. Open the Inspector window by clicking on the Patch Inspector button in the toolbar or choose Editor->Show Inspector.

Turning on both Background_Enable and Buttons_Enable in the inspector

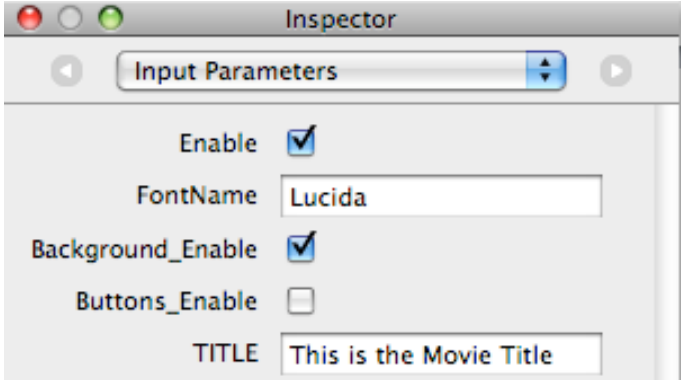


Enable ☑
FontName Lucida
Background_Enable ☑
Buttons_Enable ☑
TITLE This is the Movie Title

results in the following view of the Blue Green template:



Turning off Buttons_Enable



Enable ☑

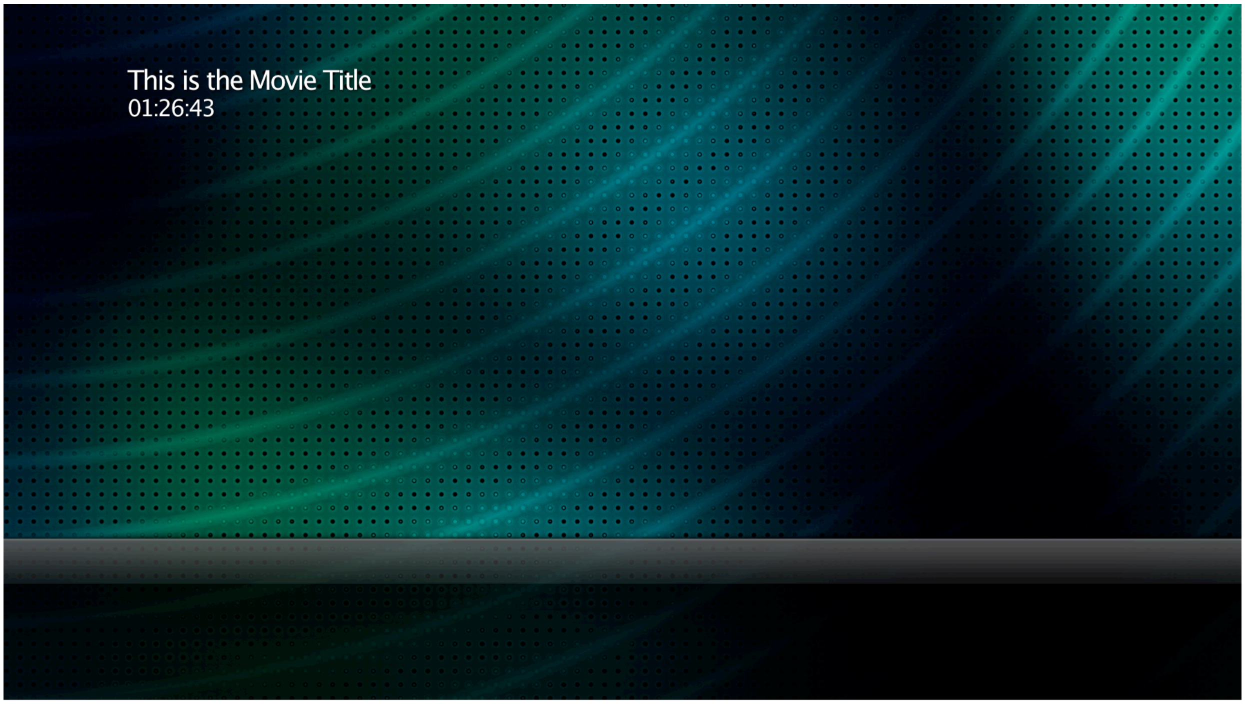
FontName Lucida

Background_Enable ☑

Buttons_Enable □

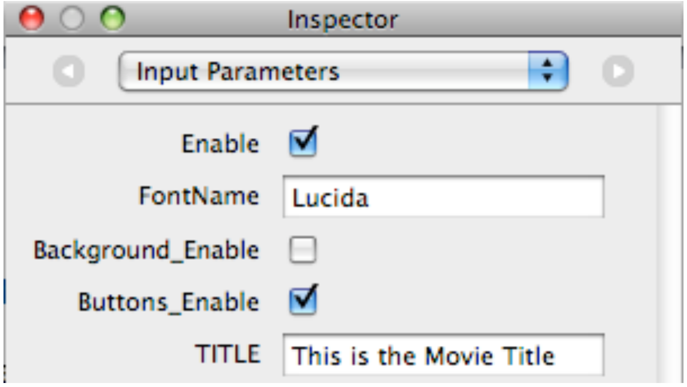
TITLE This is the Movie Title

displays the following (just the background) view of the Blue Green template:



This is the Movie Title
01:26:43

Turning off Background_Enable and turning on Buttons_Enable



Enable ☑

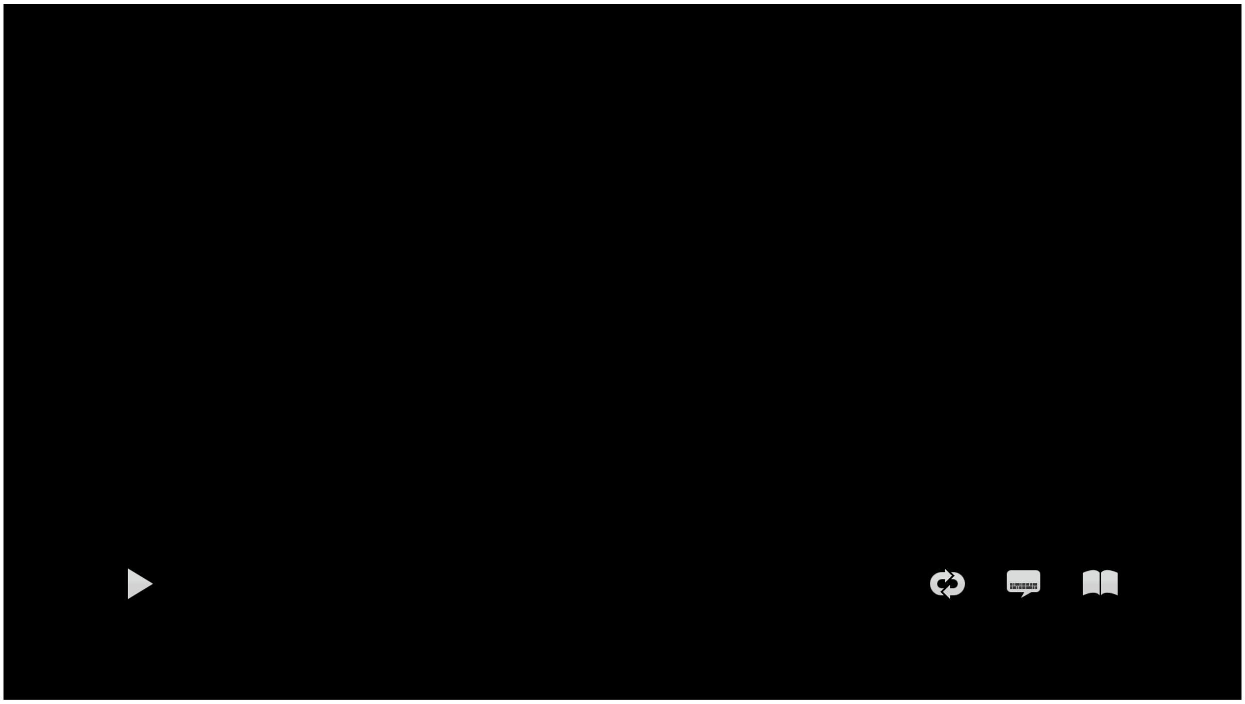
FontName Lucida

Background_Enable □

Buttons_Enable ☑

TITLE This is the Movie Title

shows just the buttons:

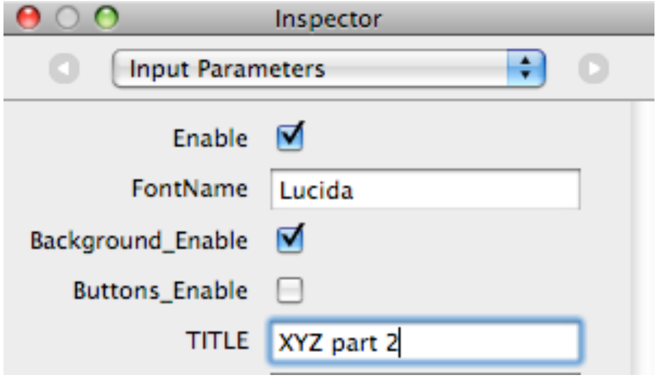


Other inputs include the URLs to the author's media files, such as logoGraphicURL, titleGraphicURL, and backgroundGraphicURL. These string values are set to the URLs provided by the author in the Share window in Final Cut Pro and Motion as well as the Compressor UI. The images are rendered into the background.

Text Substitution

Custom menus are created by rendering user-entered text into the menu background. For example, in the Compressor or Share UI, if the user assigns a title to the disc, such as “XYZ part 2”, this text will appear in the menu background. The input for the title is appropriately named, “TITLE”.

Setting TITLE to “XYZ part 2” in the inspector



Enable ☑

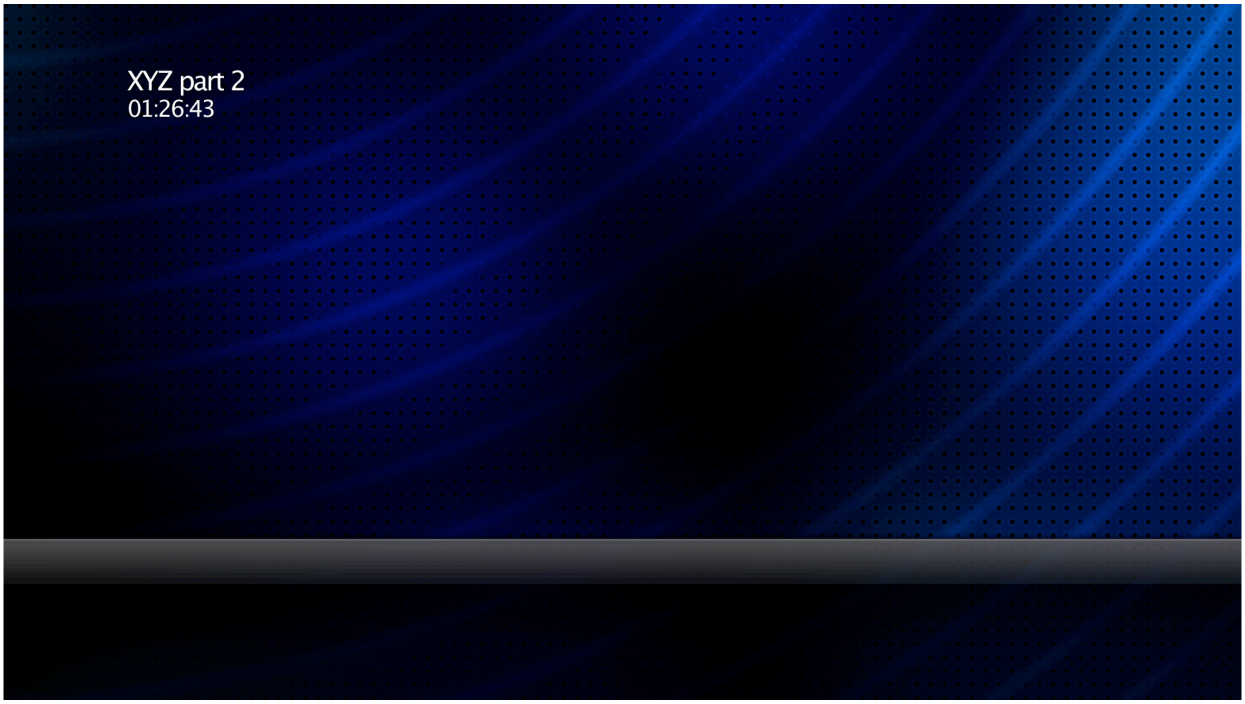
FontName Lucida

Background_Enable ☑

Buttons_Enable □

TITLE XYZ part 2

causes “XYZ part 2” to appear in the background:



Rendering Menu Buttons

Buttons in the Blu-ray Disc™ format have three states: normal, selected, and activated. For each state, a button may appear differently. In the selected state, the button appears highlighted. When the user activates the selected button by pressing the Enter key on the Blu-ray remote, the button appears in the activated state for a brief moment before the player executes the action associated with that button, such as "play the movie". A button can have a different image for each of the normal, selected, and activated states. The disc publisher renders the Quartz Composition with different input values to create these images.

Rendering button images via compositions is somewhat more involved, but similar: set the appropriate input values, render as an image, and save that image to use as a button on the disc. This is done for each button in each state on every menu page in the project.

Warning: there are some "special" buttons that are treated differently. Be careful if you decide to change the loop toggle and subtitle on/off buttons because these have additional states (on/off) that vary depending on the menu page for which they are rendered.

The project.plist file and the Quartz Composition files (MainMenu.qtz and ChapterIndexMenu.qtz) share button names. A button in the project plist file is associated with its visual counterpart in the composition by its name.

To render an image for button state, the template processor sets the appropriate input values such as Buttons_Enable to YES, Background_Enable to NO, LoopToggleState, SubtitleToggleState, ButtonState (Normal/Selected/Activated), and renders the entire frame.

To get the button state image out of the frame, its coordinate values are needed: position and dimensions. These are available as "outputs" in the composition. Position consists of X and Y, while dimensions consists of Width and Height. To get these values for a particular button, just concatenate the button name and the property with an underscore in-between. For example, if the button is named "Play" in project.plist, then the relevant outputs are:

- Play_X (X coordinate of the horizontal center of the button)
- Play_Y (Y coordinate of the vertical center of the button)
- Play_Width
- Play_Height

Note that these output values are in Quartz Composer dimensional units (not pixels).

For each button, these four values MUST be available as published outputs in the composition otherwise the button area cannot be determined.

After rendering every button state for every button in every page of a menu, the button images are quantized to 8-bit indexed color images with 8-bit alpha. In other words, the color values are analyzed and adjusted so that there are at most 256 different colors in the button images. The palette entries are 32-bits each so as to include 8 bits of alpha (transparency). The palette determined by this quantization process (for all buttons in all pages of a menu) is automatically used as the palette for every page in that menu. To be clear, even though the Blu-ray Disc™ format allows multiple palettes per menu, the quantization step generates a palette shared by all pages in the menu. There is currently no way to specify the palette for a menu page externally.

Automatic Text Substitutions

During the disc publishing process, certain values in the template (i.e. input values in the Quartz Compositions) are replaced with values provided by the user, such as “title”, or determined by the user’s media, such as “duration”. The following is a table of valid text substitution input keys and how they are determined:

QC Input	Source	Description
TITLE	Compressor or Share UI “Title” field	Disc title displayed at the top of the main menu and chapter index menu. The title is also used as the Blu-ray disc volume name.
DURATION	Main video file	Duration in HH:MM:SS format, determined by analyzing the encoded version of the user-supplied main video file. This is also displayed at the top of each menu.