



Mac OS X Server Podcast Producer Workflow Tutorial

For Version 10.6 Snow Leopard



Apple Inc.
© 2009 Apple Inc. All rights reserved.

The owner or authorized user of a valid copy of Mac OS X Server software may reproduce this publication for the purpose of learning to use such software. No part of this publication may be reproduced or transmitted for commercial purposes, such as selling copies of this publication or for providing paid-for support services.

Every effort has been made to make sure that the information in this manual is correct. Apple Inc. is not responsible for printing or clerical errors.

Apple
1 Infinite Loop
Cupertino CA 95014
408-996-1010
www.apple.com

The Apple logo is a trademark of Apple Inc., registered in the U.S. and other countries. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Apple, the Apple logo, Final Cut Studio, iCal, iChat, iPhone, Keynote, Leopard, Mac, Macintosh, Numbers, Pages, Quartz, QuickTime, Snow Leopard, Xgrid, and Xserve are trademarks of Apple Inc., registered in the U.S. and other countries. Finder is a trademark of Apple Inc.

Adobe and PostScript are trademarks of Adobe Systems Incorporated.

UNIX® is a registered trademark of The Open Group.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

019-1420/2009-11

Contents

5	Preface: About This Guide
5	What's in This Guide
6	Using Onscreen Help
7	Documentation Map
7	Viewing PDF Guides Onscreen
8	Printing PDF Guides
8	Getting Documentation Updates
9	Getting Additional Information
10	Chapter 1: Introduction
10	Before You Begin
11	Downloading the Workflow Tutorial Files
12	Additional Sources of Information
13	Chapter 2: Building a Sample Transition
13	Before You Begin
13	Summary
14	Building a Fade Transition and Making It Available to Podcast Producer
18	Adding Direction to a Transition
19	Chapter 3: Building a Bumper Title Composition
19	Before You Begin
19	Summary
20	Building a Bumper Title Composition and Making It Available to Podcast Producer
29	Chapter 4: Building an Overlay Title Composition
29	Before You Begin
29	Summary
30	Building an Overlay Title Composition and Making It Available to Podcast Producer
36	Chapter 5: Building a Dual-Source Composition
36	Before You Begin
36	Summary
37	Building a Dual-Source Composition and Making It Available to Podcast Producer

42	Chapter 6: Building a Custom Compressor Encoder
42	Before You Begin
44	Summary
44	Creating a Compressor Encoder for Use in Podcast Composer
47	Chapter 7: Creating a Transcription Workflow
47	Before You Begin
47	Summary
48	Creating and Testing the Workflow

About This Guide

Use this tutorial to learn how to expand Podcast Composer workflows and create Quartz Composer compositions and Compressor encoders and make them available to Podcast Composer.

Podcast Producer Workflow Tutorial takes you step by step through creating Quartz Composer compositions and a Compressor encoder that you can use with Podcast Composer.

This tutorial also shows you how to manually expand workflows and use the new Xgrid retry functionality by taking you step by step through creating a sample transcription service.

What's in This Guide

This guide includes the following chapters:

- Chapter 1, "Introduction," provides information you should know before going through this tutorial.
- Chapter 2, "Building a Sample Transition," describes how to create a fade transition and make it available to Podcast Composer.
- Chapter 3, "Building a Bumper Title Composition," describes how to create a bumper title composition and make it available to Podcast Composer.
- Chapter 4, "Building an Overlay Title Composition," describes how to create an overlay title composition and make it available to Podcast Composer.
- Chapter 5, "Building a Dual-Source Composition," describes how to create a dual-source overlay composition and make it available to Podcast Composer.
- Chapter 6, "Building a Custom Compressor Encoder," describes how to create a Compressor encoder and make it available to Podcast Composer.
- Chapter 7, "Creating a Transcription Workflow," describes how to expand a workflow by building a workflow that transcribes a movie and add the transcriptions as subtitles.

Note: Because Apple periodically releases new versions and updates to its software, images shown in this book may be different from what you see on your screen.

Using Onscreen Help

You can get task instructions onscreen in Help Viewer while you're managing Mac OS X Server. You can view help on a server, or on an administrator computer. (An administrator computer is a Mac OS X computer with Mac OS X Server administrator software installed on it.)

To get the most recent onscreen help for Mac OS X Server:

- Open Server Admin or Workgroup Manager and then:
 - Use the Help menu to search for a task you want to perform.
 - Choose Help > Server Admin Help or Help > Workgroup Manager Help to browse and search the help topics.

The onscreen help contains instructions taken from *Advanced Server Administration* and other administration guides.

To see the most recent server help topics:

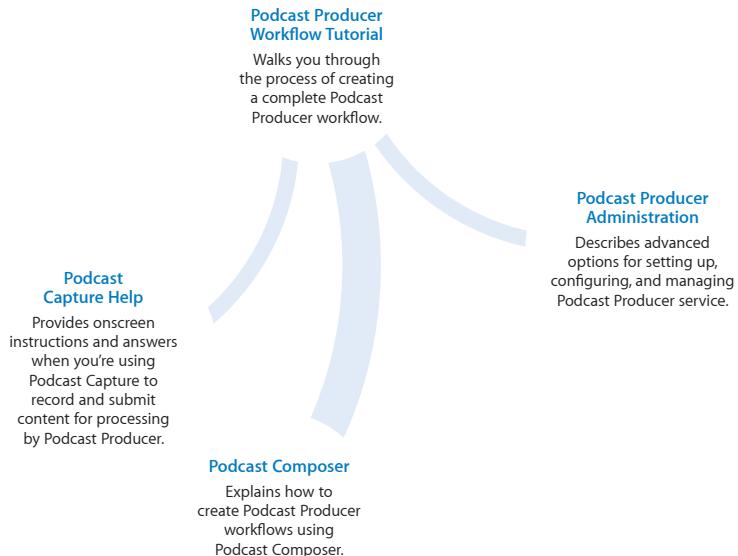
- Make sure the server or administrator computer is connected to the Internet while you're getting help.

Help Viewer automatically retrieves and caches the most recent server help topics from the Internet. When not connected to the Internet, Help Viewer displays cached help topics.

Documentation Map

Mac OS X Server has a suite of guides that cover management of individual services. Each service may depend on other services for maximum utility. The documentation map below shows some related guides that you may need in order to fully configure Podcast Producer Server to your specifications. You can get these guides in PDF format from the Mac OS X Server resources website:

www.apple.com/server/macosx/resources/



Viewing PDF Guides Onscreen

While reading the PDF version of a guide onsite:

- Show bookmarks to see the guide's outline, and click a bookmark to jump to the corresponding section.
- Search for a word or phrase to see a list of places where it appears in the document. Click a listed place to see the page where it occurs.
- Click a cross-reference to jump to the referenced section. Click a web link to visit the website in your browser.

Printing PDF Guides

If you want to print a guide, you can take these steps to save paper and ink:

- Save ink or toner by not printing the cover page.
- Save color ink on a color printer by looking in the panes of the Print dialog for an option to print in grays or black and white.
- Reduce the bulk of the printed document and save paper by printing more than one page per sheet of paper. In the Print dialog, change Scale to 115% (155% for *Getting Started*). Then choose Layout from the untitled pop-up menu. If your printer supports two-sided (duplex) printing, select one of the Two-Sided options. Otherwise, choose 2 from the Pages per Sheet pop-up menu, and optionally choose Single Hairline from the Border menu. (If you're using Mac OS X v10.4 or earlier, the Scale setting is in the Page Setup dialog and the Layout settings are in the Print dialog.)

You may want to enlarge the printed pages even if you don't print double sided, because the PDF page size is smaller than standard printer paper. In the Print dialog or Page Setup dialog, try changing Scale to 115% (155% for *Getting Started*, which has CD-size pages).

Getting Documentation Updates

Periodically, Apple posts revised help pages and new editions of guides. Some revised help pages update the latest editions of the guides.

- To view new onscreen help topics for a server application, make sure your server or administrator computer is connected to the Internet and click "Latest help topics" or "Staying current" in the main help page for the application.
- To download the latest guides in PDF format, go to the Mac OS X Server Resources website at: www.apple.com/server/resources/
- An RSS feed listing the latest updates to Mac OS X Server documentation and onscreen help is available. To view the feed, use an RSS reader application such as Safari or Mail and go to: <feed://helposx.apple.com/rss/snowleopard/serverdocupdates.xml>

Getting Additional Information

For more information, consult these resources:

- *Read Me documents*—get important updates and special information. Look for them on the server discs.
- *Mac OS X Server website* (www.apple.com/server/macosx/)—enter the gateway to extensive product and technology information.
- *Mac OS X Server Support website* (www.apple.com/support/macosxserver/)—access hundreds of articles from Apple’s support organization.
- *Apple Discussions website* (discussions.apple.com/)—share questions, knowledge, and advice with other administrators.
- *Apple Mailing Lists website* (www.lists.apple.com/)—subscribe to mailing lists so you can communicate with other administrators using email.
- *Apple Training and Certification website* (www.apple.com/training/)—hone your server administration skills with instructor-led or self-paced training, and differentiate yourself with certification.

Before you start, read this chapter to get more information about the tutorial.

This tutorial takes you step by step through creating Quartz Composer compositions and a Compressor encoder that you can use with Podcast Composer.

This tutorial also shows you how to manually expand workflows and use the new Xgrid retry functionality by taking you step by step through creating a sample transcription service.

Before You Begin

Before you start working through this tutorial, do the following:

- Set up a Podcast Producer server.

For more information about setting up Podcast Producer, see *Podcast Producer Administration*.

- Download the workflow tutorial files (see “Downloading the Workflow Tutorial Files” on page 11).
- Install Mac OS X v10.6 developer tools (see <http://developer.apple.com/mac/>).
- Install Final Cut Studio 3 if you want to create a Compressor encoder for use in Podcast Composer.

Also consider going through the Podcast Producer Workflow Tutorial for Mac OS X Server v10.5 to learn how to manually create and customize workflows.

Important: If you follow the steps in the Podcast Producer Workflow Tutorial for Mac OS X Server v10.5 to create a workflow that runs on Mac OS X Server v10.6, make sure that the workflow’s art.rb script checks for version 10.6 of Mac OS X Server.

Downloading the Workflow Tutorial Files

You can download the files used in this tutorial from <http://connect.apple.com>.

To download the tutorial files:

- 1 Go to <http://connect.apple.com>.
- 2 If you have an Apple Developer Connection (ADC) account, enter your Apple ID and password in the relevant fields and click Login.
- 3 If you don't have an ADC account, click Join Now to create an account, and then log in.
- 4 Click Downloads.
- 5 In the Downloads list on the right, click Mac OS X Server.
- 6 In the Podcast Producer Workflow Tutorial section, click Podcast Producer Workflow Tutorial 2 (Disk Image) to download the tutorial files.
- 7 Open the disk image and copy the Podcast Producer Workflow Tutorial Files folder to your Desktop.

Following is a description of the tutorial files in the Podcast Producer Workflow Tutorial Files folder:

- Titles/Bumper/

- Bumper Title Start.qtz

The bumper title composition you start with in Chapter 3, "Building a Bumper Title Composition."

- Bumper Title End.qtz

The completed bumper title composition you create in Chapter 3, "Building a Bumper Title Composition."

- Titles/Overlay/

- Overlay Title Start.qtz

The overlay title composition you start with in Chapter 4, "Building an Overlay Title Composition."

- Overlay Title End.qtz

The completed overlay title composition you create in Chapter 4, "Building an Overlay Title Composition."

- DualSources/Dual Source End.qtz

The completed dual-source composition you create in Chapter 5, "Building a Dual-Source Composition."

- Transcription Service Files/

The files you use in Chapter 7, "Creating a Transcription Workflow."

Additional Sources of Information

For information about Quartz Composer, see: http://developer.apple.com/mac/library/documentation/GraphicsImaging/Conceptual/QuartzComposer/qc_intro/qc_intro.html.

For information about the concept of bundles in Mac OS X, see: <http://developer.apple.com/mac/library/documentation/CoreFoundation/Conceptual/CFBundles/Introduction/Introduction.html>.

For information about YAML Ain't Markup Language (YAML), see: <http://www.yaml.org>.

For information about Embedded Ruby (ERB), see: <http://ruby-doc.org/stdlib/libdoc/erb/rdoc/classes/ERB.html>.

Building a Sample Transition

2

In this chapter, you use Quartz Composer to create a simple fade transition and then make it available to Podcast Composer.

Quartz Composer provides a template that allows you to automatically create a basic Fade transition. In this chapter, you use Quartz Composer's Graphic Transition template to create a basic fade transition.

Then, you make the transition available to Podcast Composer by adding the transition to a sample Podcast Composer transition bundle and storing the bundle in the relevant folder.

Note: Although this tutorial doesn't assume familiarity with Quartz Composer, it is recommended that you skim through the Quartz Composer documentation to learn how Quartz Composer works.

Before You Begin

To use Quartz Composer, install the Mac OS X v10.6 developer tools.

Summary

To create a sample transition, perform the following tasks:

- 1 Use Quartz Composer to create a basic fade transition (see "Step 1: Create the Fade Transition" on page 14).
- 2 Add the transition to a transition bundle and make it available to Podcast Composer (see "Step 2: Add the Transition to a Transition Bundle" on page 15).
- 3 Restart Podcast Composer and verify that you can access the new transition (see "Step 3: Restart Podcast Composer" on page 17).

Building a Fade Transition and Making It Available to Podcast Producer

Step 1: Create the Fade Transition

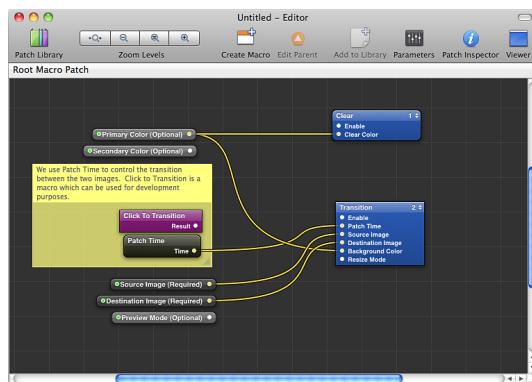
To create the Fade transition:

- 1 Launch Quartz Composer (in /Developer/Applications/).
- 2 Click Templates.

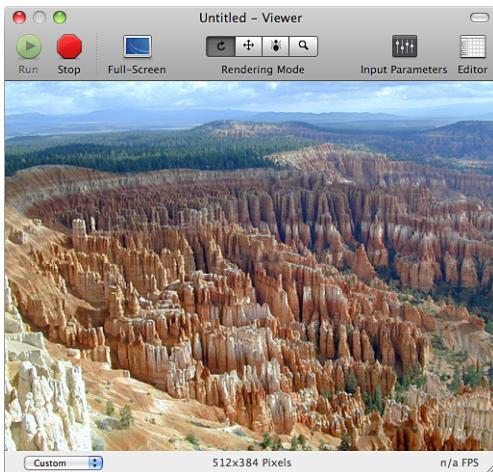


- 3 Select Graphic Transition and click Choose.

Quartz Composer opens an editor window and adds the patches and input parameters required for a basic fade transition.



- 4 To see how the fade transition works, choose Window > Untitled - Viewer.



- 5 In the Viewer window, click Stop and then Run to see a preview of the effect.
A sample source image fades into a sample destination image.
- 6 Choose File > Save As.
- 7 In the Save As field, enter Transition.qtz, choose Desktop as the destination, then click Save.
Note: To use in Podcast Composer, the name of the effect must always be Transition.qtz.
- 8 Quit Quartz Composer.

Step 2: Add the Transition to a Transition Bundle

After creating a transition using Quartz Composer, the next step is to add it to a composition bundle, which is required by Podcast Composer.

To add the Transition.qtz transition to a transition bundle:

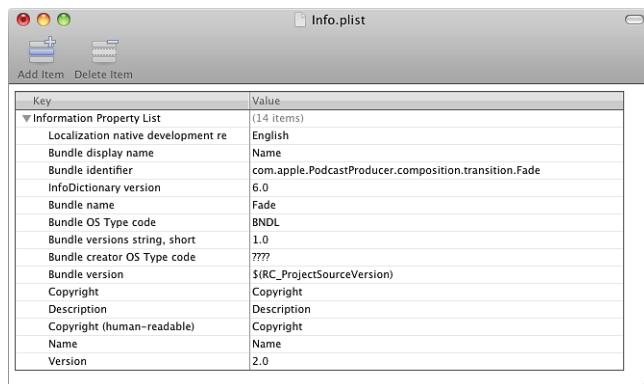
- 1 In /Applications/Server/, select Podcast Composer.
- 2 Control-click Podcast Composer and choose Show Package Contents from the pop-up menu.
- 3 Copy (Edit > Copy) Contents/Resources/Materials/Compositions/Transitions/MosaicFlipTransition.pqz.

Because the Mosaic Flip transition is similar to the Fade transition, you will use the bundle structure of the Mosaic Flip transition as a starting point instead of creating the bundle manually.

- 4 Paste MosaicFlipTransition.pqz into /Library/Application Support/Podcast Composer/Resources/Compositions/Transitions/.

This is the folder where you should store transition effects to make them available to Podcast Composer. If the folder path is not present, create it.

- 5 Rename the bundle from MosaicFlipTransition.pqz to Fade.pqz.
- 6 Modify the bundle's contents as follows:
 - a Delete the following:
 - Contents/Resources/de.lproj
 - Contents/Resources/fr.lproj
 - Contents/Resources/ja.lproj
 - Contents/Resources/Transition.qtz
 - b Move the fade transition you created earlier from the Desktop (~/Desktop/Transition.qtz) to Contents/Resources/.
 - c Open Contents/Info.plist.
 - d Change the last element in the value of the "Bundle identifier" key from MosaicFlipTransition to Fade.
 - e Change the value of the "Bundle name" key to Fade, as shown here.



- f Save and close Contents/Info.plist.
- g Open Contents/Resources/en.lproj/InfoPlist.strings.

h Change the value of the Name and Description keys as follows:

```
Name = "Fade";  
Description = "Fade transition.;"
```

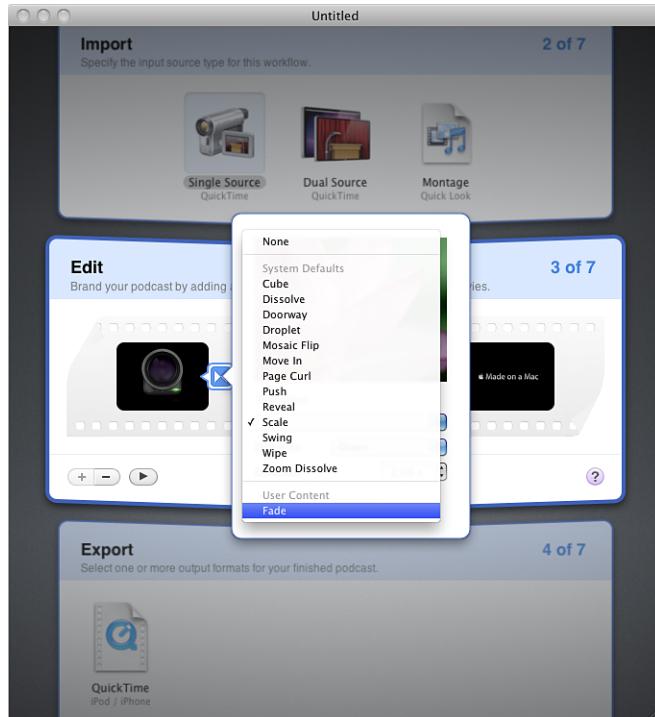


i Save and close Contents/Resources/en.lproj/InfoPlist.strings.

Step 3: Restart Podcast Composer

To verify that you can access the transition in Podcast Composer:

- 1 If opened, quit Podcast Composer.
- 2 Launch Podcast Composer.
- 3 In the Edit stage, double-click a transition icon.
- 4 Click the Transition pop-up menu and verify that you can choose Fade in the User Content section, as shown here.



Adding Direction to a Transition

Some transition compositions include a direction input port. For example, the default Cube transition composition includes a direction input port that allows you to choose one of four directions.

If you create a transition composition with a direction input port, you'll need to add an array of human readable keys to Info.plist in the transition's bundle. Each key in the array represents a direction.

For example, the Cube transition has the following array in its Info.plist file:

```
<key>Directions</key>
<array>
    <string>Top to Bottom</string>
    <string>Bottom to Top</string>
    <string>Left to Right</string>
    <string>Right to Left</string>
</array>
```

In addition, you must add the localized direction strings to the relevant InfoPlist.strings file.

For example, the Cube transition stores the German version of the direction keys in the Contents/Resources/de.lproj/InfoPlist.strings file:

```
/* Transition directions */
"Top to Bottom" = "Von oben nach unten";
"Bottom to Top" = "Von unten nach oben";
"Left to Right" = "Von links nach rechts";
"Right to Left" = "Von rechts nach links";
```

Building a Bumper Title Composition

In this chapter, you use Quartz Composer to create a bumper title composition and then make it available to Podcast Composer.

In this chapter, you use Quartz Composer to expand a sample bumper title composition and then make it available to Podcast Composer.

Before You Begin

If you haven't done so, download the sample bumper title composition as described in "Downloading the Workflow Tutorial Files" on page 11.

Summary

To create a title composition for use by Podcast Composer, perform the following tasks:

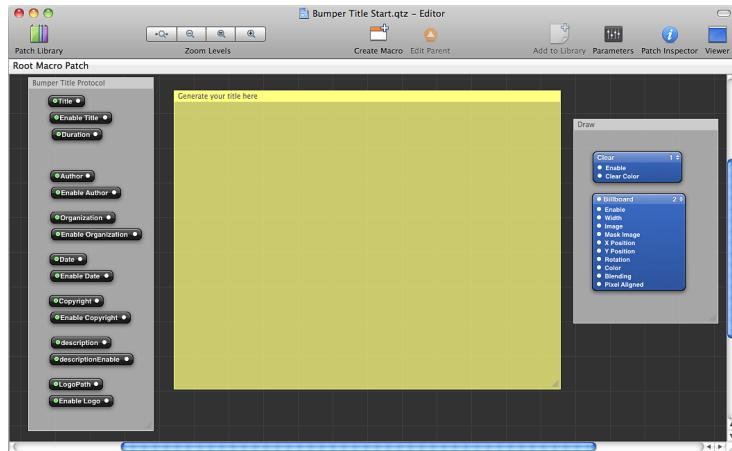
- 1 Use Quartz Composer to expand a sample bumper title composition (see "Step 1: Expand the Provided Sample Bumper Title Composition" on page 20).
- 2 Store the composition bundle in the relevant folder (see "Step 2: Add the Composition to a Composition Bundle" on page 27).
- 3 Restart Podcast Composer and verify that you can access the new bumper title composition (see "Step 3: Restart Podcast Composer" on page 28).

Building a Bumper Title Composition and Making It Available to Podcast Producer

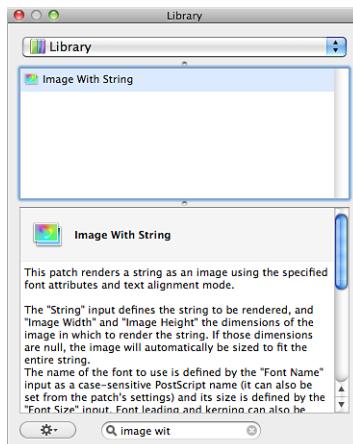
Step 1: Expand the Provided Sample Bumper Title Composition

To expand the sample bumper composition:

- 1 Open the ~/Desktop/Podcast Producer Workflow Tutorial Files/Titles/Bumper/ folder.
- 2 Double-click Bumper Title Start.qtz to open the composition in Quartz Composer.



- 3 Choose File > Save As.
- 4 Save the composition as BumperTitle.qtz on the Desktop.
- 5 Add a patch for rendering a string as an image:
 - a Open the Library window (Window > Show Patch Library).
 - b In the Library window, enter "image w" in the Search field:



The Search field allows you to quickly find the patch you're looking for.

- c Drag the Image With String patch onto the yellow note.

Note: The notes in the Editor window help you organize the elements of your composition.

- d Connect the output of the Title port (an instance of an Input Splitter patch) in the Bumper Title Protocol to the String port (an input port) of the Image With String patch.

This connection passes the title of the podcast (specified in Podcast Capture, the Podcast Capture web application, or the `podcast` command) to the Image With String patch.

For more information about the Input Splitter patch, check its description in the Library window.

- e Connect the Image port (an output port) of the Image With String patch to the Image port (an input port) of the Billboard patch.

For more information about the Billboard patch, check its description in the Library window.

- f Connect the Enable Title port (an instance of an Input Splitter patch) to the Enable port (an input port) of the Billboard patch.

- g Select the Billboard patch.

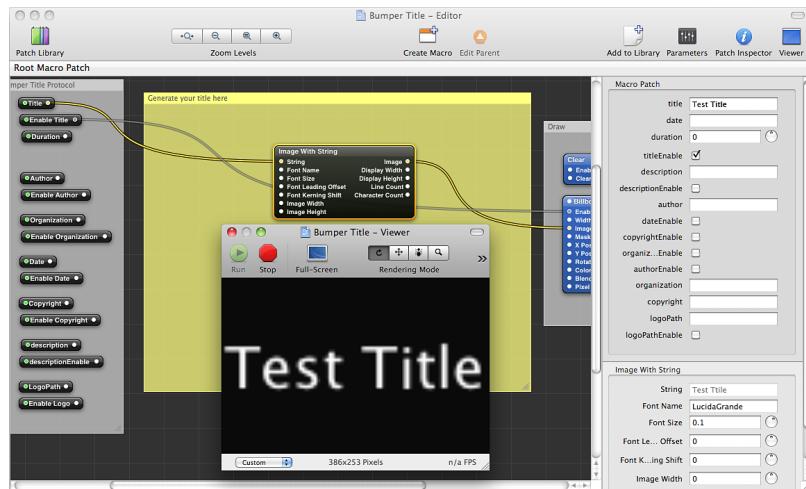
- h In the toolbar, click the Parameters button to display the Macro Patch and Billboard panes.

- i In the Macro Patch pane, enter Test Title in the title field and select titleEnable.

- j In the Billboard pane, set the Width and Y Position parameters of the Billboard patch to 1.5 and 0 respectively.

- k In the toolbar, click Viewer to open the Viewer window.

- | To verify that your connections work, in the Viewer window, click Stop, then click Run:



You should see the words Test Title rendered in the Viewer window.

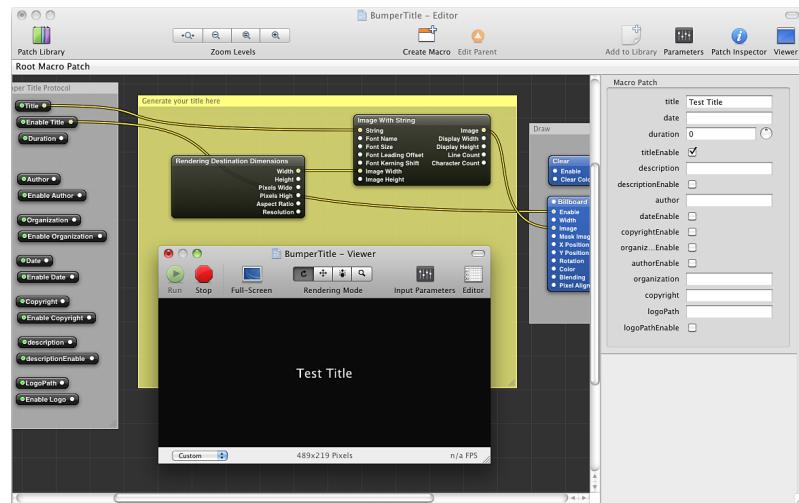
m Choose File > Save.

- 6 Add a patch that returns the dimensions of the rendering destination:
- In the Library window, enter “renderi” in the Search field.
 - Drag the Rendering Destination Dimensions patch onto the yellow note in the Editor window.
 - Connect the Width port of the Rendering Destination Dimensions patch to the Image Width port of the Image With String patch.

These connections provide the width and height of the title image that the composition renders.

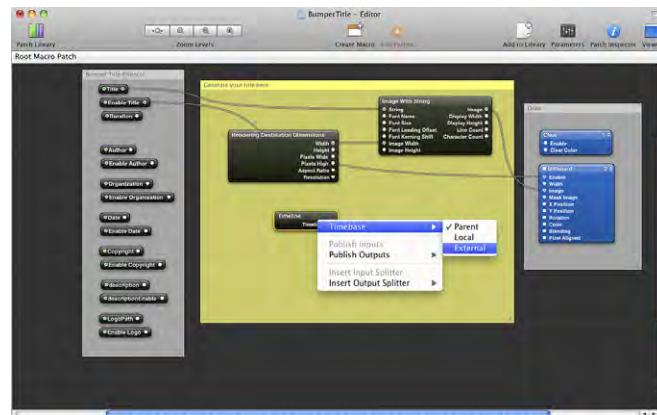
- d To verify that your connections work, in the Viewer, click Stop, then click Run.

The words Test Title should appear smaller and sharper in the Viewer window:



7 Add a Timeline patch:

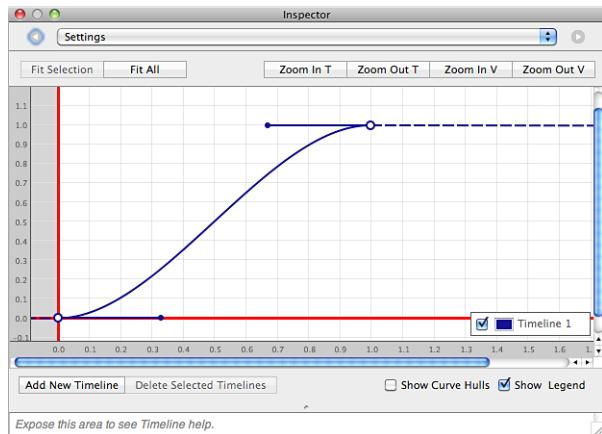
- In the Library window, enter “time” in the Search field.
- Drag the Timeline patch onto the yellow note in the Editor window.
- Control-click the Timeline patch and choose Timebase > External from the pop-up menu:



This action adds the Patch Time input port.

- With the Timeline patch selected, choose Editor > Show Inspector.

e Choose Settings from the pop-up menu.



You can also click the button at the left or right of the pop-up menu repeatedly until the Settings pane appears

f Click Add New Timeline.

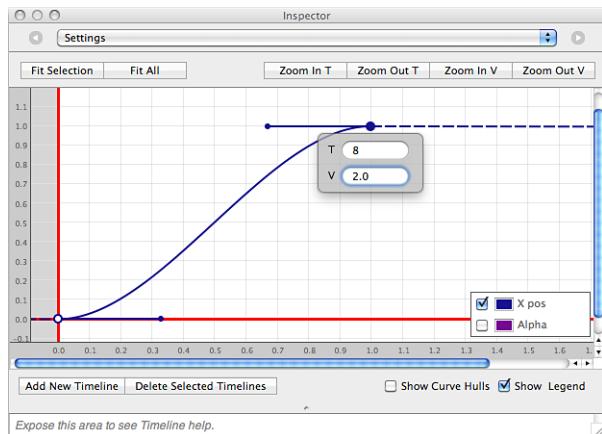
g Rename Timeline 2 to Alpha and press Enter.

h Double-click Timeline 1 and rename it X pos.

8 Configure the timeline of the Timeline patch:

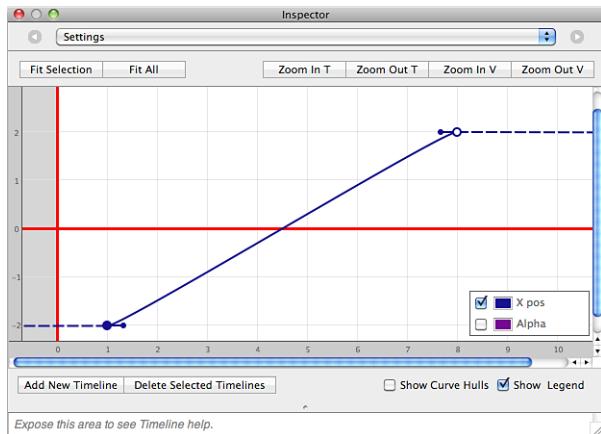
a In the timeline's legend, deselect Alpha and leave X pos selected.

b Double-click the point at (1.0, 1.0) and enter (8.0, 2.0):



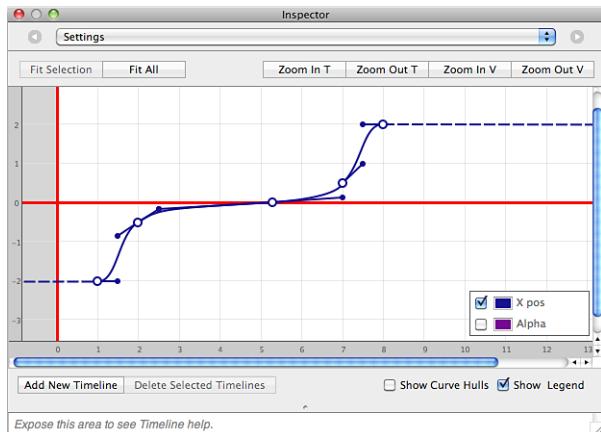
c Click Fit All.

- d Double-click the point at (0.0, 0.0) and enter (1.0, -2.0):



- e Click Fit All.

- f Add three points to the timeline and configure them to match the timeline below.

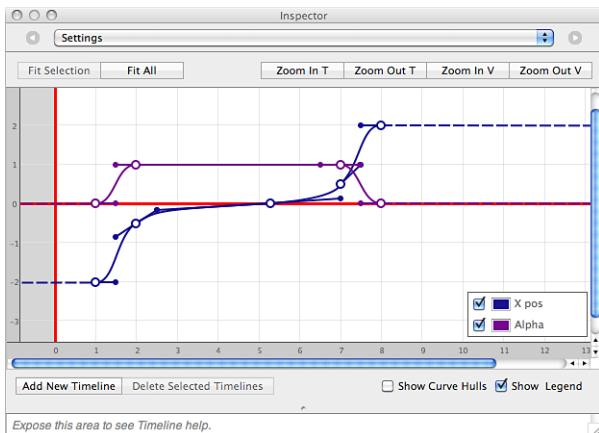


To add a point to the timeline, hold the Control, Option, and Command keys, and double-click anywhere on the timeline.

To add control handles to the point, Control-click the point while moving the mouse sideways. Use Control handles to smooth the timeline curve.

- g In the timeline's legend, select Alpha.

- h** Configure the Alpha timeline to match the Alpha timeline below.



- 9** Add input to the Timeline patch:

- Use the Patch Library to add a Patch Time patch to the Editor.
- Use the Patch Library to add a Math patch to the Editor.
- Select the Math patch and set the number of operations to 2 in the Settings pane of the Inspector.
- Connect the Time port of the Patch Time patch to the Initial Value port (first port) of the Math patch.
- Double-click the first operation and choose the division symbol from the pop-up menu.
- Connect the Duration port in the Bumper Title Protocol to the Operand #1 port (second port) of the Math patch.
- Double-click the second operation and choose the multiplication symbol from the pop-up menu.
- Double-click the Operand #2 port (third port) and enter 9.
- Connect the Resulting Value port (output port) of the Math patch to the Patch Time port of the Timeline patch.
- In the toolbar, click Parameters to display the Macro Patch pane.
- In the Macro Patch pane, enter 10 in the duration field.
- Connect the X pos port of the Timeline patch to the X Position port of the Billboard patch.
- In the Viewer window, click Stop and then Run to see how the title animation runs.

The title should fly in, hover for a few seconds, then fly out.

- 10** Add an RGB Color patch:
 - a** Use the Patch Library to add an RGB Color patch to the Editor.
 - b** Connect the Alpha port of the Timeline patch to the Alpha port of the RGB Color patch.
 - c** Connect the Color port of the RGB Color patch to the Color port of the Billboard patch.
 - d** Use the Viewer to preview the title animation.

In the timeline, which lasts about 9 seconds, the title starts at -2 (outside of the field of view) until 1 second, flies in from 1 to 3 seconds, hovers from 3 to 7 seconds, and then flies out.

- 11** Choose File > Save.
- 12** Quit Quartz Composer.

Step 2: Add the Composition to a Composition Bundle

After creating a composition using Quartz Composer, the next step is to add it to a composition bundle, which is required by Podcast Composer.

To add the BumperTitle.qtz composition to a composition bundle:

- 1** In /Applications/Server/, select Podcast Composer.
- 2** Control-click Podcast Composer and choose Show Package Contents.
- 3** Copy Contents/Resources/Materials/Compositions/Titles/Bumper/FlyingTitle.pqz.
- 4** Paste FlyingTitle.pqz into /Library/Application Support/Podcast Composer/Resources/Compositions/Titles/Bumper/.

This is the folder where you should store bumper title compositions to make them available to Podcast Composer. Create the folder path if necessary.

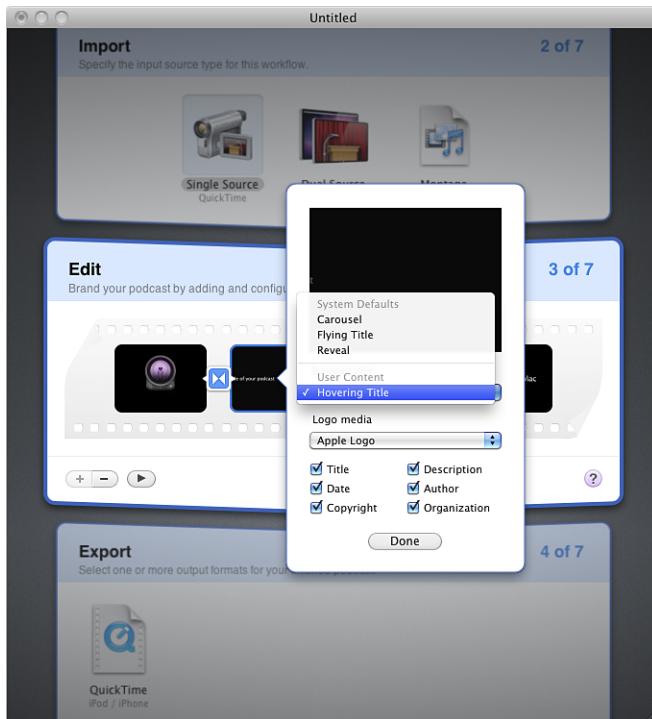
- 5** Rename the bundle from FlyingTitle.pqz to HoveringTitle.pqz.
- 6** Modify the bundle's contents as follows:
 - a** Delete the following:
 - Contents/Resources/de.lproj
 - Contents/Resources/fr.lproj
 - Contents/Resources/ja.lproj
 - Contents/Resources/BumperTitle.qtz
 - b** Move the BumperTitle.qtz composition you created earlier from the Desktop to Contents/Resources/.
 - c** Open Contents/Info.plist.

- d Change the last element in the value of the “Bundle identifier” key from FlyingTitle to HoveringTitle.
 - e Change the value of the “Bundle name” key to Hovering Title.
 - f Save Contents/Info.plist.
 - g Open Contents/Resources/en.lproj/InfoPlist.strings.
 - h Change the value of the Name and Description keys as follows:
- ```
Name = "Hovering Title";
Description = "Hovering title composition.";
```
- i Save and close Contents/Resources/en.lproj/InfoPlist.strings.

### Step 3: Restart Podcast Composer

To verify that you can access the new composition in Podcast Composer:

- 1 If opened, quit Podcast Composer.
- 2 Launch Podcast Composer.
- 3 In the Edit stage, double-click the title video clip.
- 4 Click the Bumper title pop-up menu and verify that you can choose Hovering Title in the User Content section of the menu:



# Building an Overlay Title Composition

4

In this chapter, you use Quartz Composer to create an overlay title composition and then make it available to Podcast Composer.

In this chapter, you use Quartz Composer to expand a sample overlay title composition and then make it available to Podcast Composer.

## Before You Begin

If you haven't done so, download the sample overlay title composition, as described in "Downloading the Workflow Tutorial Files" on page 11.

## Summary

The overlay composition you create in this chapter is similar to the composition you created in the previous chapter, except that the overlay composition doesn't generate a movie. Rather, the composition is laid over another movie.

The overlay composition conforms to Podcast Composer's Image Filter protocol, which requires at least an image as input and an image as output.

To create a title composition for use by Podcast Composer, perform the following tasks:

- 1 Use Quartz Composer to expand a sample overlay title composition (see "Step 1: Expand the Provided Overlay Title Composition" on page 30).
- 2 Store the composition bundle in the relevant folder (see "Step 2: Add the Composition to a Composition Bundle" on page 33).
- 3 Restart Podcast Composer and verify that you can access the new overlay title composition (see "Step 3: Restart Podcast Composer" on page 34).

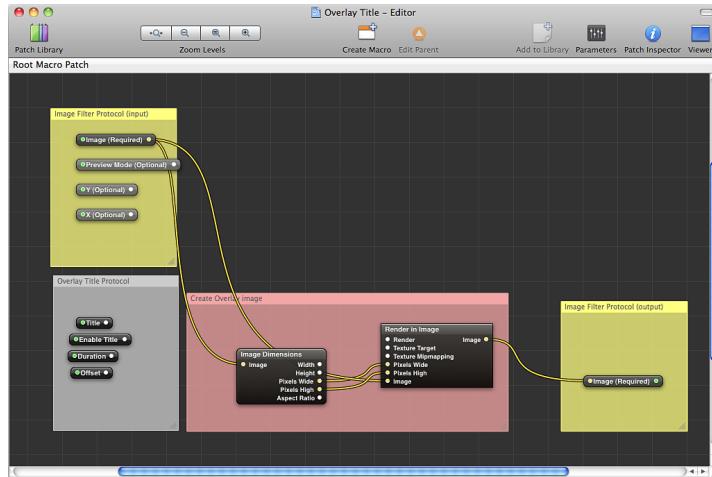
# Building an Overlay Title Composition and Making It Available to Podcast Producer

## Step 1: Expand the Provided Overlay Title Composition

To expand the provided sample overlay title composition:

- 1 Open the ~/Desktop/Podcast Producer Workflow Tutorial Files/Titles/Overlay/ folder.
- 2 Double-click Overlay Title Start.qtz.

The following screen appears.

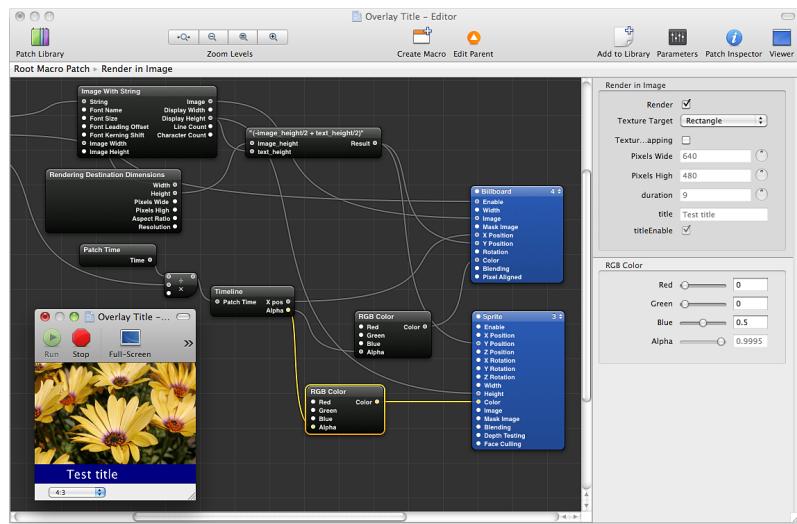


- 3 Choose File > Save As.
- 4 Save the composition as OverlayTitle.qtz on the Desktop.
- 5 Control-click the Render in Image patch and choose Edit Macro Patch.
- 6 Add to the Render in Image patch the patches to create an image from a title string:
  - a Open the BumperTitle.qtz composition you created in the last chapter.
  - b Copy all the contents of the composition, except the bumper title protocol ports and the Clear patch.
  - c Close the BumperTitle.qtz composition and don't save.
  - d Paste the patches you have copied inside the Render in Image patch of the overlay title composition.
  - e In the toolbar of the Editor, click Edit Parent.
  - f Copy the Title, Enable Title, and Duration input patches.
  - g Double-click the Render in Image patch to edit its contents.
  - h Paste the Title, Enable Title, and Duration input patches.

- i Connect the output ports of the Title, Enable Title, and Duration patches to the corresponding ports Image With String (String port), Billboard (enable port), and Math (Operand #1) patches, as in BumperTitle.qtz.
  - j In the toolbar of the Editor, click Edit Parent.
- 7 Connect the output port of the Title, Enable Title, and Duration patches to the corresponding ports in the Render in Image patch.
- 8 In the Viewer window, run the animation and verify that you can see the title flying in, hovering, and then flying out.
- If, instead of text, you see text on a black background flying in, change the value of the Blending property of the Billboard patch inside the Render in Image patch to Over.
- 9 Move the overlay title to the lower third of the screen:
- a Double-click the Render in Image patch to edit its contents.
  - b In the Patch Library, find the Mathematical Expression patch and add it to the Render in Image patch.
  - c With the Mathematical Expression patch selected, choose Editor > Show Patch Settings.
  - d Replace the existing expression with the following expression:  
$$(-\text{image\_height}/2 + \text{text\_height}/2)$$
  - e Connect the Height port of Rendering Destination Dimensions to the `image_height` port of the mathematical expression.
  - f Connect the Display Height port of Image With String to the `text_height` port of the mathematical expression.
  - g Connect the Result port of the mathematical expression to the Y Position port of Billboard.
- 10 Add background to the text:
- a In the Patch Library, find the Sprite patch and add it to the Render in Image patch.
  - b In the Sprite patch, click the top right pop-up menu and choose Layer 3.

You want the background to be behind the text.
  - c Connect the Enable Title port to the Enable port of the Sprite patch.
  - d In the Parameters pane of Sprite, set the Width parameter to 2 and the Blending parameter of the Sprite patch to Over.

- e Connect the Display Height port of Image With String to the Height port of Sprite.
- f Connect the Result port of the Mathematical Expression patch ("(-image\_height/2 + text\_height/2)") to the Y Position port of Sprite.
- g Using the Patch Library, add an RGB Color patch.
- h In the Parameters pane of RGB Color, change the values of the Red, Green, and Blue parameters to 0, 0, and 0.5 respectively.
- i Connect the Color port of the RGB Color patch to the Color port of the Sprite patch.
- j Select the Image With String patch and set its Image Height parameter to 0.2.
- k Connect the Alpha port of Timeline to the Alpha port of the RGB Color patch.
- l In the Viewer window, verify the animation works as expected:



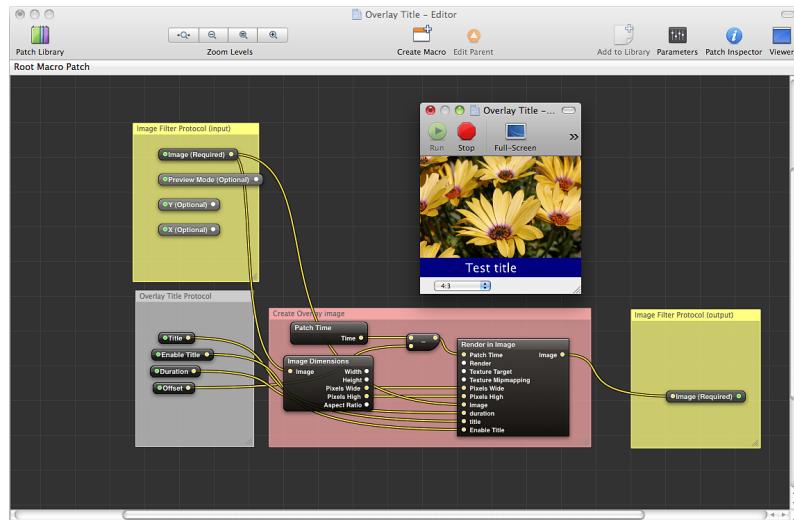
The blue background appears at the bottom on the window, with text flying in, hovering, and then flying out.

- 11 In the toolbar of the Editor, click Edit Parent.
- 12 Add an offset:
  - a Using the Patch Library, add a Patch Time patch and a Math patch.
  - b Change the operation of the Math patch to subtraction.
  - c Connect the Time port of the Patch Time patch to the Initial Value port of the Math patch.
  - d Connect the Offset port in the Overlay Title Protocol to the Operand #1 port of the Math patch.

- e Control-click Render in Image and choose Timebase > External from the pop-up menu.

This adds Patch Time as a parameter to Render in Image.

- f Connect the Result port of the Math patch to Patch Time of Render in Image.



13 Choose File > Save.

14 Quit Quartz Composer.

## Step 2: Add the Composition to a Composition Bundle

After creating the Overlay Title composition using Quartz Composer, the next step is to add it to a composition bundle, which is required by Podcast Composer.

**To add the OverlayTitle.qtz composition to a composition bundle:**

- 1 In /Applications/Server/, select Podcast Composer.
- 2 Control-click Podcast Composer and choose Show Package Contents.
- 3 Copy Contents/Resources/Materials/Compositions>Title/Overlay/LowerThirdTitle.pqz.
- 4 Paste LowerThirdTitle.pqz into /Library/Application Support/Podcast Composer/Resources/Compositions/Titles/Overlay/.

This is the folder where you should store overlay title compositions to make them available to Podcast Composer. Create the folder path if necessary.

- 5** Rename the bundle from LowerThirdTitle.pqz to OverlayTitle.pqz.
- 6** Modify the bundle's contents as follows:
  - a** Delete the following:
    - Contents/Resources/de.lproj
    - Contents/Resources/fr.lproj
    - Contents/Resources/ja.lproj
    - Contents/Resources/OverlayTitle.qtz
  - b** Move the OverlayTitle.qtz composition you created earlier from the Desktop to Contents/Resources/.
  - c** Open Contents/Info.plist.
  - d** Change the last element in the value of the "Bundle identifier" key from LowerThirdTitle to OverlayTitle.
  - e** Change the value of the "Bundle name" key to Overlay Title.
  - f** Save and close Contents/Info.plist.
  - g** Open Contents/Resources/en.lproj/InfoPlist.strings.
  - h** Change the value of the Name and Description keys as follows:

```
Name = "Overlay Title";
Description = "Overlay title composition.;"
```

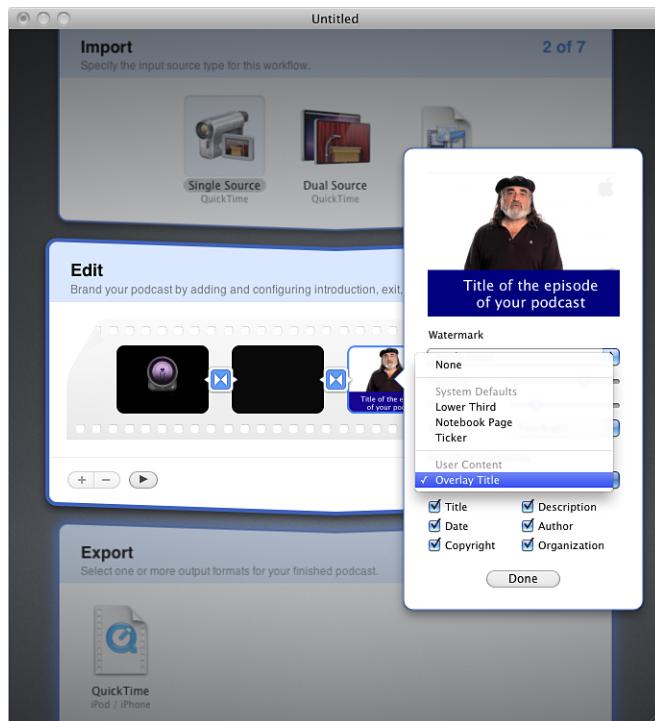
- i** Save and close Contents/Resources/en.lproj/InfoPlist.strings.

### Step 3: Restart Podcast Composer

To verify that you can access the new composition in Podcast Composer:

- 1** If opened, quit Podcast Composer.
- 2** Launch Podcast Composer.
- 3** In the Edit stage, double-click the input video clip.

- 4 Click the “Introduction overlay” pop-up menu and verify that you can choose Overlay Title in the User Content section of the menu:



# Building a Dual-Source Composition

In this chapter, you use Quartz Composer to create a composition that combines two video inputs into one movie.

In this chapter, you create a dual-source composition that renders one movie in the background and a second movie, one-fourth the size of the first movie, on top of the first movie at the lower right of the screen.

## Before You Begin

If you haven't done so, download the sample bumper title composition, as described in "Downloading the Workflow Tutorial Files" on page 11.

## Summary

To create a dual-source composition for use in Podcast Composer, perform the following tasks:

- 1 Use Quartz Composer to create a dual-source composition (see "Step 1: Create the Dual-Source Composition" on page 37).
- 2 Add the composition to a sample composition bundle and make it available to Podcast Composer (see "Step 2: Add the Composition to a Composition Bundle" on page 40).
- 3 Restart Podcast Composer and verify that you can access the new composition (see "Step 3: Restart Podcast Composer" on page 41).

# Building a Dual-Source Composition and Making It Available to Podcast Producer

## Step 1: Create the Dual-Source Composition

To create the dual-source composition:

- 1 Launch Quartz Composer (in /Developer/Applications/).
- 2 Click Templates, click Basic Composition, then click Choose.
- 3 Choose File > Save As.
- 4 In the Save As field, enter DualSource.qtz, choose Desktop as the destination, then click Save.

The name of any dual-source effect must always be DualSource.qtz if it's to be used in Podcast Composer.

- 5 Use the Patch Library to add the following patches:

- Two Input Splitter patches
- Two Movie Importer patches
- Two Billboard patches
- Two Math patches
- An Image Dimensions patch
- A Rendering Destination Dimensions patch
- An Anchor Position patch

- 6 Configure the Input Splitter patches:

- a Set the type of the two Input Splitter patches to String.

Select the patch, choose Editor > Show Inspector, choose Settings from the pop-up menu, then choose String from the Type pop-up menu.

- b Control-click one of the patches and choose Publish Inputs > Input.

- c Replace Input with primaryInput and press Enter.

The port changes color to green.

- d Control-click the second patch and choose Publish Inputs > Input.

- e Replace Input with secondaryInput and press Enter.

The port changes color to green.

- f Rename the two patches as primaryInput and secondaryInput.

To rename a patch, double-click its name, enter the new name, then press Enter.

- g In the toolbar of the Editor, click Parameters.

- h In the Input field of primaryInput, enter:

/Applications/Server/Podcast Composer.app/Contents/PlugIns/  
PictureInPictureImportPlugin.bundle/Contents/Resources/Primary.mov

This is a sample movie you use to verify that the composition works as intended. When you deploy this composition, this parameter holds the path of the input movie.

- i In the Input field of secondaryInput, enter:

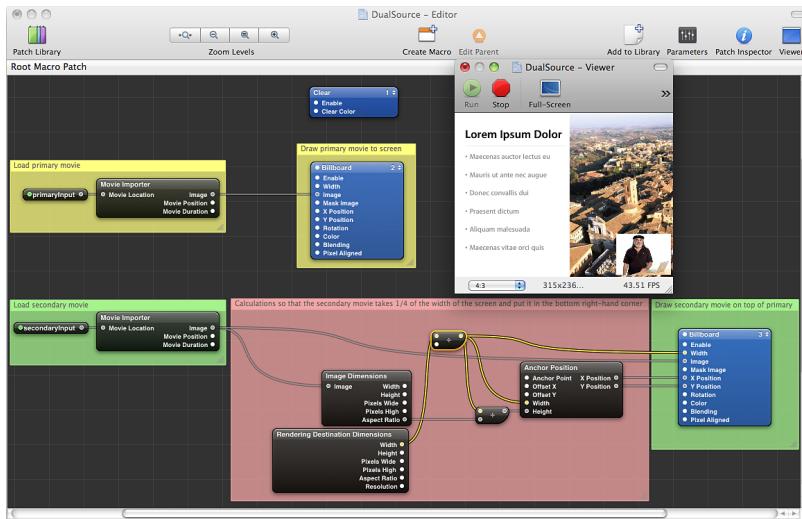
/Applications/Server/Podcast Composer.app/Contents/PlugIns/  
PictureInPictureImportPlugin.bundle/Contents/Resources/Secondary.mov

This is a sample movie you use to verify that the composition works as intended.

- 7 Connect the two video sources to the Billboard patches:
  - a Connect the output port of primaryInput to the Movie Location port of one of the Movie Importer patches.
  - b Connect the output port of the secondaryInput to the Movie Location port of the second Movie Importer patches.
  - c Connect the Image port of one of the Movie Importer patches to the Image port of one of the Billboard patches.
  - d Connect the Image port of the second Movie Importer patch to the Image port of the second Billboard patch.
- 8 Set the width of the two movies:
  - a Select the Billboard patch that renders the primary video source.
  - b In the Billboard pane, set Width to 2.

You can also use the Inspector to set the width.
  - c Set the Width parameter of the second Billboard patch to 0.5.
- 9 Connect the Image port of the Movie Importer patch that imports the second video source to the Image port of the Image Dimensions patch.
- 10 Connect the Width port of the Rendering Destination Dimensions patch to the Initial Value port of one of the Math patches.
- 11 Set the value of the second operand of the Math patch to 4 and change the operation to division.
- 12 Connect the output port of the Math patch to the following ports:
  - Initial Value port (first input port) of the second Math patch
  - Width port of the Anchor Position patch
  - Width port of the Billboard patch that renders the second movie

- 13 Configure the second Math patch:
  - a Set the operation to division.
  - b Connect the Aspect Ratio port of the Image Dimensions patch to the second port of the Math patch.
  - c Connect the output port of the Math patch to the Height port of the Anchor Position patch.
- 14 Connect the X Position and Y Position ports of the Anchor Position patch to the corresponding ports in the Billboard patch that renders the second movie.
- 15 Configure the Anchor Position patch parameters:
  - a Set the Anchor Point parameter to Bottom Right.
  - b Set the Offset X and Offset Y parameters to 0.02.
- 16 Verify that the animation works:



- 17 Choose File > Save.
- 18 Export a preview movie of the composition:
  - a Choose File > Export as QuickTime Movie.
  - b In the Export As field, enter Preview.mov.
  - c Choose Desktop from the pop-up menu.
  - d Click Save.
  - e In the Quartz Composer Composition Export window, set the dimensions to 640 x 480 and the duration to 30 seconds.

- f** Click Export.
- g** Quit Quartz Composer.

## Step 2: Add the Composition to a Composition Bundle

After creating the composition using Quartz Composer, the next step is to add it to a composition bundle, which is required by Podcast Composer.

### To add the DualSource.qtz composition to a sample dual-source bundle:

- 1** In /Applications/Server/, select Podcast Composer.
- 2** Control-click Podcast Composer and choose Show Package Contents.
- 3** Copy Contents/Resources/Materials/Compositions/DualSources/Overlay.pqz.
- 4** Paste Overlay.pqz into /Library/Application Support/Podcast Composer/Resources/Compositions/DualSources/.

This is where you should store dual source compositions to make them available to Podcast Composer. If necessary, create the folder path.

- 5** Rename the bundle from Overlay.pqz to DualSource.pqz.
- 6** Modify the bundle's contents as follows:
  - a** Delete the following:
    - Contents/Resources/de.lproj
    - Contents/Resources/fr.lproj
    - Contents/Resources/ja.lproj
    - Contents/Resources/DualSource.qtz
    - Contents/Resources/Preview.com
  - b** Move the dual-source composition you created earlier from the Desktop to Contents/Resources/.
  - c** Move Preview.mov from the Desktop to Contents/Resources/.
  - d** Open Contents/Info.plist.
  - e** Change the value of the last element of the "Bundle identifier" key to DualSource.
  - f** Change the value of the "Bundle name" key to Dual Source.
  - g** Save and close Contents/Info.plist.
  - h** Open Contents/Resources/en.lproj/InfoPlist.strings.

- i** Change the value of the Name and Description keys as follows:

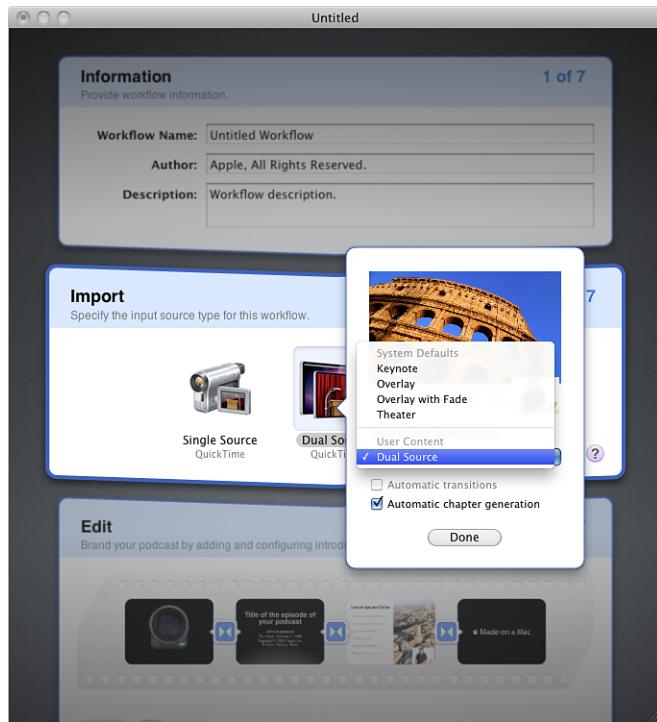
```
Name = "Dual Source";
Description = "Dual-source composition.";
```

- j** Save and close Contents/Resources/en.lproj/InfoPlist.strings.

## Step 3: Restart Podcast Composer

To verify that you can access the new dual source composition in Podcast Composer:

- 1 If opened, quit Podcast Composer.
- 2 Launch Podcast Composer.
- 3 In the Import stage, double-click Dual Source.
- 4 Click the Dual Source composition pop-up menu and verify that you can choose Dual Source in the User Content section:



# Building a Custom Compressor Encoder

In this chapter, you learn how to create a Compressor encoder and make it available to Podcast Composer.

Compressor is an application that is included with Final Cut Studio 3. In this chapter, you use Compressor to create a custom encoder, which becomes available to Podcast Composer the next time you launch Podcast Composer.

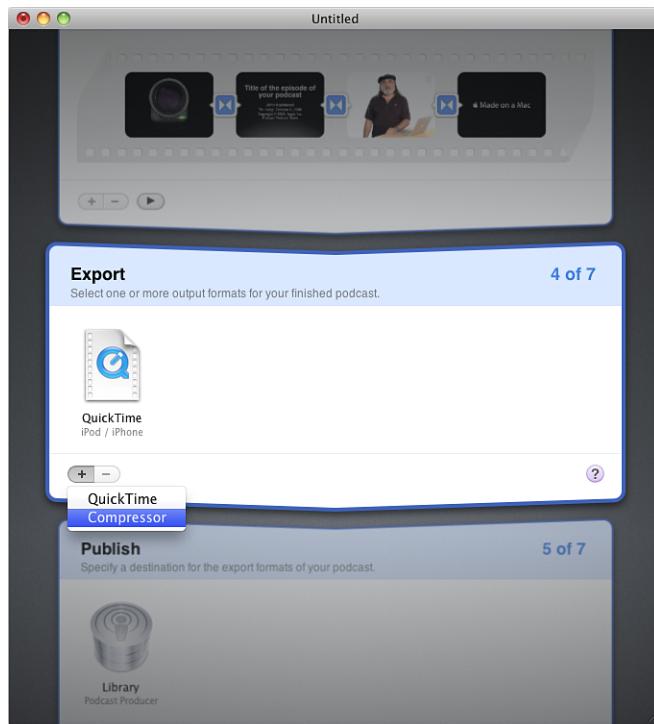
## Before You Begin

If the Compressor application is not installed on your computer, obtain and install Final Cut Studio 3. Compressor is installed by default when you install Final Cut Studio 3.

With Compressor installed, make sure that Podcast Composer has access to Compressor encoders.

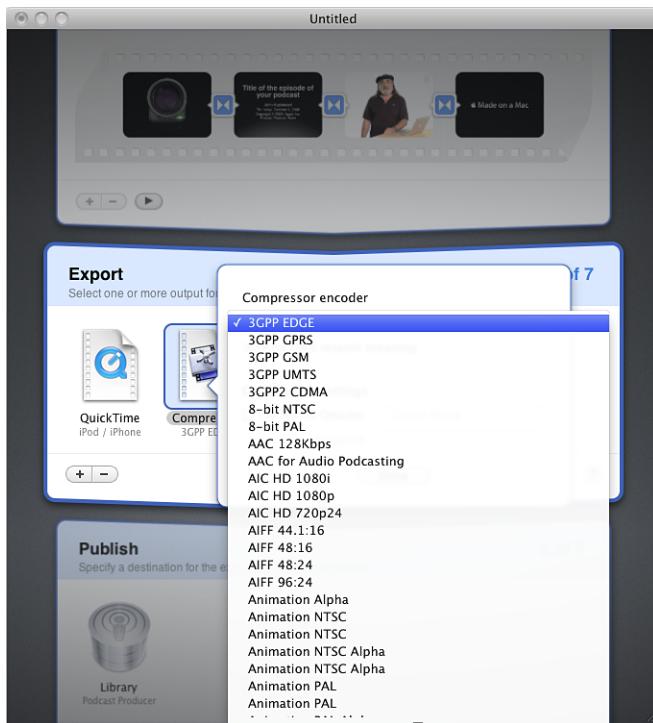
To check whether Podcast Composer has access to Compressor encoders:

- 1 Launch Podcast Composer.
- 2 In the Export stage (stage 4 of 7), click the Add (+) button and choose Compressor from the pop-up menu:



If you don't see Compressor listed in the pop-up menu, Compressor is not installed on your system.

- 3 Click the “Compressor encoder” pop-up menu to make sure that you have access to the encoders provided by Compressor:



- 4 Quit Podcast Composer.

## Summary

To create a custom encoder and make it available to Podcast composer, perform the following tasks:

- 1 Use Compressor to create the encoder (see “Step 1: Create a Custom Composer Encoder” on page 44).
- 2 Restart Podcast Composer and verify that the new encoder can be selected (see “Step 2: Verify that You Can See the New Encoder in Podcast Composer” on page 46).

## Creating a Compressor Encoder for Use in Podcast Composer

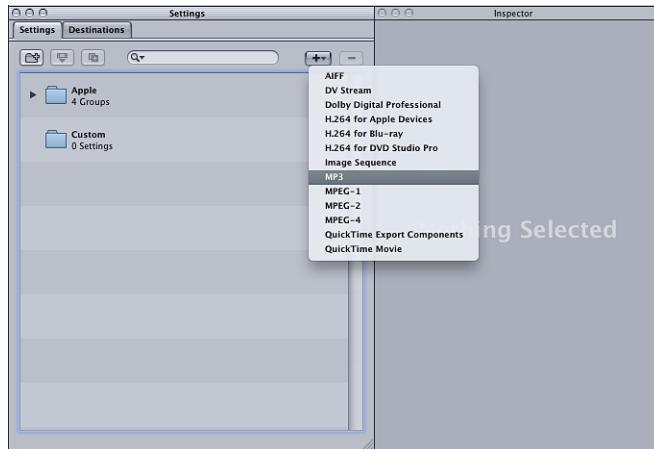
### Step 1: Create a Custom Composer Encoder

Use Compressor to create a custom encoder.

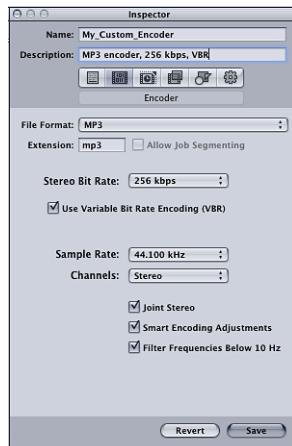
**To create a custom encoder using Compressor:**

- 1 Launch Compressor.

- 2 If prompted to choose a template for your batch, click Cancel.
- 3 Click the Settings tab.
- 4 Click the Add (+) button and choose the MP3 encoding category:



- 5 In the Inspector window, configure the settings of the encoder you are creating.
  - a In the Name field, enter "My\_Custom\_Encoder."
  - b From the Stereo Bit Rate pop-up menu, choose 256 kbps.
  - c Select "Use Variable Bit Rate Encoding (VBR)."
  - d In the Description field, enter a description of the encoder.



- e Click Save.

By saving the new encoder in Compressor, it becomes available to Podcast Composer. If Podcast Composer is running, you need to restart it to see the new encoder.

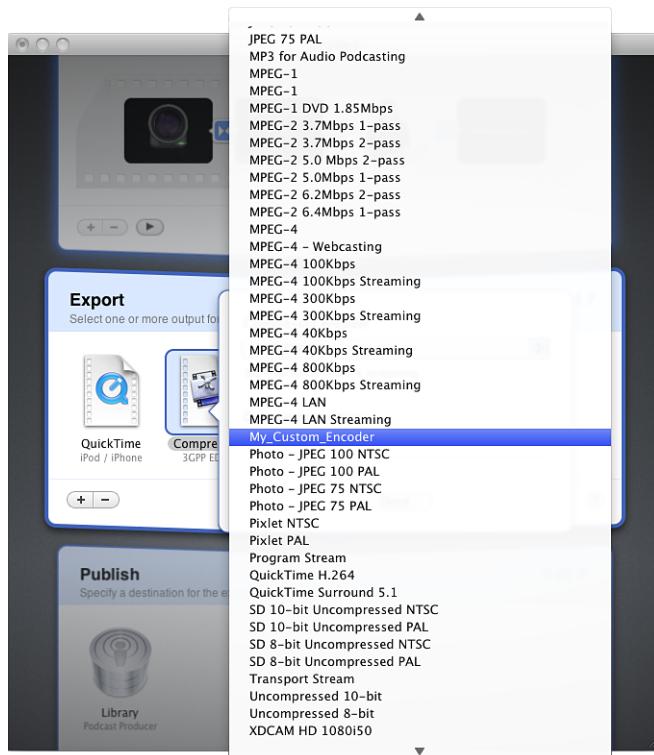
- 6 Quit Compressor.

## Step 2: Verify that You Can See the New Encoder in Podcast Composer

Restart Podcast Composer and add the new encoder to the Export stage.

To verify that the new encoder is available to Podcast Composer:

- 1 If Podcast Composer is running, quit it and start it again.
- 2 In the Export stage (stage 4 of 7), click the Add (+) button and choose Compressor from the pop-up menu.
- 3 Click the Compressor encoder pop-up menu and choose My\_Custom\_Encoder:



- 4 Quit Podcast Composer.

# Creating a Transcription Workflow

7

In this chapter, you learn how to manually expand a workflow created with Podcast Composer. You also learn how to use the new Xgrid retry functionality for polling.

In this chapter, you use Podcast Composer to create a basic single source workflow. Then, you manually expand the workflow to submit a low resolution version of the podcast to a sample transcription service you create, use the Xgrid retry functionality to check for the finished subtitle files, integrate the transcribed text into the podcast, and publish the podcast.

## Before You Begin

If you haven't done so, download the transcription service files, as described in "Downloading the Workflow Tutorial Files" on page 11.

## Summary

To create the Transcription workflow, perform the following tasks:

- 1 Create a basic workflow using Podcast Composer (see "Step 1: Create a Basic Workflow" on page 48).
- 2 Manually customize the workflow (see "Step 2: Manually Customize the Workflow" on page 48).
- 3 Deploy the custom scripts referenced in the workflow (see "Step 3: Deploy Custom Scripts" on page 56).
- 4 Set up a sample transcription service (see "Step 4: Set Up a Transcription Service" on page 57).
- 5 Test the workflow (see "Step 5: Test the Workflow" on page 58).

## Creating and Testing the Workflow

### Step 1: Create a Basic Workflow

Use Podcast Composer to create a basic workflow that generates a single-source podcast.

To create the workflow:

- 1 Launch Podcast Composer.
- 2 In Stage 1, enter the following information:
  - Workflow Name—Transcription
  - Author—*Your name*
  - Description—*A brief description*
- 3 In the Import stage, select Single Source.
- 4 In the Export stage, export to QuickTime Mobile and to QuickTime iPod / iPhone.
- 5 (Optional) In the Notify stage, add an iChat notification.  
Make sure you provide the iChat Sender information.
- 6 Verify the workflow (File > Verify Workflow).
- 7 Save the workflow to the Desktop as Transcription.

### Step 2: Manually Customize the Workflow

In this section, you modify the template.plist file to add the functionality needed to generate a podcast with transcribed subtitles.

**Important:** When adding XML code to the workflow, use the provided workflow (Single Source Transcription.pwf) to copy and paste the XML code into your workflow.

Customizing the workflow involves the following:

- “Step A: Update One of the QuickTime Export Tasks” on page 49
- “Step B: Add a Task to Submit the Input Movie to the Transcription Service” on page 50
- “Step C: Add a Task that Polls the Transcription Service and Downloads the Transcriptions” on page 51
- “Step D: Add the Task that Adds Subtitles to the Movie” on page 52
- “Step E: Add a Task that Flattens the Reference Movie Created by the add\_subtitle Task” on page 53
- “Step F: Update the Publishing Task” on page 53
- “Step G: (Optional) Update the Notification Job” on page 54

## Step A: Update One of the QuickTime Export Tasks

In this section, you modify an encoding task to encode only the input movie and generate a low-resolution version of the input movie to be used for transcription. In addition, you modify this task to run as early as possible.

### To export a low-resolution version of the input movie:

- 1 In ~/Desktop, Control-click the Transcription workflow and choose Show Package Contents.
- 2 Open the template.plist file using your favorite XML editor, such as Dashcode or Xcode (in /Developer/Applications/).

The template.plist file is in ~/Desktop/Transcription.pwf/Contents/Resources/.

- 3 Locate the encoding task that uses the "iphone\_cellular.plist" encoder.

The task name starts with "export-plugin-quicktime" and the encoder argument contains "iphone\_cellular.plist."

- 4 Edit the task to appear as the following:

```
<!-- Encode original to low quality for the transcription service. -->
<!-- Task was generated by Podcast Composer but was manually modified.
-->
<key>export-plugin-quicktime-encode-low-res-version-for-transcription-service</key>
<dict>
 <key>arguments</key>
 <array>
 <string>encode</string>
 <string>--prb=$$GLOBAL::Library Bundle Path$$</string>
 <string>--input=$$SOURCE::0::Filename$$</string>
 <string>--output=$$SOURCE::0::Basename$$-low-res.mp4</string>
 <string>--encoder=$$GLOBAL::Workflow Resource Path$$/Presets/
mp4_low.plist</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>preflight</string>
 </array>
</dict>
```

In the code above, the text in black represents the changes to the existing task:

- Instead of encoding the movie resulting from the Edit stage, you specify the input movie as the source.

- For output, you change the name to reflect MPEG-4 encoding.
  - For the encoder, you specify mp4\_low.plist, which includes the settings for low resolution encoding.
  - Rather than waiting until the `edit-core-annotate` task completes running, you configure this task to run after the `preflight` task completes running.
- 5 Search the file and replace any reference to the old name of the task with the new name.
  - 6 Save the file.

### Step B: Add a Task to Submit the Input Movie to the Transcription Service

After encoding a low-resolution version of the input movie, add a task to submit the low-resolution movie to the transcription service.

**To submit the input movie to the transcription service:**

- 1 Add the following task to the taskSpecification dictionary in the template.plist file of the Transcription workflow:

```
<!-- Submit low-res file to transcription service. -->
<!-- This task was manually added. -->
<key>upload-to-transcription-service</key>
<dict>
 <key>arguments</key>
 <array>
 <string>shell</string>
 <string>--basedir=$$Base Directory$$</string>
 <string>--</string>
 <string>$$Workflow Resource Path$$/Tools/upload_to_transcription_
service.rb</string>
 <string>$$SOURCE::0::Basename$$-low-res.mp4</string>
 <string>$$SFTP Hostname$$</string>
 <string>$$SFTP Incoming Folder Path$$</string>
 <string>$$SFTP Username$$</string>
 <string>$$Global Resource Path$$/id_rsa</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>export-plugin-quicktime-encode-low-res-version-for-
transcription-service</string>
 </array>
</dict>
```

This task executes the `upload_to_transcription_service.rb` script, which submits the low-resolution movie to the transcription service.

For more information about this script, see “`upload_to_transcription_service.rb`” on page 56.

- 2 Save the file.

### Step C: Add a Task that Polls the Transcription Service and Downloads the Transcriptions

To add the polling and transcription downloading task:

- 1 Add the following task to the `taskSpecification` dictionary in the `template.plist` file of the Transcription workflow:

```
<!-- Download the subtitle file(s) from the transcription service -->
<!-- This task knows how to tell Xgrid to retry the task after a certain
 amount of time -->
<!-- if the subtitle files aren't yet available using the Xgrid retry
 mechanism. -->
<!-- (See the download_subtitles.rb script for more) -->
<!-- This task was manually added -->

<key>download-subtitles</key>
<dict>
 <key>arguments</key>
 <array>
 <string>shell</string>
 <string>--basedir=$$Base Directory$$</string>
 <string>--</string>
 <string>$$Workflow Resource Path$$/Tools/download_subtitles.rb</
 string>
 <string>$$SOURCE::0::Basename$$-low-res.mp4</string>
 <string>$$Base Directory$$</string>
 <string>$$SFTP Hostname$$</string>
 <string>$$SFTP Outgoing Folder Path$$</string>
 <string>$$SFTP Username$$</string>
 <string>$$Global Resource Path$$/id_rsa</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>upload-to-transcription-service</string>
 <string>export-plugin-quicktime-encode-ipod-version</string>
 </array>
```

```
</dict>
```

This action executes the download\_subtitles.rb script, which retrieves the subtitle files from the transcription service.

For more information about download\_subtitles.rb, see “download\_subtitles.rb” on page 56.

**2** Save the file.

### Step D: Add the Task that Adds Subtitles to the Movie

To add the task that adds subtitles to the movie:

- 1 Add the following task to the taskSpecification dictionary in the template.plist file of the Transcription workflow:

```
<!-- Add the subtitles to the iPod encoded movie -->
<!-- This task was manually added -->
<key>add-subtitles</key>
<dict>
 <key>arguments</key>
 <array>
 <string>shell</string>
 <string>--basedir=$$Base Directory$$</string>
 <string>--</string>
 <string>$$Workflow Resource Path$$/Tools/add_subtitles.rb</string>
 <string>$$Base Directory$$</string>
 <string>$$SOURCE::0::Basename$$-ipod.m4v</string>
 <string>$$SOURCE::0::Basename$$-low-res.mp4</string>
 <string>$$SOURCE::0::Basename$$-ipod-with-subtitles.mov</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>download-subtitles</string>
 <string>export-plugin-quicktime-UUID-encode</string>
 </array>
</dict>
```

This action executes the add\_subtitles.rb script, which retrieves the subtitle files from the transcription service.

For more information about add\_subtitles.rb, see “add\_subtitles.rb” on page 56.

- 2 Replace *UUID* with the UUID of the task that exports the edit-core-master.mov.
- 3 Save the file.

### Step E: Add a Task that Flattens the Reference Movie Created by the add\_subtitle Task

The movie generated by the add-subtitles task is a reference movie. In this section, you add a task that generates a flat movie.

#### To flatten the reference movie generated by the add\_subtitle task:

- 1 Add the following task to the taskSpecification dictionary in the template.plist file of the Transcription workflow:

```
<!-- Flattens the reference movie created by the add_subtitle task -->
<!-- This task was manually added -->
<key>flatten</key>
<dict>
 <key>arguments</key>
 <array>
 <string>flatten</string>
 <string>--prb=$$GLOBAL::Library Bundle Path$$</string>
 <string>--input=$$SOURCE::0::Basename$$-ipod-with-subtitles.mov</string>
 <string>--output=$$SOURCE::0::Basename$$-ipod-with-subtitles.m4v</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>add-subtitles</string>
 </array>
</dict>
```

- 2 Save the file.

### Step F: Update the Publishing Task

After flattening the movie, the next step is to publish the movie to the Podcast Library.

Your workflow already includes a publishing task, but you must modify it so that it publishes the flattened movie.

#### To update the publishing task:

- 1 Locate the publish-plugin-podcastlibrary-edit-core-master task and update it as shown here.

The text in black indicates the changes you must make to the code.

```
<!-- Publish the iPod encoded version to the Podcast library. -->
<!-- Task was generated by Podcast Composer, but then modified
manually.-->
<key>publish-plugin-podcastlibrary-export-plugin-quicktime-ipod-with-
subtitles</key>
<dict>
 <key>arguments</key>
 <array>
 <string>publish2library</string>
 <string>--prb=$$GLOBAL::Library Bundle Path$$</string>
 <string>--file= $$SOURCE::0::Basename$$-ipod-with-subtitles.m4v</
 string>
 <string>--outfile=publish-plugin-podcastlibrary-export-plugin-
quicktime-ipod-with-subtitles.yaml</string>
 <string>--type=production</string>
 <string>--mimetype=video/x-m4v</string>
 </array>
 <key>command</key>
 <string>/usr/bin/pcastaction</string>
 <key>dependsOnTasks</key>
 <array>
 <string>flatten</string>
 </array>
</dict>
```

- 2 Save the file.

### Step G: (Optional) Update the Notification Job

The last modification to the workflow is to update the notification task to include links to the movie that contains the transcribed subtitles.

#### To modify the notification task:

- 1 Locate the notification task (its name starts with “notify-plugin-ichat”) and edit it as indicated by the text in black.

```
<!-- Notify via iChat -->
<!-- Task was generated by Podcast Composer but then modified manually-->
<key>notify-plugin-ichat</key>
<dict>
 <key>arguments</key>
 <array>
 <string>jabber</string>
 <string>--prb=$$GLOBAL::Library Bundle Path$$</string>
```

```

<string>--jabberid=$$Administrator Jabber ID$$</string>
<string>--password=$$Administrator Jabber Password$$</string>
<string>--no_fail</string>
<string>--message=$$Title$$ has been
transcribed.</string>
<string>--to=$$User Jabber ID$$</string>
<string>--send_publishing_information</string>
<string>--publish_info=publish-plugin-podcastlibrary-import-
plugin-movie-original.yaml</string>
<string>--publish_info=publish-plugin-podcastlibrary-edit-core-
master.yaml</string>
<string>--publish_info=publish-plugin-podcastlibrary-export-
core-
preview-image.yaml</string>
<string>--publish_info=publish-plugin-podcastlibrary-export-
plugin-quicktime-ipod-with-subtitles.yaml</string>
</array>
<key>command</key>
<string>/usr/bin/pcastaction</string>
<key>dependsOnTasks</key>
<array>
<string>publish-plugin-podcastlibrary-import-plugin-movie-
original</string>
<string>publish-plugin-podcastlibrary-edit-core-master</string>
<string>publish-plugin-podcastlibrary-export-core-preview-image</
string>
<string>publish-plugin-podcastlibrary-export-plugin-quicktime-
ipod-with-subtitles</string>
</array>
</dict>

```

- 2** Save the file.

## Step 3: Deploy Custom Scripts

In this section, you add the custom scripts referenced in the workflow to the workflow bundle. You also add the mp4\_low.plist file, which contains the presets for generating a low resolution podcast.

### To deploy custom scripts:

- 1 If you haven't done so, download the transcription tutorial files, as described in "Downloading the Workflow Tutorial Files" on page 11.
- 2 Copy the following files and folders from the Podcast Producer Workflow Tutorial Files/Transcription Service Files/ folder to the Contents/Resources/Tools/ folder of the workflow's bundle:
  - add\_subtitles\_sources/
  - add\_subtitles.rb
  - addSubtitles
  - download\_subtitles.rb
  - upload\_to\_transcription\_service.rb
- 3 Copy the ~/Desktop/transcription\_service\_fles/Presets/mp4\_low.plist file to Contents/Resources/Presets/ folder of the workflow's bundle.

### **add\_subtitles.rb**

This script adds subtitles to the input movie.

### **addSubtitles**

This is an executable file that is called by add\_subtitles.rb to extract subtitles from the XML files provided by the transcription service and then add the subtitles to the input movie.

### **download\_subtitles.rb**

This script uses the new Xgrid retry functionality for polling the transcription service to check whether the XML files containing the transcription are ready for downloading.

### **upload\_to\_transcription\_service.rb**

This script uploads the low-resolution version of the source movie.

## Step 4: Set Up a Transcription Service

To set up the transcription service:

1 In Server Admin, go to the Podcast Producer > Settings > Properties pane.

2 Add the following properties to the Custom Properties list and provide the corresponding values:

- SFTP Hostname

This is the SFTP server you post files to and retrieve subtitles from.

By default, Mac OS X Server v10.6 includes an SFTP server. Either use this server or set up an SFTP server on another system and provide the hostname.

To use the SFTP server running on your system, make sure that Remote Login in System Preferences > Sharing is enabled.

- SFTP Username

This is the SFTP username.

**Important:** Make sure that this SFTP user account has read and write privileges to the incoming and outgoing folders on the SFTP server.

- SFTP Incoming Folder Path

This is the path of the incoming folder where you post files to.

- SFTP Outgoing Folder Path

This is the path of the outgoing folder where you download subtitles from.

3 (Optional) If you have Jabber set up on a computer running Mac OS X v10.6, add the following properties to the Custom Properties list in Server Admin and provide the corresponding values:

- Administrator Jabber ID

This is the Jabber ID of the user sending the notification. For best results, create a user just for this.

- Administrator Jabber Password

This is the password for the user sending the notification.

**Note:** In the sample code used in this chapter, the password is sent in the clear.

- User Jabber ID

This is the Jabber ID of the user to be notified (but it can be a list of Jabber IDs separated by commas).

4 On the Podcast Producer server, create a private/public SSH key pair and copy them to the server's resource folder by doing the following:

```
$ mkdir keys
```

```
$ cd keys
```

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Volumes/Data/kjell/.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa.

.
.
.
Your public key has been saved in ./id_rsa.pub.
```

```
$ ls
id_rsa id_rsa.pub
$ sudo cp id_rsa* /Library/PodcastProducer/Shared/Server/Resources/
```

- 5 Restart the Podcast Producer service to synchronize the resource caches to the shared file system:

```
$ sudo pcastctl server restart
```

- 6 Copy the public key to the SFTP server and add it to the authorized keys file:

```
$ scp id_rsa.pub USERNAME@SFTP_SERVER:~/id_rsa.pub
$ ssh USERNAME@SFTP_SERVER
$ cat id_rsa.pub >> .ssh/authorized_keys
$ exit
```

- 7 Copy the public key to the SFTP server and add it to the authorized keys file:

```
$ ssh -i id_rsa USERNAME@SFTP_SERVER
```

- 8 Install the workflow on your Podcast Producer server:

```
$ podcast --installworkflow --path ~/Desktop/Transcription.pwf --server
HOSTNAME --user ADMIN_USER --pass PASSWORD
```

If you must reinstall the workflow, use the `--overwrite` argument.

## Step 5: Test the Workflow

To test the workflow:

- 1 Use Podcast Capture, the Podcast Capture web application, or the `podcast` command to submit a video to the Transcription workflow.

The workflow encodes a low resolution version of the movie, creates a submission file (.sub), and uploads them to the incoming folder on your SFTP server.

For example, if you submit test.mov to the workflow, you see `test-UUID-low-res.mp4` and `test-UUID-low-res.sub` (where `UUID` is a unique identifier generated by Podcast Producer). In some cases, the name of the input movie is not included. In these cases, the names will be `UUID-low-res.mp4` and `UUID-low-res.sub`.

The workflow continues its tasks until it is blocked waiting for the subtitles to be submitted by the transcription service.

If you monitor the job progress in Xgrid Admin, you notice that CPU usage remains at 0% while the workflow is waiting for the subtitle files. However, after the *UUID-low-res.done* file appears in the outgoing folder, the job resumes running and downloads all the files with a name similar to *UUID-low-res\*.3gp.xml*.

For example, there can be two subtitle files: one for English *UUID-low-res.eng.3gp.xml* and another for French and *UUID-low-res.fra.3gp.xml*.

After the job downloads the subtitles, it adds them to the encoded video, publishes them to Podcast Library, and, if configured, notifies the Jabber user specified in the Custom Properties list.

**2** Add subtitles to the provided *UUID.eng.3gp.xml* file.

This file uses the QuickTime TeXML subtitle format. For more information about this format, see [http://developer.apple.com/mac/library/documentation/QuickTime/QT6\\_3/QT6\\_3WhatsNew.pdf](http://developer.apple.com/mac/library/documentation/QuickTime/QT6_3/QT6_3WhatsNew.pdf).

**3** Rename the file as *UUID-low-res.eng.3gp.xml*.

For example, if the name of the low-resolution movie in the incoming folder is *test-9B4F4A91-5886-4D85-BF92-B0462B976EEC-low-res.mp4*, the name of the subtitles file is *test-9B4F4A91-5886-4D85-BF92-B0462B976EEC-low-res.eng.3gp.xml*.

If the name of the low-resolution movie does not include the name of the submitted movie (*UUID-low-res.mp4*), the name of the subtitles file is *UUID-low-res.eng.3gp.xml*.

**4** Place the file in the outgoing folder of the SFTP server.

**5** Create a file usingTextEdit and save it as *name\_of\_input\_movie-UUID-low-res.done* or *UUID-low-res.done* in the outgoing folder.

If the name of the low-resolution movie does not include the name of the submitted movie (*UUID-low-res.mp4*), the name of the .done file is *UUID-low-res.done*.

**Important:** Make sure that the .done file you create doesn't have a different hidden extension.

**6** When the workflow completes, verify that the subtitles have been added to the input movie.

Open the podcast using iTunes and enable subtitles. In iTunes, choose Controls > Audio & Subtitles > English or choose English from the Subtitles pop-up menu on the Control bar.