




Mac OS X Server

Podcast Producer
Administration
For Version 10.5 Leopard

 Apple Inc.
© 2007 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 ensure that the information in this manual is accurate. Apple Inc. is not responsible for printing or clerical errors.

Apple
1 Infinite Loop
Cupertino CA 95014-2084
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, AirPort, AppleScript, FireWire, iMac, iMovie, iTunes, Mac, the Mac logo, Macintosh, Mac OS, PowerBook, Power Mac, QuickTime, 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 of these products.

019-0942/2007-09-01

Contents

Preface	7 About This Guide
	7 What's in This Guide
	8 Using Onscreen Help
	8 Mac OS X Server Administration Guides
	10 Viewing PDF Guides on Screen
	10 Printing PDF Guides
	10 Getting Documentation Updates
	11 Getting Additional Information
Chapter 1	13 Overview of Podcast Producer
	13 How Podcast Producer Works
	14 The Architecture of the Podcast Producer System
	15 Podcast Producer Server
	15 Podcast Capture
	15 The podcast Command-Line Tool
	16 Podcast Producer Agent
	16 Shared File System
	16 Workflows
	17 Xgrid
	17 The Podcast Producer Security Model
	18 Client/Server Communication
	18 Authentication
	18 Xgrid
	19 Publishing
	19 Podcast Producer Customization
	20 Supported Audio and Video Formats
Chapter 2	21 Setting Up Podcast Producer
	21 Hardware and Software Requirements for Podcast Producer
	21 Podcast Producer Agent Computer Requirements
	21 Podcast Server Computer Requirements
	22 Xgrid Size and Bandwidth Considerations
	22 Before You Set Up Podcast Producer

22	Starting Required Services
23	Configuring the DNS Service
23	Configure Mail
24	Configuring the Open Directory Service
24	Creating Required Users
25	Configuring Web Service
25	Creating a Group Blog
26	Configuring Xgrid Service
26	Configuring QuickTime Streaming Service
26	Setting Up Podcast Producer
26	Configuring General Settings
28	Configuring the Default Workflow Settings
28	Starting the Podcast Producer Server
29	Binding a Mac to the Podcast Producer Service
30	Verifying Your Setup

Chapter 3


31	Managing Workflows
31	Controlling Access to Workflows in Podcast Capture
32	Monitoring Workflow Usage
32	Filtering Workflows
32	Displaying Workflow Information
33	Configuring Workflow Properties
33	Configuring Default Workflow Properties
35	Configuring Custom Workflow Properties

Chapter 4

37	Managing Cameras
37	Controlling Access to Cameras in Podcast Capture
38	Removing Cameras
38	Filtering Cameras

Chapter 5

39	Customizing Workflows
39	The Structure of a Workflow Bundle
40	Global Encodings and Resources
41	The Structure of a Workflow
42	Workflow Task Specifications
43	Property Keys
45	Task Dependencies
46	Workflow Commands
47	Podcast Producer Default Workflows
48	Blog without archive
49	Blog
50	Blog and iTunes with watermark
51	Blog and iTunes with intro

- 52 Blog and iTunes with intro and effects
- 53 Blog and iTunes with intro and effects for 
- 54 iTunes U
- 55 iTunes U with intro
- 56 Blog with streaming and no archive
- 57 Blog with streaming
- 58 Blog with streaming and intro
- 59 Blog with streaming, intro and effects
- 60 Submit for Approval
- 60 Customizing Workflows
 - 60 Tools for Editing Workflows
 - 61 Modifying Workflow Resources
 - 61 Adding Custom Properties
 - 62 Duplicating and Modifying Workflows
 - 62 Adding Localized Strings

Chapter 6

- 63 **The Podcast Producer Shared File System**
- 63 Specifying a Shared File System
- 63 The Structure of the Shared File System
- 64 Using an NFS Shared File System

Chapter 7

- 65 **Deploying Scalable Podcast Producer Solutions**
- 65 Resource Planning
 - 66 Manual Submission Systems
 - 66 Video Recording Systems
 - 66 Recording Quality
 - 68 Network Bandwidth
 - 68 Publishing Systems
 - 68 Storage
 - 68 Xgrid Agents
 - 69 Workflows
- 69 Workflow Benchmarking
- 69 Deployment Scenarios
 - 69 Small Deployment
 - 70 Partially Scalable Deployment
 - 71 Highly Scalable Deployment
- 71 Case Study
 - 72 Recording System Configuration
 - 72 Workflow Benchmarks
 - 73 Recording Schedule
 - 73 Performance
 - 75 Summary

Chapter 8	77	Podcast Producer Command-Line Tools
	77	The podcast Tool
	78	The pcastconfig Tool
	78	The pcastctl Tool
	78	The pcastaction Tool
Chapter 9	79	Monitoring Podcast Producer
	79	Viewing Podcast Producer Logs
	80	Monitoring Movie Transfers
	80	Monitoring Xgrid Job Progress
	80	Using Xgrid Admin
	81	Using the Command Line
Glossary	83	
Index	91	

About This Guide

Use this guide to set up and manage Podcast Producer solutions.

Podcast Producer Administration describes how to set up and manage Podcast Producer solutions for publish podcasts of lectures, training, or other audio and video projects.

What's in This Guide

This guide includes the following chapters:

- Chapter 1, “Overview of Podcast Producer,” introduces Podcast Producer and describes its architecture and security model.
- Chapter 2, “Setting Up Podcast Producer,” describes how to get Podcast Producer up and running.
- Chapter 3, “Managing Workflows,” provides a high-level overview of workflows and how to manage them using Server Admin.
- Chapter 4, “Managing Cameras,” describes how to manage and monitor camera usage using Server Admin.
- Chapter 5, “Customizing Workflows,” describes how to customize workflows.
- Chapter 6, “The Podcast Producer Shared File System,” describes the structure of the Podcast Producer shared file system.
- Chapter 7, “Deploying Scalable Podcast Producer Solutions,” describes how to plan the deployment of Podcast Producer solutions.
- Chapter 8, “Podcast Producer Command-Line Tools,” describes the Podcast Producer command-line tools.
- Chapter 9, “Monitoring Podcast Producer,” describes how to monitor and troubleshoot Podcast Producer issues.

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 the Help Viewer application while you're managing Leopard Server. You can view help on a server or an administrator computer. (An administrator computer is a Mac OS X computer with Leopard Server administration software installed on it.)

To get help for an advanced configuration of Leopard 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 *Server Administration* and other advanced administration guides described in "Mac OS X Server Administration Guides," next.

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.

Mac OS X Server Administration Guides

Getting Started covers basic installation and initial setup methods for an advanced configuration of Leopard Server as well as for a standard or workgroup configuration. An advanced guide, *Server Administration*, covers advanced planning, installation, setup, and more. A suite of additional guides, listed below, covers advanced planning, setup, and management of individual services. You can get these guides in PDF format from the Mac OS X Server documentation website:

www.apple.com/server/documentation

This guide ...	tells you how to:
<i>Getting Started and Mac OS X Server Worksheet</i>	Install Mac OS X Server and set it up for the first time.
<i>Command-Line Administration</i>	Install, set up, and manage Mac OS X Server using UNIX command-line tools and configuration files.
<i>File Services Administration</i>	Share selected server volumes or folders among server clients using the AFP, NFS, FTP, and SMB protocols.
<i>iCal Service Administration</i>	Set up and manage iCal shared calendar service.
<i>iChat Service Administration</i>	Set up and manage iChat instant messaging service.

This guide ...	tells you how to:
<i>Mac OS X Security Configuration</i>	Make Mac OS X computers (clients) more secure, as required by enterprise and government customers.
<i>Mac OS X Server Security Configuration</i>	Make Mac OS X Server and the computer it's installed on more secure, as required by enterprise and government customers.
<i>Mail Service Administration</i>	Set up and manage IMAP, POP, and SMTP mail services on the server.
<i>Network Services Administration</i>	Set up, configure, and administer DHCP, DNS, VPN, NTP, IP firewall, NAT, and RADIUS services on the server.
<i>Open Directory Administration</i>	Set up and manage directory and authentication services, and configure clients to access directory services.
<i>Podcast Producer Administration</i>	Set up and manage Podcast Producer service to record, process, and distribute podcasts.
<i>Print Service Administration</i>	Host shared printers and manage their associated queues and print jobs.
<i>QuickTime Streaming and Broadcasting Administration</i>	Capture and encode QuickTime content. Set up and manage QuickTime streaming service to deliver media streams live or on demand.
<i>Server Administration</i>	Perform advanced installation and setup of server software, and manage options that apply to multiple services or to the server as a whole.
<i>System Imaging and Software Update Administration</i>	Use NetBoot, NetInstall, and Software Update to automate the management of operating system and other software used by client computers.
<i>Upgrading and Migrating</i>	Use data and service settings from an earlier version of Mac OS X Server or Windows NT.
<i>User Management</i>	Create and manage user accounts, groups, and computers. Set up managed preferences for Mac OS X clients.
<i>Web Technologies Administration</i>	Set up and manage web technologies, including web, blog, webmail, wiki, MySQL, PHP, Ruby on Rails, and WebDAV.
<i>Xgrid Administration and High Performance Computing</i>	Set up and manage computational clusters of Xserve systems and Mac computers.
<i>Mac OS X Server Glossary</i>	Learn about terms used for server and storage products.

Viewing PDF Guides on Screen

While reading the PDF version of a guide onscreen:

- 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 documentation website:
www.apple.com/server/documentation

Getting Additional Information

For more information, consult these resources:

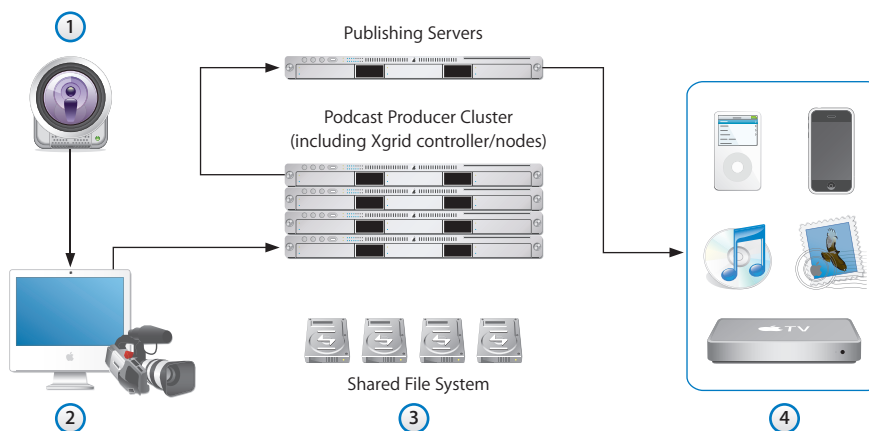
- *Read Me documents*—important updates and special information. Look for them on the server discs.
- *Mac OS X Server website* (www.apple.com/server/macosx)—gateway to extensive product and technology information.
- *Mac OS X Server Support website* (www.apple.com/support/macosxserver)—access to hundreds of articles from Apple’s support organization.
- *Apple Training website* (www.apple.com/training)—instructor-led and self-paced courses for honing your server administration skills.
- *Apple Discussions website* (discussions.apple.com)—a way to 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.

This chapter introduces Podcast Producer and describes the architecture of the Podcast Producer system.

Podcast Producer is a video capture, processing, and publishing system. It is an elegant solution that automates the process of creating and publishing podcasts of lectures, training, or other audio and video projects.

How Podcast Producer Works

Podcast Producer does to the production of podcasts what the assembly line did to automobile production. It automates and streamlines the production of podcasts, as illustrated in the following figure.



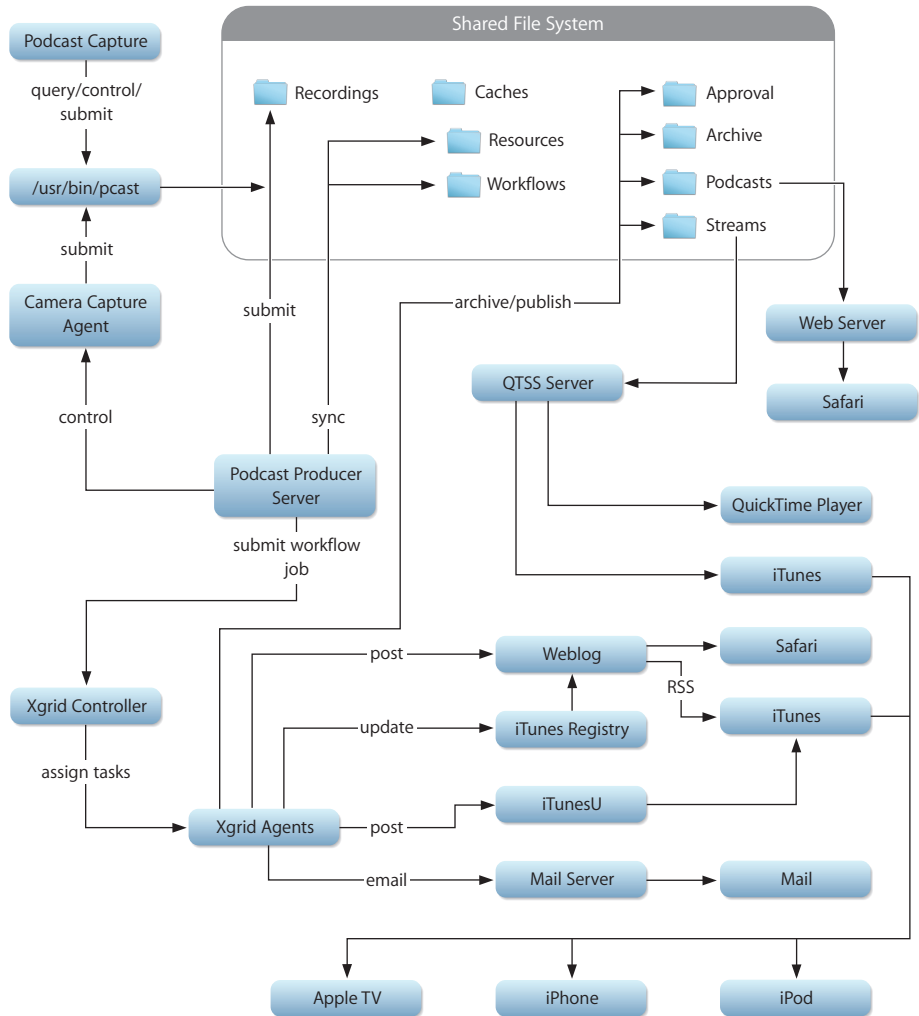
Here's how the process works:

- 1** The user uses Podcast Capture to remotely record audio or video.
The user can also record audio and video locally, record screen activity, or directly submit a QuickTime movie to the Podcast Producer server.
- 2** If remotely recording audio or video, the recording system submits the recording to the Podcast Producer server.

- 3 The Podcast Producer server takes the submission and sends it to an Xgrid cluster to be processed according to the workflow selected in Podcast Capture. When finished, the Xgrid cluster publishes the resulting podcasts. The Xgrid cluster also sends mail notifications to users with instructions on how to access the podcasts.
- 4 Users use iTunes and Safari to access the podcasts and download them to their iPhone, iPod, or Apple TV devices.

The Architecture of the Podcast Producer System

The following figures illustrates the architecture of the Podcast Producer system.



The Podcast Producer system consists of the following main components:

- “Podcast Producer Server” on page 15
- “Podcast Capture” on page 15
- “Podcast Producer Agent” on page 16
- “Shared File System” on page 16
- “Workflows” on page 16
- “Xgrid” on page 17

Podcast Producer Server

The Podcast Producer server is the central point for the administration of a Podcast Producer solution.

The Podcast Producer server manages camera capture agents, provides access control and centralized management, and accepts QuickTime movies to be processed on an Xgrid cluster.

You can use the Podcast Producer server to:

- Specify a shared folder where QuickTime movies are submitted for processing and where podcasts are stored
- Specify the Xgrid controller to be used for processing QuickTime movies
- Control and monitor access to cameras
- Control and monitor access to workflows
- Customize workflow properties

Podcast Capture

Podcast Capture (/Application/Utilities) is the application you use to record audio, video, or screen activity. You can also use Podcast Capture to upload QuickTime movies for processing by the Podcast Producer server.

Podcast Capture is available on Macs with Leopard and Leopard Server.

For more information about how to use Podcast Capture, see its onscreen help.

The podcast Command-Line Tool

The `/usr/bin/podcast` command-line tool provides client-side functionality for recording and submitting QuickTime files. Podcast Capture is a graphical user interface (GUI) that wraps `podcast`.

For more information about `podcast`, see “The podcast Tool” on page 77.

Podcast Producer Agent

The Podcast Producer agent is a daemon controlled by the Podcast Producer server. The agent is responsible for the recording of audio and video from an attached camera.

After the recording is done, the Podcast Producer agent uses the `podcast` tool to upload the resulting QuickTime movie to the Podcast Producer server for encoding and publishing as a podcast.

Shared File System

The Podcast Producer server requires a shared file system for:

- Uploading QuickTime movies.
The Podcast Producer agents use the shared file system to upload QuickTime movies for processing by the Podcast Producer server.
- Keeping a cached copy of workflows.
The Xgrid controller uses the workflows stored in the shared file system to process QuickTime movies.
- Storing the generated podcasts.
The Xgrid agents use the shared file system to store the generated podcasts.

The supported shared file systems are Xsan and NFS.

Workflows

Workflows are the heart of the Podcast Producer system. A workflow is a template that defines a set of customizable Xgrid tasks for encoding and publishing podcasts.

When submitting a QuickTime movie to the Podcast Producer server for processing, you also specify the workflow to use. Podcast Producer makes the necessary replacements in the workflow based on the configuration information entered in Server Admin and sends the workflow as an Xgrid job to the Xgrid controller for processing.

The Podcast Producer server provides a set of sample workflows that define common encoding and publishing tasks for encoding and publishing QuickTime movies as podcasts. However, you can modify these workflows to suit your needs, or you can create your own workflows.

Each sample workflow defines a set of default properties that you can configure using Server Admin, as described later in this guide. You can also define new properties in your custom workflows and use the Podcast Producer server to configure their values.

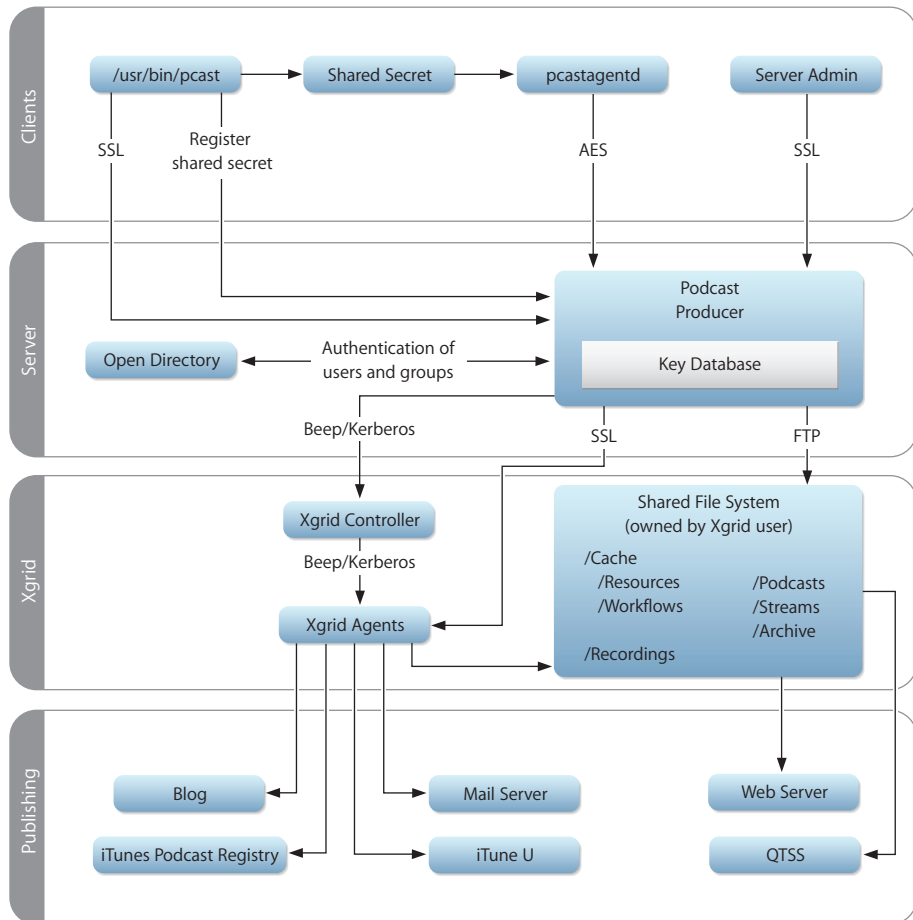
Xgrid

The Podcast Producer server lets you specify an Xgrid controller for processing QuickTime movies on one or more Xgrid agents with Leopard.

Using Xgrid agents for processing QuickTime movies allows you to increase the throughput of the Podcast Producer solution by increasing the number of Xgrid agents. In addition, using Xgrid allows you to scale your system by increasing the number of Xgrid agents.

The Podcast Producer Security Model

Podcast Producer is a secure end-to-end solution, as shown in the following illustration.



Client/Server Communication

To protect sensitive information, the `podcast` command-line tool and Server Admin use the Secure Sockets Layer (SSL) protocol to communicate with the Podcast Producer server. By default, these SSL connections use the server's default self-signed certificate, which comes with the server.

For example, when you specify the passwords for Podcast Producer properties in Server Admin, Server Admin securely passes the passwords to the Podcast Producer server using SSL. The Podcast Producer server encrypts these passwords and stores them in its database.

In addition, the Podcast Producer agent (`pcastagentd`) uses an Advanced Encryption Standard (AES)-secured tunnel to communicate with the Podcast Producer server. This tunnel allows the Podcast Producer server to control the Podcast Producer agent at all times.

When you bind a camera system to Podcast Producer, the `podcast` command-line tool creates a shared secret and sends it to the server using SSL. The server encrypts the shared secret, stores it in its database, and sends back information including the shared secret to the `podcast` tool.

The `podcast` tool stores the information it receives from the Podcast Producer server in a property list. When started, the Podcast Producer agent uses the shared secret stored in its property list to establish an AES-secured tunnel with the Podcast Producer server.

Authentication

The Podcast Producer server uses Open Directory for user and group authentication.

The Podcast Producer server also uses Open Directory to authenticate users and groups specified in Access Control Lists (ACLs) defined in Server Admin. The Podcast Producer server stores these ACLs in the same database it uses to store shared secrets and other sensitive information.

Xgrid

The Podcast Producer server uses the Blocks Extensible Exchange Protocol (BEEP) and Kerberos to communicate with the Xgrid controller using the `xgrid` command-line tool.

The Podcast Producer server uses BEEP and Kerberos to authenticate itself to the Kerberos Key Distribution Center (KDC). The server uses the standard method to get Kerberos tickets from the KDC using the Xgrid name and password supplied in Server Admin.

The Podcast Producer server uses the ticket it gets from the KDC when sending jobs to the Xgrid controller.

As for the Xgrid agent, it authenticates to the Xgrid controller using BEEP and Kerberos. However, when communicating with the Podcast Producer server, the Xgrid agent uses SSL. The Xgrid agent calls the Podcast Producer server to obtain property values and responses to challenges.

For example, if an Xgrid agent tries to post to a blog, the agent gets back a 401 HTTP error with a challenge. The agent sends the challenge to the Podcast Producer server and receives a response, which it passes to the blog to be granted access.

The Podcast Producer server and Xgrid controller must belong to the same Kerberos realm.

Publishing

In the Podcast Producer security model, Xgrid agents have the proper credentials to publish podcasts. If an agent is asked to respond to a challenge, as in the case of a blog, the Xgrid agent can securely obtain the response from the Podcast Producer server and provide the appropriate response.

Podcast Producer Customization

Podcast Producer is an open system that you can customize to meet your needs:

- Workflows are Xgrid jobs, which you can compose using any text editor.
- Podcast Producer leverages Mac OS X technologies like QuickTime and Quartz. QuickTime provides a rich set of video/audio codecs and is the premier platform for media data sets. Anything you can do with QuickTime can be done using Podcast Producer. You can create command-line tools that access QuickTime APIs and use these tools in workflows. Quartz is a powerful image composition framework for image manipulation. You can create tools for adding movie effects and use these tools in Podcast Producer workflows. An example of such a tool is `qc2movie`, which ships with Mac OS X v10.5. This tool is used by some default Podcast Producer workflows and is described in Chapter 5, “Customizing Workflows.”
- All hosting technology used by Podcast Producer is based on Internet standards like HTTP, RSS, Podcast, QTSS, and FTP. You can publish content to any hosting server that supports these standards.
- You can turn virtually any UNIX shell script into an Xgrid job and, therefore, part of a Podcast Producer workflow.

Supported Audio and Video Formats

Podcast Producer supports the following video formats:

- MPEG-4
- H.263
- H.264
- DV NTSC and PAL

In addition, Podcast Producer supports the following audio formats:

- AAC (MPEG-4 Audio)
- AMR Narrowband
- QUALCOMM PureVoice (QCELP)

This chapter describes how to set up the Podcast Producer server.

The instructions in this chapter assume that you have installed and configured Mac OS X Server v10.5. For instructions on setting up Mac OS X Server, see *Getting Started* (included on the Mac OS X Server v10.5 installation disc and downloadable at www.apple.com/server/documentation/).

Hardware and Software Requirements for Podcast Producer

This section describes the hardware and software requirements for providing Podcast Producer services.

For more information about estimating hardware and software requirements, see “Deploying Scalable Podcast Producer Solutions” on page 65.

For more information about minimum system requirements and other Podcast Producer topics not covered in this guide, visit the *Mac OS X Server Support website* at www.apple.com/support/macosxserver.

Podcast Producer Agent Computer Requirements

Following is a list of the minimum hardware and software requirements for the Mac used to capture video:

- Mac OS X v10.5
- 20 GB of free disk space
- Network connectivity (100 Mbps)

Podcast Server Computer Requirements

Following is a list of the minimum requirements for the Mac running the Podcast Producer service:

- A Mac with Mac OS X Server v10.5
- Network connectivity (100 Mbps)

Xgrid Size and Bandwidth Considerations

Following is a list of issues to consider when deciding the size of the Xgrid and the amount of bandwidth needed to process QuickTime movies:

- Number of recording systems and manual submissions
- Type of workflows to be used

The type of workflow used determines the time needed to complete the processing and publishing of podcasts. Also, the type of workflows used determines how many tasks can be performed in parallel.
- A typical recording day schedule for each recording system and manual submission
- Number of uploading Macs
- Number of compute nodes in the grid

Note: All Xgrid agent computers must have a graphics card that supports Quartz Extreme.

Before You Set Up Podcast Producer

The instructions in this section assume that you have chosen the Advanced Configuration option to set up your server. If not, convert your server to an advanced configuration. For information about converting your server to an advanced configuration, see *Server Administration*.

Starting Required Services

Before you configure the services required to set up Podcast Producer, start them. If DNS, Mail, Open Directory, QuickTime Streaming, Web, and Xgrid services are already available on your network, you do not need to start them.

To start required services:

- 1 Launch Server Admin and connect to the server you want to configure.

A sheet appears notifying you that no services are configured on the server.
- 2 Click Choose Configured Services.
- 3 Select the following services:
 - DNS
 - Mail
 - Open Directory
 - Podcast Producer
 - QuickTime Streaming
 - Web
 - Xgrid
- 4 Click Save.

Configuring the DNS Service

Use the instructions in this section to configure the DNS service.

To configure the DNS service:

- 1 Launch Server Admin and connect to the server you are configuring.
- 2 From the list of computers and services on the left, select DNS.
- 3 Click Zones.
- 4 Click Add Zone > Add Primary Zone (Master).
- 5 Click the new zone (its name is example.com.).
- 6 In the Primary Zone field, replace example.com with pcast.private.
- 7 In the Nameservers field, double-click the ns entry and replace ns with server.
- 8 In the Mail Exchangers field, click the Add button (+) and complete the following:
 - a In the Hostname field, enter server.
 - b In the Priority field, enter 10.
- 9 Click Add Record > Add Machine (A).
- 10 Expand the pcast.private.zone by clicking the triangle to its left.
 - a In the Machine Name field of the first machine entry, enter server.
 - b Set the server's IP address to 10.0.0.1.
 - c In the Machine Name field of the first machine entry, enter client.
 - d Set the client's IP address to 10.0.0.10.
- 11 Click Save.
- 12 Start the service by clicking Start DNS.
- 13 To make sure that DNS is running, ping server.pcast.private and client.pcast.private using Network Utility or from the command line.

Configure Mail

The Podcast Producer service requires Mail service to send notification mails.

To configure Mail service:

- 1 In Server Admin, select Mail from the list of computers and services.
- 2 Go to the General tab in the Settings pane.
- 3 Enable SMTP.
- 4 Click Save.
- 5 Start Mail service.

Configuring the Open Directory Service

Follow the steps in this section to configure the Open Directory service.

To configure the Open Directory service:

- 1 In Server Admin, select Open Directory from the list of computers and services.
- 2 Click Settings.
- 3 Click General.
- 4 Click Change.
- 5 Click Open Directory Master, then click Continue.
- 6 Enter an administrator username and password for the Open Directory master domain (record these for later), then click Continue.
- 7 Verify that you have the correct Kerberos realm and search base and click Continue.
- 8 Review settings, then click Continue.
- 9 When a message appears notifying you that service configuration is complete, click Close.
- 10 Click Overview and verify that Kerberos is running.

Creating Required Users

Use Workgroup Manager to create the user accounts required by Podcast Producer.

To create the required users:

- 1 Launch Workgroup Manager (in /Application/Server) and log in to the system hosting the Open Directory service.
- 2 Confirm that the Viewing directory is /LDAPv3/1270.0.1.
- 3 Click the lock icon (upper-right corner).
- 4 Enter the administrator username and password for the Open Directory master domain.
- 5 Click Authenticate.
- 6 Create the following users and assign each a password (to create a user, click New User in the toolbar):
 - pcastadmin: Allow this user to administer this server. Also assign this user a mail address by clicking Info, clicking the (+) button for the email field, and entering the mail address (pcastadmin@server.pcast.private). If Web service is running on a different server, make sure this user is also an administrator on the Web server.
 - pcastxgrid: Set the home folder for this user to /var/empty.
 - pcastuser: Enable mail for this user (by clicking Mail and then clicking Enabled). Also, give this user a mail address (pcastuser@server.pcast.private). To assign the user a mail address, click Info, click the (+) button for the email field, and enter the mail address.

Configuring Web Service

Follow the steps in this section to configure Web service.

To configure Web service:

- 1 In Server Admin, select Web from the list of computers and services.
- 2 Click Sites and select the default site (empty domain name).
- 3 Click Web Services and select Wiki and blog.
- 4 Click Save.
- 5 Start the Web service by clicking Start Web.

Creating a Group Blog

Some workflows require a group blog to post podcasts to.

To create a group blog:

- 1 Open Directory (in /Applications/Utilities).
- 2 Choose Directory > Preferences and make sure Authentication is set to Password; then close the Preferences window.
- 3 Click (+) (bottom left) and choose New Group.
- 4 Enter the administrator username and password for the Open Directory master domain, then click OK.
- 5 Replace Untitled Group with Podcasts.
- 6 Click (+).
- 7 From the People and Groups window, drag the pcastadmin and pcastuser users to the member list of the Podcasts group.
- 8 Close the People and Groups window.
- 9 Click Save.
- 10 Click Ownership and set the Owner and Contact fields to pcastadmin.
- 11 Click Services.
- 12 Select Wiki and Blog (Calendar and Mailing list web archive options are not required).
- 13 Click Save and quit Directory.
- 14 Verify that you can access the following URL when logging in as pcastuser:
<http://server.pcast.private/groups/podcasts>

Important: To ensure that subscribe-to-podcast links in notification emails work, enable podcasting for the group blog. Go to <http://server.pcast.private/groups/podcasts> and log in as an administrator user. Click Settings in the Admin menu. Click the edit (pencil) button (upper right). Select “Create a podcast for this group’s weblog,” then give the podcast an iTunes category. Click Save.

Configuring Xgrid Service

Follow the steps in this section to configure the Xgrid service.

Important: The Xgrid service must be fully Kerberized.

To configure the Xgrid service:

- 1 In Server Admin, select Xgrid from the list of computers and services.
- 2 Click Overview.
- 3 Click Configure Xgrid Service (bottom right), then click Continue.
- 4 Select Host a Grid and click Continue.
- 5 Provide the administrator username and password for the Open Directory master domain and click Continue.
- 6 Confirm the configuration settings and click Continue.
- 7 Click Close.

The Service Configuration Assistant will quit.

Important: Use the Service Configuration Assistant as indicated above to configure Xgrid to use Kerberos for all types of authentication. To avoid making configuration mistakes, do not try to manually configure Xgrid for Kerberos authentication.

- 8 Verify that Xgrid service is running and that Xgrid has configured a Controller and Agent using Kerberos.

Configuring QuickTime Streaming Service

This section describes how to configure the QuickTime Streaming service.

To configure the QuickTime Streaming service:

- 1 In Server Admin, select QuickTime Streaming from the list of computers and services.
- 2 Click Start QuickTime Streaming.

Setting Up Podcast Producer

To start providing Podcast Producer services, you must do the following:

- Configure the Podcast Producer general settings.
- Configure the Podcast Producer default workflow properties.
- Start the Podcast Producer service.

Configuring General Settings

You can use Server Admin to configure the Podcast Producer general settings.

- “Specifying a Shared File System”
- “Specifying an Xgrid Controller”

Specifying a Shared File System

Podcast Producer lets you specify a shared folder that Podcast Capture and Xgrid agents can use for uploading QuickTime movies and podcasts. The shared folder can reside on the server where Podcast Producer is running or it can reside on another server.

To specify a shared folder:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click General.
- 5 In the Shared File System field, click Choose.
- 6 Navigate to the folder and click Choose.
The path to the shared folder appears in the Shared File System field.
- 7 Click Save.

Specifying an Xgrid Controller

Podcast Producer uses the Xgrid controller specified in the General pane of the Podcast Producer service to send Podcast Producer workflow jobs.

To specify an Xgrid controller:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click General.
- 5 In the Xgrid Controller field, enter the hostname of the Xgrid controller.
- 6 In the Xgrid User Name field, click Choose User.
- 7 In the Users and Groups window, click Users.
- 8 From the Users and Groups window, drag a user into the Xgrid User Name field.
- 9 In the Xgrid User Password field, enter the Xgrid user's password.
- 10 Click Save.

Important: Xgrid agents must be bound to the same Open Directory server used by Podcast Producer. Xgrid agents must also be Kerberized in the same realm used by Podcast Producer. In addition, Xgrid agents must have access to the Podcast Producer shared file system.

Configuring the Default Workflow Settings

You can use Server Admin to configure a set of default workflow properties.

To configure default workflow properties:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click Properties.
- 5 Click the triangle next to Default Properties to display the properties.
- 6 Set properties as follows:

Property	Value
Administrator Short Name	pcastadmin
Group Short Name	podcasts
Group Administrator User	pcastadmin
Group Administrator Password	pcastadmin's password
Approval Email List	List of mail recipients of podcast approval requests (for example, editor@example.com, producer@example.com)
Audience Email List	List of mail recipients of new podcast announcements (for example, joe@example.com, jane@example.com, freshmen@example.com)
Audience SMS List	List of SMS recipients of new podcast announcements (for example, 5552121234@sms_text_host.com, 4442121235@sms_text_host.com)

To post to iTunes U, enter the appropriate iTunes U values.

For more information about configuring default properties, see “Configuring Default Workflow Properties” on page 33.

Starting the Podcast Producer Server

You use Server Admin to start the Podcast Producer server.

To start the Podcast Producer server:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Start Podcast Producer.

To stop the Podcast Producer server, click Stop Podcast Producer.

Binding a Mac to the Podcast Producer Service

After setting up the Podcast Producer service, you can bind the Macs you want to use for remotely recording audio and video to the Podcast Producer service using the Podcast Capture application.

To bind a Mac to the Podcast Producer service:

- 1 If you are using an external video camera, make sure the camera is connected to your Mac via FireWire and is turned on.
- 2 Open the Podcast Capture application (in /Applications/Utilities).
- 3 When prompted, enter a valid username and password.
- 4 Choose Podcast Capture > Local Camera Settings.
- 5 Click the lock and authenticate if necessary.
- 6 From the Video Source pop-up menu choose a local camera.
- 7 Verify that you see a preview of the camera's video.
- 8 From the Microphone pop-up menu choose the audio source.
- 9 In the Camera Name field enter the name of the camera.

Because Podcast Producer uses the name you enter in this field to identify and control access to the camera, you should enter a unique, meaningful name to help the Podcast Producer administrator and Podcast Capture users better identify and determine the location of the camera. For example, instead of using a name like Camera7, use a name like Camera007-Eng. Building/Room 304.

- 10 In the Server field enter the fully qualified name of the Podcast Producer server (for example, server.pcast.private) or the IP address of the server.
- 11 Click Bind to bind the camera to the Podcast Producer server.
- 12 Enter your Podcast Producer administrative credentials and click OK.

Verifying Your Setup

You can verify your setup by uploading a QuickTime movie for processing by your Podcast Producer server.

To verify your setup:

- 1 On a Mac running Mac OS X v10.5 and connected to the same network that your Podcast Producer server is connected to, launch Podcast Capture (in /Applications/Utilities).
- 2 Log in to the Podcast Producer server as pcastuser.
- 3 Select File and choose the QuickTime movie you want to submit to the Podcast Producer server for encoding and publishing.
- 4 From the Workflow pop-up menu, select a workflow and give the movie a title and description.
- 5 Click Publish.
- 6 To monitor the progress of your submission, use Xgrid Admin.

For more information about Xgrid Admin, see the *Xgrid Administration* guide.

You should receive a mail notification after your video is ready to be viewed.

This chapter describes how to manage workflows and configure their default and custom properties in Server Admin.

Workflows are an essential part of Podcast Producer. Learning how to manage workflows and how to configure their properties allows you to streamline the podcast generation process and make the best use of your computing resources.

Controlling Access to Workflows in Podcast Capture

Podcast Producer server lets you specify which workflows users can see (and which are allowed to be used) in Podcast Capture and the `podcast` command-line tool.

To control access to a workflow:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Workflows.
- 4 Select a workflow in the Workflow list.
- 5 To restrict access to the workflow, click “Allow access to *workflow name* for the following users and groups.”
- 6 Click the (+) button to add users and groups to the list of users and groups that can access the selected workflow.

In the Users and Groups window, click Users and drag one or more users to the list.

In the Users and Groups window, click Groups and drag one or more groups to the list.

To delete users and groups from the list, select them and click (-).

- 7 Click Save.

You can also select multiple workflows and configure their access control settings at the same time.

To allow all users and groups to see the selected workflows in Podcast Capture, click “Allow access to *workflow name* for all users and groups.”

Monitoring Workflow Usage

You can use the Workflows pane of the Podcast Producer service to see the last time a workflow was used and by whom.

To monitor workflow usage:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Workflows.

The Last Used column shows the last time a workflow was used. The Last Used By column shows the short name of the user who last used the workflow.

Filtering Workflows

You can use the search field in the Workflows pane of the Podcast Producer service to specify a search criteria for listing workflows.

To filter workflows:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Workflows.
- 4 To search for workflows by keywords, from the Search pop-up menu, choose Workflow and enter the keywords.
Only workflows whose names contain the specified keywords appear in the list.
- 5 To search for workflows by filename, from the Search pop-up menu, choose Workflow File Name and enter the name of the workflow file.
Only the workflow whose file name is specified in the Search field appears in the list.
- 6 To list the workflows that were last used by a particular user, from the Search pop-up menu, choose Last Used By and enter the short name of the user.

Displaying Workflow Information

You can display information about a workflow using its corresponding information button.

To display workflow information:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Workflows.
- 4 Select a workflow in the Workflow list.
- 5 Click the information button.

The Workflow Information window appears. This window displays information about the workflow, including the workflow's filename and location on the Podcast Producer server, a description of what the workflow does, and any notes that were added to the workflow file.

Configuring Workflow Properties

Server Admin allows you to configure default and custom workflow properties.

- “Configuring Default Workflow Properties” on page 33
- “Configuring Custom Workflow Properties” on page 35

Configuring Default Workflow Properties

You can use Server Admin to configure a set of default workflow properties, which are used by all Podcast Producer workflows that ship with Mac OS X Server version 10.5.

The following table describes these properties.

Property Name	Description
Administrator Short Name	The short name of an admin user that Podcast Producer sends automatic mail notifications to.
Approval Email List	Workflow approval request mail is sent to this address.
Approval Folder	Content files requiring workflow approvals are stored in this folder.
Archive Root	The path to the folder in the Podcast Producer shared file system where the Podcast Producer server stores backup copies of uploaded QuickTime movies.
Audience Email List	A list of mail recipients of new podcast announcements from the Podcast Producer.
Audience SMS List	A list of SMS recipients of new podcast announcements from the Podcast Producer.
Copyright	The copyright notice to apply to all podcasts as an annotation.
Exit Video Path	The path to a QuickTime movie in the shared file system to add at the end of podcasts.
Group Short Name	The short name of the group that receives blog announcements about new podcasts from Podcast Producer.
Groups Administrator Password	The password of the administrator group specified by Group Short Name.
Groups Administrator Username	The short name of the administrator user who is the group's owner and can post blogs on behalf of normal users. This user must also be an administrator on the system running the Web service used by Podcast Producer to post blogs.
Groups Web Server URL	The base URL of the web server where group blog announcements about new podcasts are published.

Property Name	Description
Intro Transition Duration	The transition duration (in seconds) between the introduction video and the main content.
Introduction Video Path	The path to a QuickTime movie in the shared file system to add at the beginning of a podcast.
iTunes U Posting Credentials	The credentials of a user that can post content to an institution's iTunesU site.
iTunes U Shared Secret	The communications secret shared between an institution and its iTunes site.
iTunes U Site URL	The URL of the iTunes site to post podcasts on.
iTunes U Tab ID	The ID of the table view of the iTunes site where podcasts are posted.
Notification Language	The language used when sending mail and blog announcements about new podcasts.
Organization	The name of the organization to be added to the introduction video.
Postflight Script Path	The path to a script on the shared file system to run as the last script in a workflow.
Preflight Script Path	The path to a script on the shared file system to run as the first script in a workflow.
QTSS URL	The base URL of the QuickTime Streaming Server used to stream new podcasts.
Quartz Composer Filter	The name of the Quartz Composer filter used by workflows for applying filtering compositions to podcasts.
Quartz Composer Introduction Transition	The name of the Quartz Composer transition used by workflows that apply transition compositions between the introduction video and the main content.
Quartz Composer Exit Transition	The name of the Quartz Composer transition used by workflows that apply transition compositions between the main content and the exit video.
QuickTime Streaming Media Root	The QuickTime Streaming Server media folder on the shared file system where podcast streams are stored.
SMTP Server	The hostname of an SMTP mail server used by workflows to send mail notifications.
Watermark Image	The path to an image on the shared file system used by workflows that apply watermarks to podcasts.
Web Document Root	The web server root folder on the shared file system where new podcast streams are published.
Web URL	The base URL of the web server where podcasts are published.

To configure default workflow properties:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click Properties.
- 5 Click the triangle next to Default Properties to display the properties.
- 6 To change the value of a property, double-click the property's value and enter a new value.
- 7 Click Save.

Configuring Custom Workflow Properties

You can modify the sample Podcast Producer workflows and define customizable properties. You can then add the properties to the list of Custom Properties in Server Admin and change their values as needed.

Adding Custom Workflow Properties

You can use Server Admin to add custom workflow properties.

To add a custom workflow property:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click Properties.
- 5 Click the triangle next to Custom Properties to display the custom properties.
- 6 Click the (+) button to add a custom property.
- 7 Double-click the name field of the property and enter its name.
- 8 Double-click the value field of the property and enter its value.
- 9 Click Save.

Deleting Custom Workflow Properties

You can use Server Admin to delete custom workflow properties.

To delete custom workflow properties:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click Properties.
- 5 Click the triangle next to Custom Properties to display the custom properties.
- 6 Select the property to delete and click the (-) button.
- 7 Click Save.

Modifying Custom Workflow Properties

You can use Server Admin to modify custom workflow properties.

To modify custom workflow properties:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Settings.
- 4 Click Properties.
- 5 Click the triangle next to Custom Properties to display the custom properties.
- 6 To change the value of a property, double-click the property's value and enter a value.
- 7 (Optional) To encrypt a property, click the property's checkbox.
- 8 Click Save.

This chapter describes how to manage and monitor camera usage.

The Podcast Producer server allows you to control and monitor camera usage.

- “Controlling Access to Cameras in Podcast Capture” on page 37
- “Removing Cameras” on page 38
- “Filtering Cameras” on page 38

Controlling Access to Cameras in Podcast Capture

The Podcast Producer server lets you specify which cameras users can choose in Podcast Capture.

To control access to a camera:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Cameras.
- 4 Select a camera in the Cameras list.
- 5 To restrict access to the camera, click “Allow access to *camera name* for the following users and groups.”
- 6 Click the (+) button to add users and groups to the list of users and groups that can access the selected camera.

In the Users and Groups window, click Users and drag users to the list.

In the Users and Groups window, click Groups and drag groups to the list.

To delete users or groups from the list, select them and click (-).

- 7 Click Save.

You can also select multiple cameras and simultaneously configure their settings.

To allow all users and groups to see the selected camera in Podcast Capture, click “Allow access to *camera name* for all users and groups.”

Removing Cameras

The Podcast Producer server lets you remove cameras from the list of cameras in Server Admin.

To remove a camera:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Cameras.
- 4 Select a camera in the Cameras list.
- 5 Click Remove Camera.
- 6 Click OK.

Filtering Cameras

You can use the search field in the Cameras pane of the Podcast Producer service in Server Admin to specify a filtering or search criteria for listing cameras.

To filter cameras:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Cameras.
- 4 To search for cameras by keywords, from the Search pop-up menu, choose Camera and enter the keywords.
Only the cameras whose names contain the specified keywords appear in the list.
- 5 To list cameras that were last used by a specific user, from the Search pop-up menu, choose Last Used By and enter the short name of the user.
- 6 To list cameras based on their status, from the Search pop-up menu, choose Status and enter the name of the status.

This chapter describes workflows and shows you how to customize or create your own.

The workflows that ship with Mac OS X Server v10.5 are examples of the type of back-end processing you can perform on input QuickTime files.

Before you customize workflows, make sure you understand how workflows work.

The Structure of a Workflow Bundle

A workflow is a self-contained bundle that stores all files needed by the workflow.

By default, workflows are stored in the `/System/Library/PodcastProducer/Workflows` folder. However, Podcast Producer also checks for workflows in `/Library/PodcastProducer/Workflows`. If the same workflow is present in both folders, Podcast Producer uses the workflow in the `/Library/PodcastProducer/Workflows` folder.

The reason for giving precedence to workflows in `/Library/PodcastProducer/Workflows` is that workflows in `/System/Library/PodcastProducer/Workflows` can be overwritten without warning during a software update.

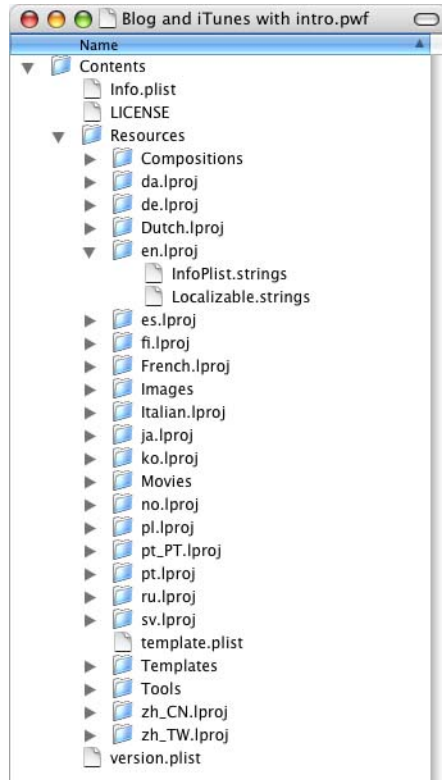
Important: To protect your modifications from being overwritten by the system during a software update, do not store your customized workflows in the `/System/Library/PodcastProducer/Workflows` folder. Instead, copy the contents of the `/System/Library/PodcastProducer/` folder to the `/Library/PodcastProducer/` folder and make your changes there.

Global Encodings and Resources

Podcast Producer uses the following folders to store global encodings and resources available to all workflows:

- /System/Library/PodcastProducer/Encodings
- /System/Library/PodcastProducer/Resources

The following figure shows the contents of a workflow bundle:



The following table describes the contents of a workflow bundle:

Element	Description
Info.plist	Contains information about the workflow bundle, including the name of the workflow as it appears in Server Admin and in the workflow's description. (Name and description strings are localizable.)
LICENSE	Contains the terms for using the workflow.
Resources	Contains all local resources used by the workflow. Global workflow resources are stored in /System/Library/PodcastProducer/Resources.
version.plist	Contains workflow version information.

The following table describes the contents of the Resources folder:

Element	Description
Compositions	Stores the Quartz compositions referenced by the workflow.
.lproj folders	Contain the localizable strings used in the workflow. For example, en.lproj contains the English strings used in the workflow.
InfoPlist.strings	Contains the name and workflow descriptions that appear in Server Admin.
Localizable.strings	Contains the localizable strings used in the workflow.
Images	Contains images used by the workflow.
Movies	Contains movies used by the workflow.
template.plist	The workflow template that specifies the tasks to execute and the order of execution.
Templates	Contains templates used by the workflow. By default, this folder contains a mail template and a template for posting to blogs. If your custom workflows use special templates, store them in this folder.
Tools	Contains command-line tools used by the workflow.

The Structure of a Workflow

A workflow template is a property list (plist) containing tasks that Xgrid agents must execute. A workflow consists of the following first-level elements:

Element	Description
artConditions artSpecifications	These elements allow you to score Xgrid jobs. You can tell the Xgrid controller to run certain jobs on certain Xgrid nodes and to prefer certain Xgrid nodes. For more information about these elements and how to configure them, see the <i>Command-Line Administration</i> guide or the <code>xgrid</code> man page.
name	This entry specifies the name of the Xgrid job. When the Podcast Producer server receives a job submission, it replaces the value of this key (<code>\$\$Xgrid Job Name\$\$</code>) in the corresponding workflow with a name it assigns to the job before sending it to the Xgrid controller.
notificationEmail	This key specifies the mail address the Podcast Server uses to notify the administrator about the status of workflow jobs submitted to the Xgrid controller. Podcast Producer replaces the value of this key (<code>\$\$Administrator Email Address\$\$</code>) with the administrator's mail address before sending the job to the Xgrid controller.
taskSpecifications	This is the most important entry. It specifies the tasks to be executed by Xgrid agents.

Workflow Task Specifications

The taskSpecifications entry consists of one or more task specifications, as shown in the following example. In this example, the taskSpecifications section lists 20 tasks. The name of the first task is annotate, the name of the second task is archive, and so on.



Each task represents a UNIX shell script command and consists of the following elements:

Element	Description
arguments	Lists the arguments required by the command.
command	Specifies the command to run (including the path).
dependsOnTasks	(Optional) Lists the tasks that must be completed before this command runs.

For example, the `encode_wifi` task runs the following command after the `preflight` task is completed:

```
/usr/libexec/podcastproducer/tasks/pcastaction encode
  --basedir=${Base Directory}
  --input=${Content File Basename}-final.mov
  --output=${Content File Basename}-wifi.mp4
  --encoder=h264_hint_server
```

When Podcast Producer receives a job submission, it replaces the `$$` property keys in the workflow template with the corresponding values before sending the job to the Xgrid controller.

Property Keys

Workflow templates use keys (strings enclosed by “`$$`”) to represent default and custom workflow properties.

Podcast Producer defines a set of default properties that are used by the workflows that ship with the product. In addition, Podcast Producer allows you to define custom properties for use in your custom workflows. You can modify the values of properties and create new properties in Server Admin.

When Podcast Producer receives a job submission from Podcast Capture, the Podcast Producer server replaces the property keys in the specified workflow with the appropriate values defined in Server Admin before submitting the Xgrid job for processing.

For example, the Podcast Producer server replaces the `$$Group Short Name$$` key with the value of the Group Short Name default property you defined in Server Admin.

Important: Podcast Producer also uses protected property keys (strings enclosed by “`##`”) in some workflows. For example, Podcast Producer replaces `##Groups Administrator Username:Groups Administrator Password##` with a one-time password for authentication challenges. Protected property keys are intended for use by Podcast Producer and should not be modified.

Default Property Keys

To represent a default property in workflow templates, Podcast Producer encloses the name of the property as it appears in Server Admin by “`$$`” characters. For example, the Archive Root property appears as `$$Archive Root$$` in workflows.

Podcast Producer does not store passwords in the workflow template because the workflow jobs are XML files and are not encrypted.

Instead, the Podcast Producer server encrypts and stores the passwords you enter in Server Admin in a special database for maximum protection, as described in “The Podcast Producer Security Model” on page 17.

Server-Generated Property Keys

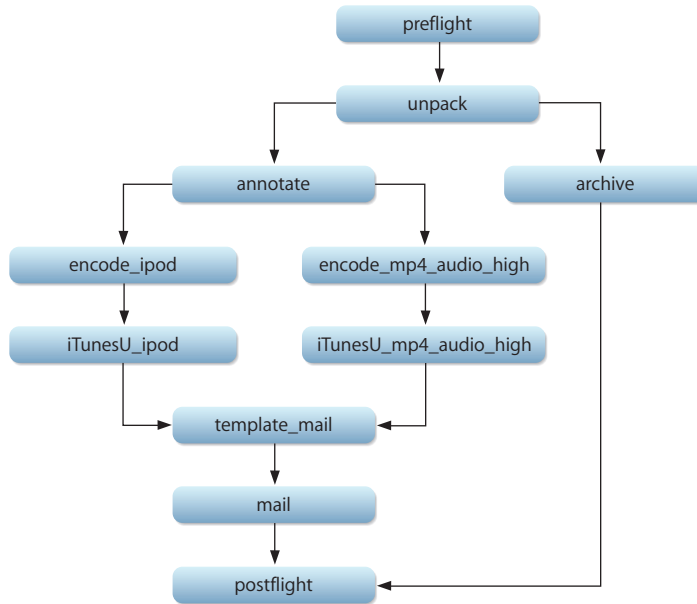
The Podcast Producer server replaces the following property keys with the appropriate values.

Property Key	Matching Property Name
\$\$Administrator Email Address\$\$	The mail address of the Podcast Producer administrator.
\$\$Administrator Full Name\$\$	The full name of the Podcast Producer administrator.
\$\$Base Directory\$\$	The base directory that Podcast Producer creates for storing submitted recordings (for example, <Shared_File_System>/Recordings/<UUID>).
\$\$Content File Basename\$\$	The name of the submitted recording.
\$\$Content File Extension\$\$	The name of the file extension of the submitted recording.
\$\$Content File Name\$\$	The combination of Basename and Extension. For example, if Content File Basename is MyMovie and Content File Extension is .mov, Content File Name is MyMovie.mov.
\$\$Date_YYYY-MM-DD\$\$	The date of the workflow submission.
\$\$Global Resource Path\$\$	The same as \$\$Workflow Resource Path\$\$ except that it refers to <Shared File System>/Caches/Resources.
\$\$Podcast Producer URL\$\$	The URL of the Podcast Producer server.
\$\$Server UUID\$\$	The UUID of the Podcast Producer server.
\$\$Shared Filesystem\$\$	The top level of the shared file system.
\$\$Title\$\$	The title of the workflow job submission specified in Podcast Capture or podcast.
\$\$User Email Address\$\$	The mail address of the user who submitted the job.
\$\$User Full Name\$\$	The full name of the user who submitted the job.
\$\$User Home Directory\$\$	The home directory path of the user submitting the job.
\$\$User ID\$\$	The ID of the user who submitted the job.
\$\$User Short Name\$\$	The short name of the user who submitted the job.
\$\$Workflow Bundle Path\$\$	The path to the top level of the workflow bundle on the shared file system (for example, <Shared File System>/Caches/Workflows/Blog.pwf).
\$\$Workflow Resource Path\$\$	The path to the workflow's Resources folder.
\$\$Xgrid Job Name\$\$	The name that Podcast Producer assigns to the job it submits to the Xgrid controller.

Task Dependencies

A workflow task specification can specify one or more tasks as dependencies. This helps the Xgrid controller determine the order of task execution.

The following illustration shows an example of task dependencies.



In this example, The unpack task won't run until the preflight task completes successfully. Also, the template_mail task won't run until iTunesU_ipod and iTunesU_mp4_audio_high tasks complete successfully.

In addition to specifying the order of task execution, task dependencies help the Xgrid controller determine which tasks can run in parallel. In the above example, the tasks at the same level (for example, the encode_ipod and encode_mp4_audio_high tasks) can be run in parallel if two Xgrid agents are available.

Workflow Commands

Podcast Producer provides a rich set of workflow commands or tasks that you can use in custom workflows. However, you can always write your own commands or use commands from third parties.

The following is a description of the `pcastaction` commands you can use in workflows.


Command	Description
<code>pcastaction annotate</code>	Adds annotations to the input movie.
<code>pcastaction approval</code>	Submits content for approval.
<code>pcastaction archive</code>	Archives the input movie at the specified location.
<code>pcastaction encode</code>	Encodes the input movie using the specified codec.
<code>pcastaction iTunes</code>	Instructs the iTunes store to check the specified RSS feed for new episodes.
<code>pcastaction iTunesU</code>	Posts the input video at the specified iTunes U tab.
<code>pcastaction groupblog</code>	Posts to a group's wiki blog.
<code>pcastaction mail</code>	Sends a notification mail to the specified user using the mail template in the workflow's Resources/Templates folder.
<code>pcastaction merge</code>	Merges two movies with a fade transition between them.
<code>pcastaction preflight</code>	Runs the preflight script (System/Library/PodcastProducer/Resources/Tools/preflight_script) with the specified arguments.
<code>pcastaction postflight</code>	Runs the postflight script (System/Library/PodcastProducer/Resources/Tools/postflight_script) with the specified arguments.
<code>pcastaction publish</code>	Publishes the input file to a web or QTSS server.
<code>pcastaction shell</code>	Runs the specified shell script with the specified arguments.
<code>pcastaction template</code>	Processes a web or mail template into a localized content block to be used in mail or web postings.
<code>pcastaction title</code>	Adds the supplied title to the input video.
<code>pcastaction unpack</code>	Unpacks folder archives prior to running the main part of any workflow.
<code>pcastaction watermark</code>	Superimposes the specified image as a watermark over the input video.

You can also use the `qceffect` command to export the specified Quartz Composer composition to the input movie by adding a track containing this composition.

For more information about the `pcastaction` command, see the *Command-Line Administration* guide.

Podcast Producer Default Workflows

Podcast Producer ships with the following workflows:

- “Blog without archive,” as described on page 48
- “Blog,” as described on page 49
- “Blog and iTunes with watermark,” as described on page 50
- “Blog and iTunes with intro,” as described on page 51
- “Blog and iTunes with intro and effects,” as described on page 52
- “Blog and iTunes with intro and effects for tv,” as described on page 53
- “iTunes U,” as described on page 54
- “iTunes U with intro,” as described on page 55
- “Blog with streaming and no archive,” as described on page 56
- “Blog with streaming,” as described on page 57
- “Blog with streaming and intro,” as described on page 58
- “Blog with streaming, intro and effects,” as described on page 59
- “Submit for Approval,” as described on page 60

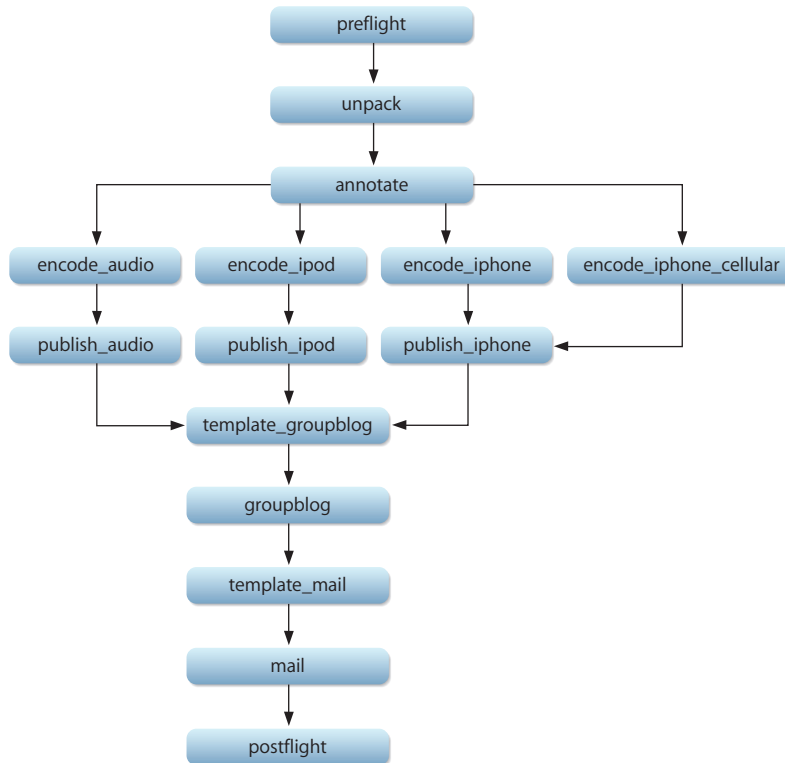
As mentioned earlier in this guide, these workflows are provided as examples. If these workflows meet your needs, you can use them without modifications. However, for some workflows you’ll need to replace the sample resources that come with Podcast Producer (for example, the intro and exit video) with your own resources.

To customize workflows, see “Customizing Workflows” on page 60.

Blog without archive

The “Blog without archive” workflow (/System/Library/PodcastProducer/Workflows/Blog without archive.pwf) encodes the submitted QuickTime movie to video and audio iPod formats, publishes the results to a group’s blog, and sends an announcement mail to users with instructions on how to access the published content.

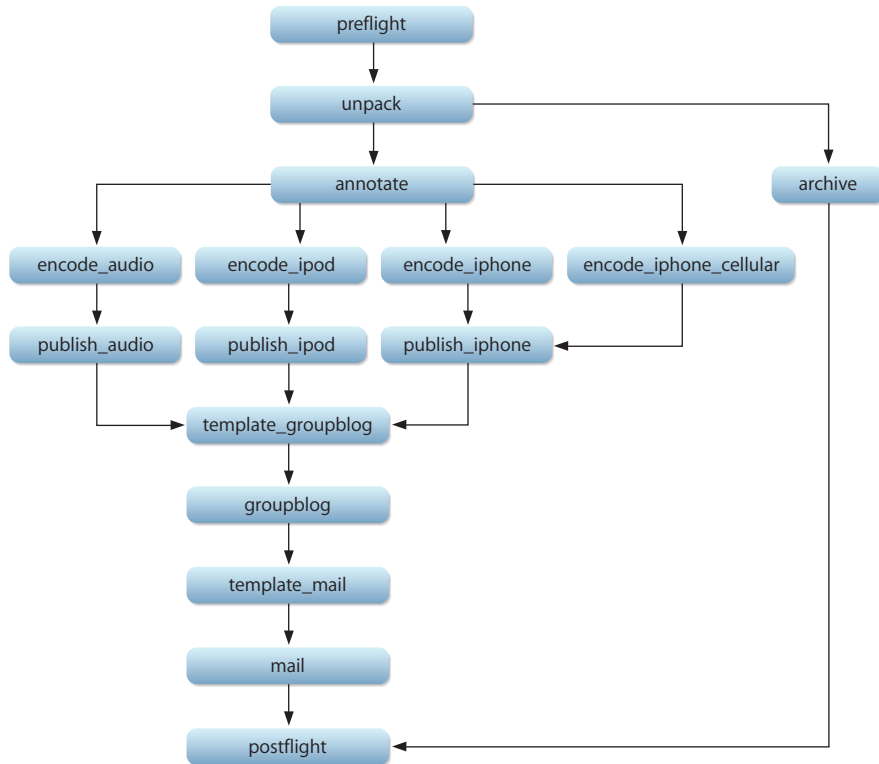
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog

The Blog workflow (/System/Library/PodcastProducer/Workflows/Blog.pwf) archives and annotates the submitted QuickTime movie. Then this workflow encodes the annotated movie to video and audio iPod formats, publishes the results to a group's blog, and sends an announcement mail to users with instructions on how to access the published content.

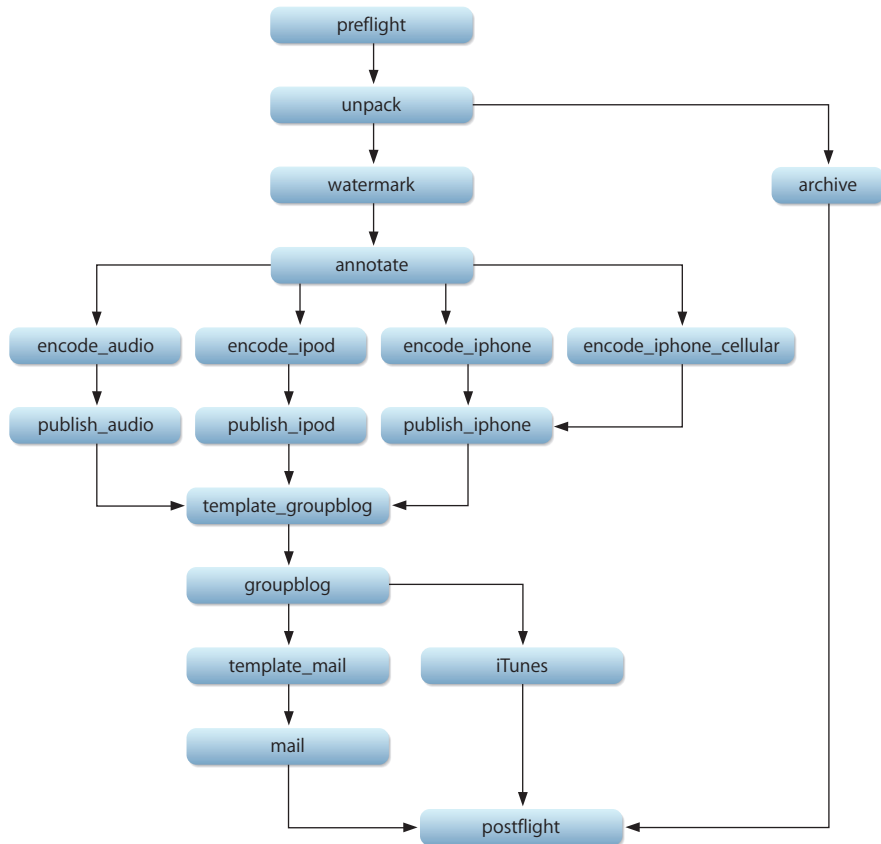
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog and iTunes with watermark

The “Blog and iTunes with watermark” workflow (/System/Library/PodcastProducer/Workflows/Blog and iTunes with watermark.pwf) archives the submitted recording. It also annotates and adds a watermark to the recording. Then, it encodes the recording to video and audio iPod formats. Finally, it publishes the resulting podcasts to the group’s blog, notifies the iTunes podcast directory of the new episode, and sends an announcement mail to users with instructions on how to access the podcasts.

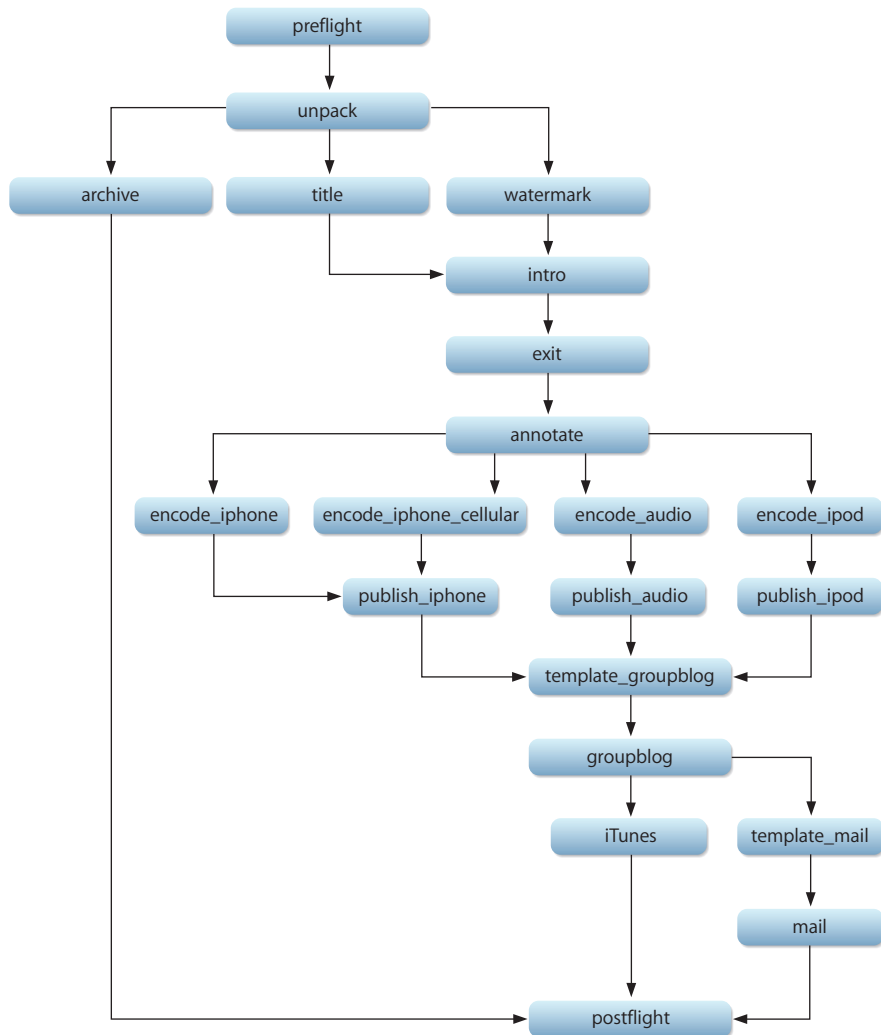
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog and iTunes with intro

The “Blog and iTunes with intro” workflow (/System/Library/PodcastProducer/Workflows/Blog and iTunes with intro.pwf) archives the recording. It also annotates and adds a watermark to the recording. It then adds a title and intro to the recording followed by an exit movie. After that, the workflow encodes the recording to video and audio iPod formats. Finally, it publishes the resulting podcasts to a group’s blog, notifies the iTunes podcast directory of the new episode, and sends an announcement mail to users with instructions on how to access the published podcasts.

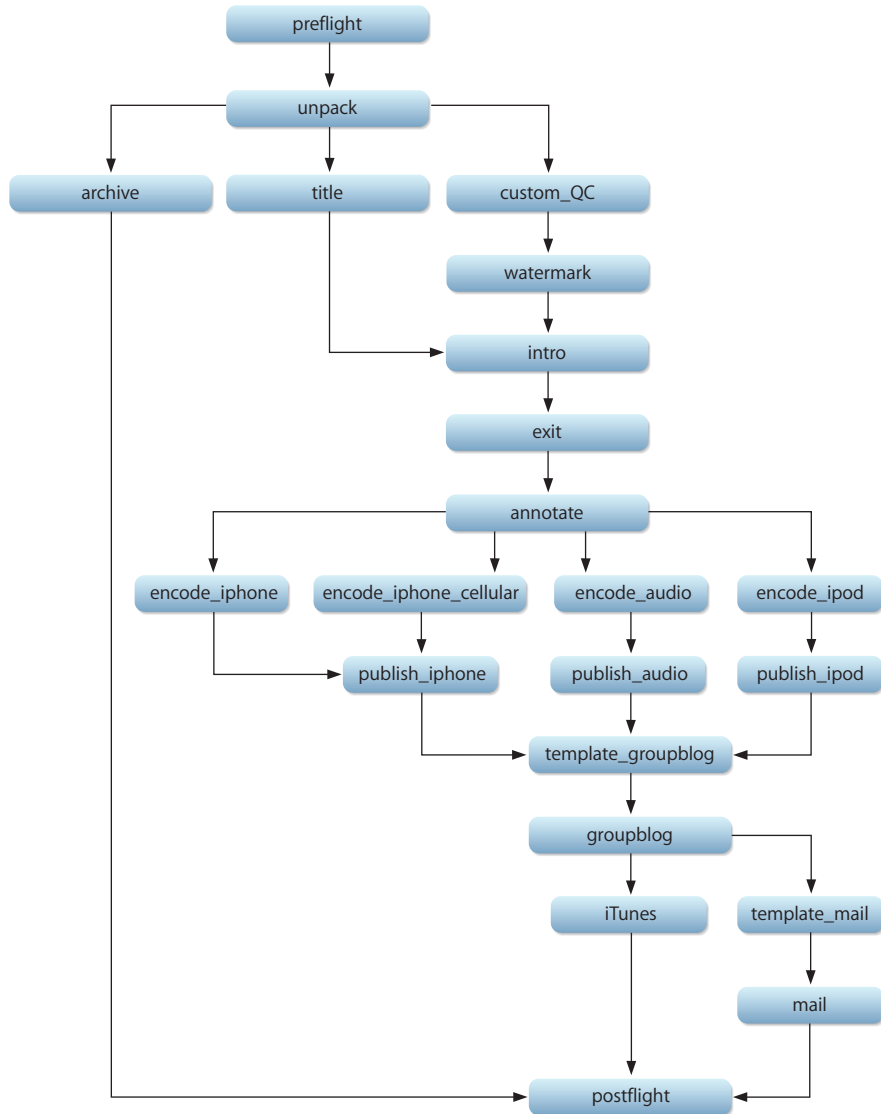
The following illustration shows the order of task execution and task dependencies in the workflow.





Blog and iTunes with intro and effects

The “Blog and iTunes with intro and effects” workflow (/System/Library/PodcastProducer/Workflows/Blog and iTunes with intro and effects.pwf) is the same as the “Blog and iTunes with intro” workflow except that this workflow applies a custom Quartz Composer filter to the recording before adding the watermark.

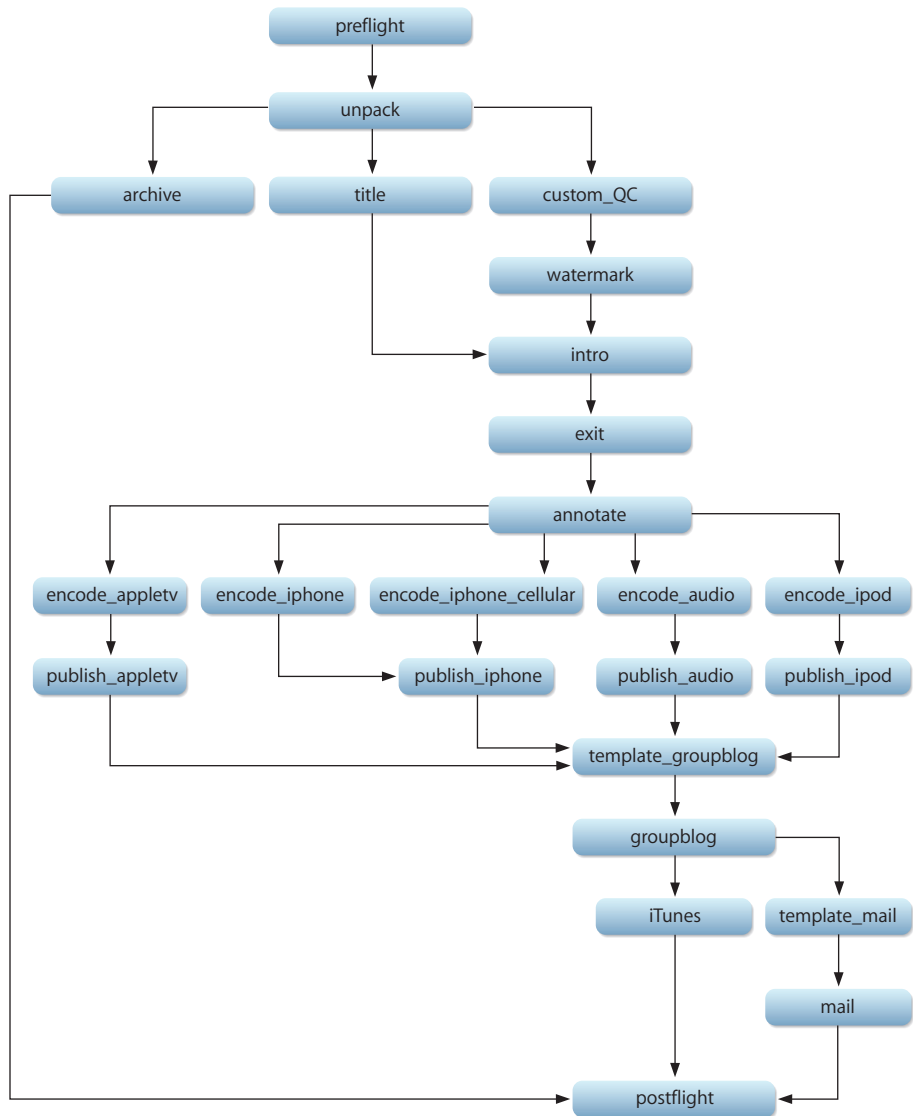
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog and iTunes with intro and effects for tv

The “Blog and iTunes with intro and effects for tv” workflow (/System/Library/PodcastProducer/Workflows/Blog and iTunes with intro and effects for tv.pwf) is the same as the “Blog and iTunes with intro and effects” workflow except that this workflow also encodes the submitted recording to Apple TV format.

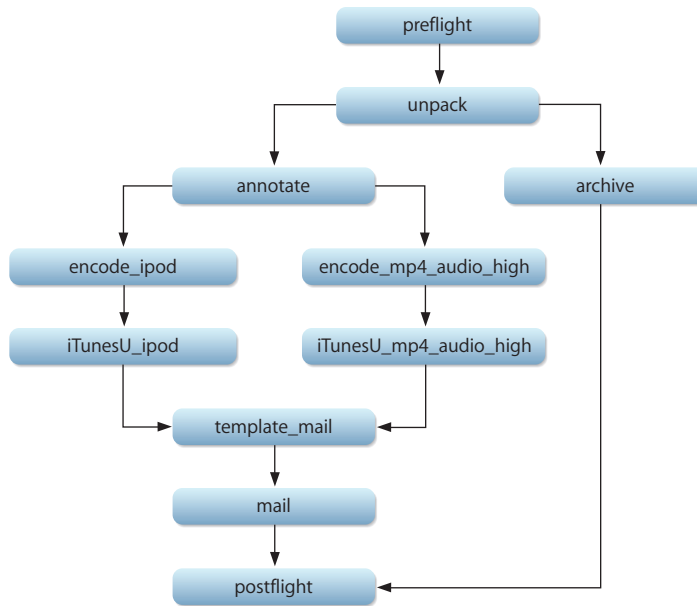
The following illustration shows the order of task execution and task dependencies in the workflow.



iTunes U

The iTunes U workflow (/System/Library/PodcastProducer/Workflows/iTunes U.pwf) archives the recording. It also annotates the recording before encoding it to video and audio iPod formats, publishes the resulting podcasts to iTunes U, and sends an announcement mail to users with instructions on how to access the published podcasts.

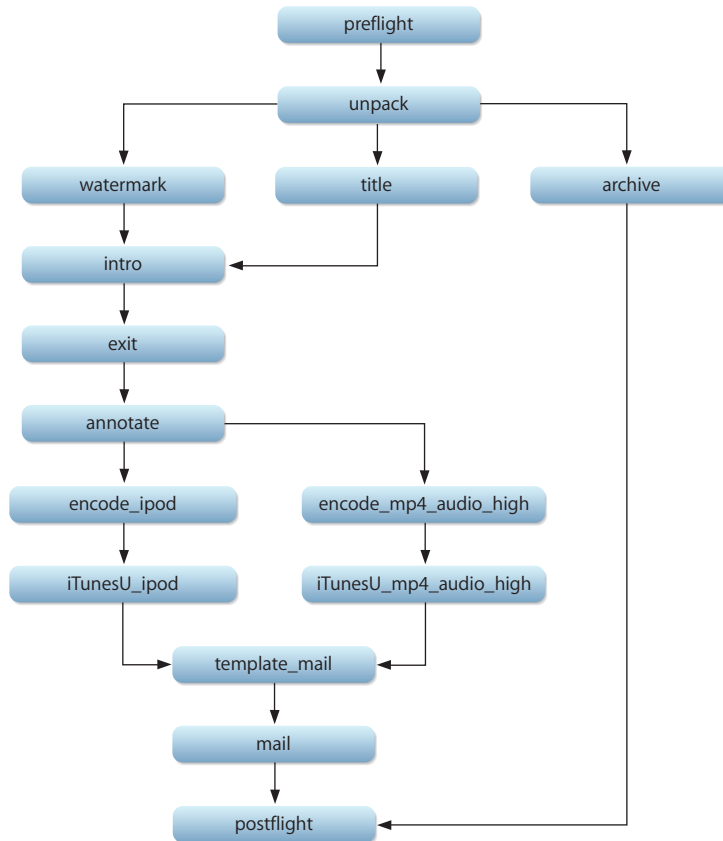
The following illustration shows the order of task execution and task dependencies in the workflow.



iTunes U with intro

The “iTunes U with intro” workflow (/System/Library/PodcastProducer/Workflows/iTunes U with intro.pwf) is the same as the iTunes U workflow with a few additions. This workflow adds intro and exit movies to the recording and then archives a copy of the processed video before encoding takes place.

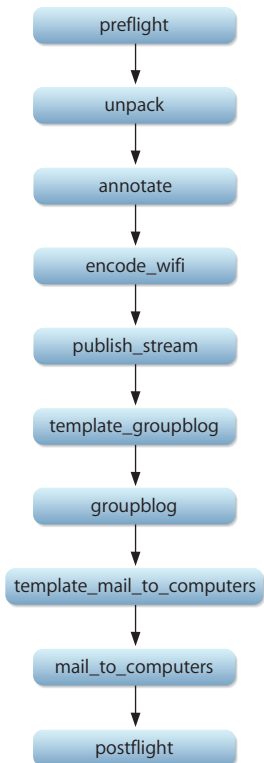
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog with streaming and no archive

The “Blog with streaming and no archive” workflow (/System/Library/PodcastProducer/Workflows/Blog with streaming and no archive.pwf) archives the recording. It also annotates the recording encoding and publishes a stream via QTSS on a group’s blog.

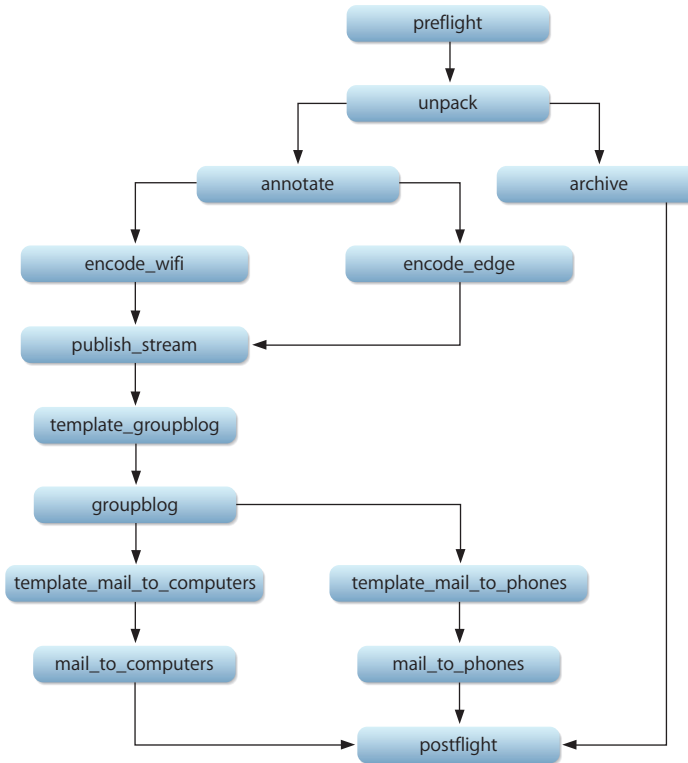
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog with streaming

The “Blog with streaming” workflow (/System/Library/PodcastProducer/Workflows/Blog with streaming.pwf) modifies the “Blog with streaming and no archive” workflow by providing computers and 3GPP-compatible telephone streams via QTSS on the group’s blog.

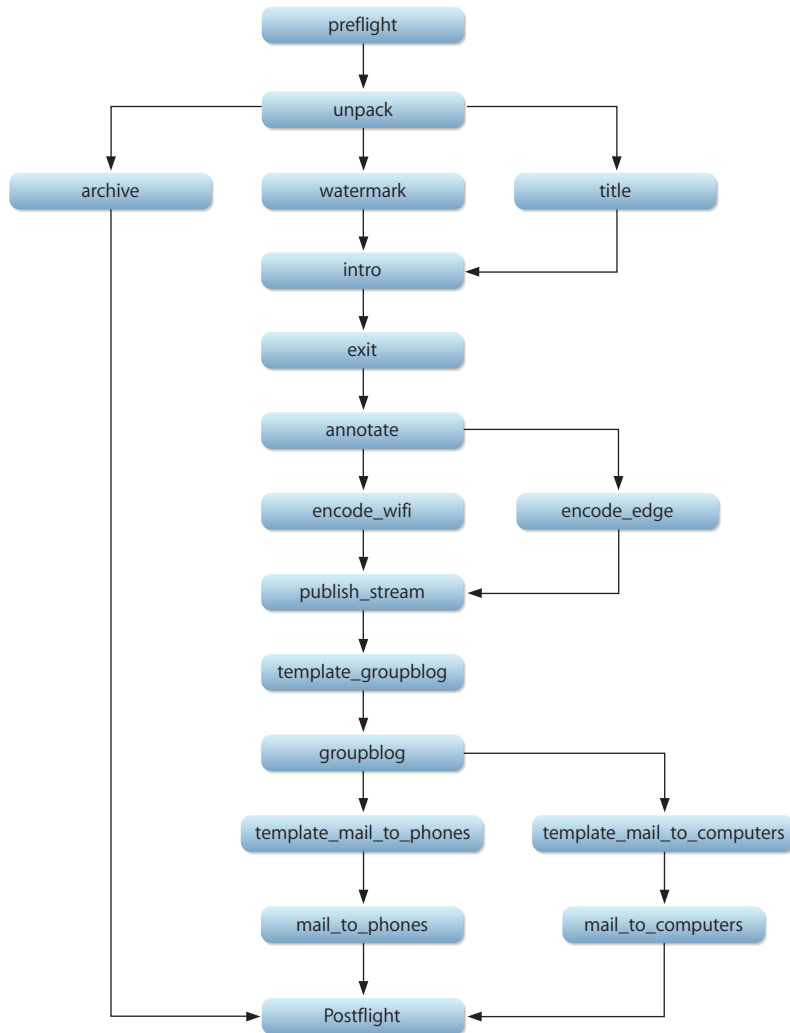
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog with streaming and intro

The “Blog with streaming and intro” workflow (/System/Library/PodcastProducer/Workflows/Blog with streaming and intro.pwf) modifies the “Blog with streaming” workflow by watermarking and adding a title to the submitted recording, adding intro and exit movies to it, and archiving the resulting movie.

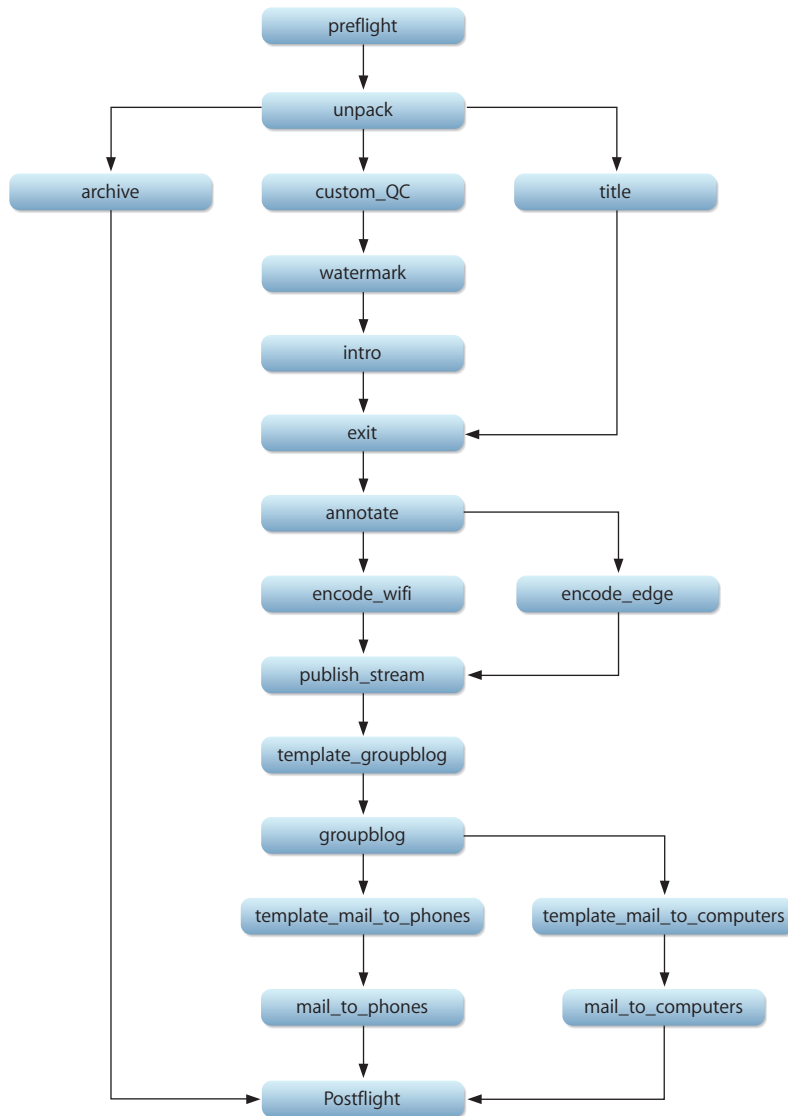
The following illustration shows the order of task execution and task dependencies in the workflow.



Blog with streaming, intro and effects

The “Blog with streaming, intro and effects” workflow (/System/Library/PodcastProducer/Workflows/Blog with streaming, intro and effects.pwf) modifies the “Blog with streaming and intro” workflow by applying a custom Quartz Composer filter to the submitted recording before adding the watermark.

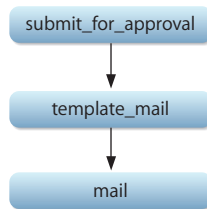
The following illustration shows the order of task execution and task dependencies in the workflow.



Submit for Approval

The Submit for Approval workflow (/System/Library/PodcastProducer/Workflows/Submit for Approval.pwf) submits content to the Approval folder on the shared file system and notifies reviewers that new content needing approval has been posted.

The following illustration shows the order of task execution and task dependencies in the workflow.



Customizing Workflows

There are several ways to customize workflows:

- Change the values of the default workflow properties.
This level of customization is very basic and gives you limited choices. For more information, see “Configuring Default Workflow Properties” on page 33.
- Modify the resources used in workflows.
- Duplicate a workflow and modify it by replacing default property keys with custom keys and adding these properties to the list of custom properties in Server Admin, as described in “Configuring Custom Workflow Properties” on page 35. Then, change the values of these custom properties in Server Admin.
Although this level of customization gives you more flexibility, it still is limited by the number of default properties you customize.
- Duplicate and modify existing workflows by adding, removing, or modifying workflows tasks. This also includes adding custom properties.
This level of customization gives you complete control over what workflows do.

Tools for Editing Workflows

A workflow template is a plist file. Therefore, you can edit it using any text editor or Property List Editor.

Modifying Workflow Resources

You can modify the resources used by workflows by replacing them with new ones.

To modify global resources used by all workflows:

- Add resources to the appropriate folder in /Library/PodcastProducer/Resources.

For example, to change the default Watermark.png image, add the new image to /Library/PodcastProducer/Resources/Images.

To modify workflow-specific resources:

- Add resources to the appropriate folder in /Library/PodcastProducer/Workflows/<workflow_name>/Contents/Resources.

Adding Custom Properties

There are cases when you want to use the same workflow but you want to change the value of certain properties based on who is using the workflow.

For example, you may want to use the same workflow for the Physics and Biology departments, but you want one workflow to post to the Physics department's blog and another to the Biology department's blog.

In this case, if you change the value of the Web URL property in Server Admin to the posting URL of the Physics department's blog, all workflows using this property will post to the Physics department's blog, which is a problem.

To solve this problem, you make multiple copies of a workflow and rename them to reflect their use (for example, Blog with streaming_Physics.pwf and Blog with streaming_Biology.pwf), create new custom properties to be used in the new workflows, and set the values of the new properties.

To add custom properties:

- 1 Duplicate the workflow bundle you want customize and store it in /Library/PodcastProducer/Workflows.

The default workflows are stored in /System/Library/PodcastProducer/Workflows.

- 2 Replace the appropriate property keys in the <workflow_name>/Contents/Resources/template.plist file with new keys.

For example, to specify a different posting web URL, replace \$\$Web URL\$\$ with \$\$Biology Web URL\$\$.

- 3 Add the new properties to the Custom Properties list in Server Admin.

For more information about adding custom properties, see "Configuring Custom Workflow Properties" on page 35.

- 4 Configure the value of the new properties in Server Admin.

- 5 If you need to change the resources used in the workflow, store the modified resources in the appropriate folder in `/Library/PodcastProducer/Resources` or relative to the workflow bundle resources (as defined by `$$Workflow Resource Path$$`).
- 6 For every new workflow:
 - Change the value of the `CFBundleName` property in the `<workflow_name>/Contents/Info.plist` file to the name of the new workflow bundle.
 - Change the value of the `CFBundleIdentifier` property in the `<workflow_name>/Contents/Info.plist` file to the appropriate value.
 - Change the Name and Description properties in the `<workflow_name>/Contents/Resources/<language>.lproj/InfoPlist.strings` file as appropriate.
- 7 Verify that you can see the new workflow in Server Admin.

Duplicating and Modifying Workflows

To create your own workflow or modify one, duplicate the workflow and make the necessary modifications.

To modify a workflow or create your own:

- 1 Duplicate a workflow bundle and store it in `/Library/PodcastProducer/Workflows`.
- 2 Add, delete, and modify tasks and custom properties as needed.
For information about workflow tasks, see “Workflow Task Specifications” on page 42.
- 3 Configure the value of the new properties in Server Admin.
- 4 If you need to change or add resources, store them in the appropriate folder in `/Library/PodcastProducer/Resources`.
- 5 Verify that you can see the new workflow in Server Admin.

Adding Localized Strings

Podcast Producer workflows are designed to allow you to add localization support for other languages.

To add localization support for a workflow:

- 1 Create an `.lproj` folder in the workflow’s Resources folder.
For example, to provide Italian support, create the folder `Italian.lproj`.
- 2 Copy `InfoPlist.strings` and `Localizable.strings` from the `en.lproj` folder and paste them into the new `.lproj` folder.
- 3 Open these files and replace the English strings with the corresponding strings in the language of your choice.
- 4 Save and close the `InfoPlist.strings` and `Localizable.strings` files.

In the Server Admin, Podcast Producer displays the localizable strings based on the Mac’s selected language.

This chapter describes the structure of the Podcast Producer shared file system.

Podcast Producer defines a shared file system that is used for storing uploaded and finished podcasts. It is also used for storing the files that Xgrid agents need to access when processing submitted recordings.

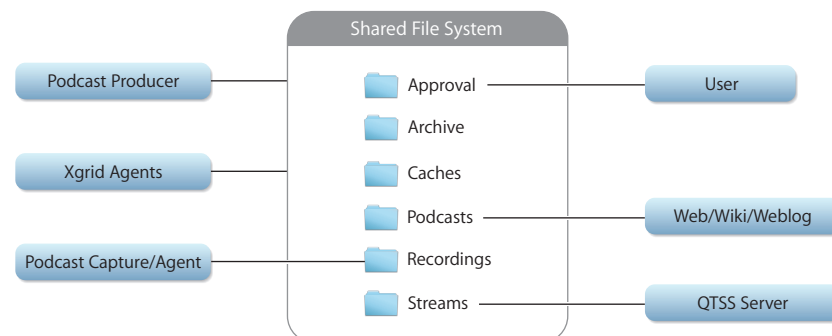
Podcast Producer supports Xsan and NFS shared file systems.

Specifying a Shared File System

You specify a shared file system using Server Admin, as described in “Configuring General Settings” on page 26.

The Structure of the Shared File System

The following illustration shows the structure of the Podcast Producer shared file system.



The following table describes the elements of the Podcast Producer shared file system.

Element	Description
Approval	The Approval folder stores content that requires approval.
Archive	The Archive folder stores copies of the submitted or processed recordings for archival purposes.
Caches	<p>The Caches folder stores any file that Podcast Producer needs to process a workflow submission.</p> <p>When you set up Podcast Producer, the shared file system does not include this folder. This folder is created only after the first workflow submission.</p> <p>The Podcast Producer server keeps the contents of the Caches folder in sync with the corresponding contents of the /Library/PodcastProducer/Workflows and /System/Library/PodcastProducer/Workflows folders on the system running the Podcast Producer.</p> <p>When synchronizing these folders, Podcast Producer checks the /Library/PodcastProducer/Workflows folder first. If the folder does not contain the files it needs, it checks the /System/Library/PodcastProducer/Workflows folder.</p> <p>Podcast Producer copies any global resources required by submitted workflow jobs from the /System/Library/PodcastProducer/Resources folder on the Podcast Producer server's Mac.</p>
Podcasts	The Podcast folder is where Podcast Producer stores the finished podcasts.
Recordings	The recordings folder stores the submitted QuickTime movie recordings. These QuickTime movies remain in this folder until the corresponding workflow job is completed successfully.
Streams	The Streams folder stores the QTSS streams produced by Podcast Producer.

Using an NFS Shared File System

The Podcast Producer server requires the ability to run the `chown` and `chmod` commands as root on the shared file system. However, by default NFS is configured to map the root user to the nobody user.

To use an NFS shared file system, configure NFS to map root to root.

For more information about configuring NFS, see *File Services Administration*.

Deploying Scalable Podcast Producer Solutions

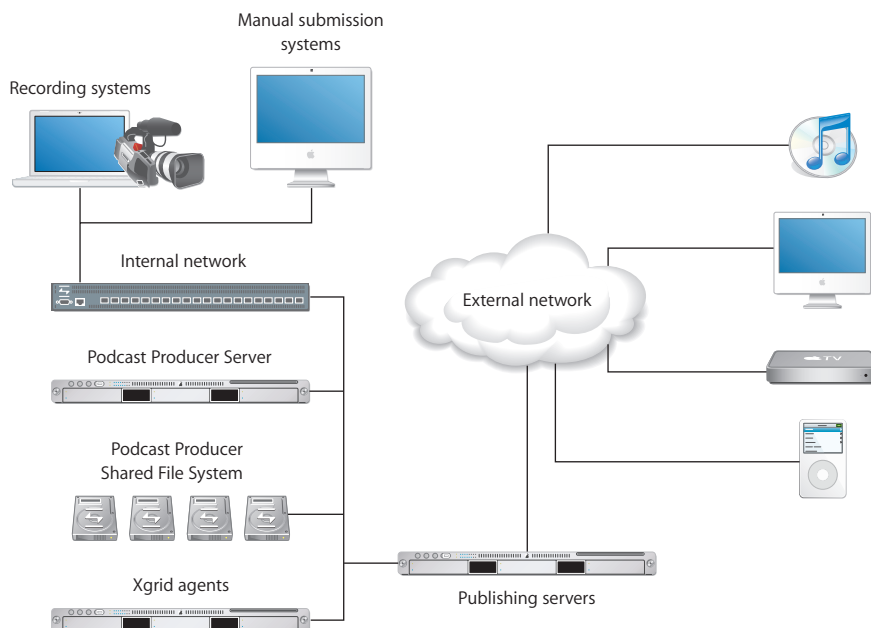
7

This chapter describes how to plan the deployment of Scalable Podcast Producer solutions.

Podcast Producer is designed for scalability. However, several factors determine how easy it is to scale your system and whether it is feasible. This chapter discusses the different scalability aspects and provides planning tips.

Resource Planning

Depending on your application, setting up Podcast Producer can require a serious investment in computing, storage, and network resources, as shown in the following illustration.



Manual Submission Systems

You use manual submission systems to upload QuickTime movies using Podcast Capture or the `podcast` command-line tool.

These systems do not need to be dedicated systems because Podcast Capture is available in Mac OS X v10.5. Users with systems running Mac OS X v10.5 can upload video content using their systems.

Any system capable of running Mac OS X v10.5 with enough hard disk space can be used for a manual submission system.

Video Recording Systems

Video recording systems are dedicated systems running Mac OS X v10.5 with a video camera connected to them. A typical video recording system is a headless Mac Mini with 40 to 60 GB of free hard disk space. These systems are remotely controlled by other systems using Podcast Capture or `podcast`.

For example, a professor could use Podcast Capture to start and stop recording on the video-recording Mac from the podium computer, which could be a MacBook. You can even write a small web application to non-Mac podium systems to control the video-recording Macs.

The number of video-recording systems depends on your needs. For example, a school might have a requirement that every classroom be equipped with a video-recording system.

Although deciding the number of systems might be a matter of policy, keep in mind the cost of acquiring and maintaining these systems. In addition, consider the impact on your network when all these systems start uploading recorded content.

Recording Quality

A very important factor to consider when planning a Podcast Producer deployment is the recording quality.

The recording quality you choose has an impact on the following:

- Storage requirements for the recording system
- Storage requirements for Podcast Producer's shared file system
- Network traffic
- Processing power

Although you can't use Server Admin or Podcast Capture to specify the recording quality, you can instead use the `podcast --presets` command. For more information about using `podcast` to specify recording quality, see the *Command-Line Administration* guide.

Recording at the Best Quality

Recording audio, video, or screen activity at the Best quality results in large QuickTime files, which require more resources to store, upload, and process. Recording video at the Best quality (DV) generates 13 GB/h. Screen recording at the Best quality can result in even bigger files because the screen resolution can be much higher than the DV resolution.

For example, a 2-hour recording at the Best quality requires more than 26 GB of free hard disk space on the recording system. In addition, while the first recorded movie is being uploaded, your recording system must have enough disk space to store a second recording or more. Otherwise, you can't use the recording system until the first movie has uploaded successfully.

To overcome this limitation, you can customize your recording systems to be connected to an Xsan system so that recordings are directly stored on the Xsan system, which provides greater storage capacity that is scalable.

Also, if the uploaded movies must be archived on the Podcast Producer's shared file system, your storage needs will increase drastically as the number of submissions increases.

In addition, the traffic on your network will increase when recording at the Best quality.

At 1 Gbit/s, it takes about 3.5 minutes to upload a 26 GB movie, assuming the network has enough bandwidth to support additional traffic. However, if the upload speed is 100 Mbit/s, the upload time is about 36 minutes under ideal circumstances. Consider the impact on the network if you have several systems uploading large files at the same time. Your network might not be able to handle such a load.

Recording at a lower Quality

As shown in the previous section, recording at the Best quality can require a significant expenditure in computing and networking resources. This is why many organizations prefer recording at a lower quality.

By default, Podcast Capture (and `podcast`) record at the Better quality (H.264 single-pass live recording at 1 GB/h for video).

The advantage of recording at this quality is that it significantly improves the efficiency of your system without compromising quality. The Better quality is almost identical to the Best quality, but the difference in size is significant, especially if you plan to use multiple recording systems on a daily basis.

The other big advantage of recording at the Better quality is that after a movie is uploaded, the Podcast Producer system can process and publish a high-quality podcast in a short period of time. The encoding of a 1-GB movie takes much less time than the encoding of a 13-GB movie.

If quality is not an issue, you can record at the Good quality, which results in even smaller files.

Network Bandwidth

When planning Podcast Producer deployment, consider using a private 1 GB/s network for submitting QuickTime movies to the Podcast Producer server. Doing so provides faster upload speed and shields the main network from traffic slowdowns when multiple systems are uploading content at the same time.

Also consider controlling the upload bandwidth at the switch level to prevent the network from being overwhelmed.

Publishing Systems

Although you can use one server to provide Podcast Producer services, including Web, Mail, and other services for users to access podcasts, consider using dedicated servers for publishing podcasts for increased reliability and better performance.

Also consider using proxy servers (for example, a proxy server for every building's network) to improve the scalability of your system.

Take into account the size of the podcasts or movies that your workflows will generate. The size of the podcasts helps you determine the number of servers you need and how much network bandwidth to allocate.

For example, the average file size for 1 hour of video encoded for iPod is 250 MB and for Apple TV is 800 MB. If you plan to serve iPod and Apple TV podcasts at an average of 400 MB per user to 1,000 users per day, you'll need servers to handle about 390 GB/day of throughput. You also need to factor in the cost per GB.

Storage

Ideally, you should use RAID arrays and Xsan to provide a scalable high-availability, high-performance storage solution for your Podcast Producer system.

In very small deployments, you can use the Podcast Producer server's hard drives for storing podcasts. However, in medium to large deployments, your storage needs can grow very large, requiring terabytes of hard disk space.

Also, the more Xgrid servers you have in your Podcast Producer system, the more data communication bandwidth you'll need. This is why Xsan is an ideal solution because it provides the necessary bandwidth to process the data.

Xgrid Agents

When planning your Podcast Producer solution, you'll need to figure out how many Xgrid agents you should deploy to provide adequate computing power and to maximize efficiency.

To determine the number of Xgrid agents needed, test the workflows you'll be using and come up with benchmarks. In many workflows, only few tasks take most of the processing power. In addition, not all tasks can be run in parallel. More Xgrid agents does not always mean better performance.

Although it might take few tries to get your system configuration right, you can always add Xgrid agents when needed to address increased computational needs.

Workflows

Workflows are very important in planning the deployment of a Podcast Producer solution. They dictate what resources will be needed and whether a deployment is feasible.

Workflows determine how submitted QuickTime movies are processed. The more processing the movies require, the more computing power and resources you'll need.

Although workflows allow you to perform batch processing for QuickTime movies, you'll need to plan your workflows based on the resources available to you.

For example, you might choose to create simplified workflows that produce only audio podcasts of conference presentations. Also, you can create workflows that do not archive the submitted QuickTime movies to save storage.

Workflow Benchmarking

When you design a workflow, test the workflow and establish benchmarks to help you plan your deployment.

Design your workflows to maximize the use of Xgrid agents and increase efficiency.

When benchmarking workflows, identify the bottleneck tasks, those that consume most of the CPU cycles available to the workflow.

Deployment Scenarios

This section discusses common deployment scenarios and describes their scalability.

Small Deployment

A small deployment is a kind of all-in-one deployment. This type of deployment is suitable for testing or for small organizations with limited computing resources and limited podcasting needs.

In a small deployment, your Podcast Producer solution can consist of the following:

- One or more recording systems
- A Podcast Producer server, which also provides storage and publishing services

This type of system is not easily scalable. To scale it up, you'll need to do the following:

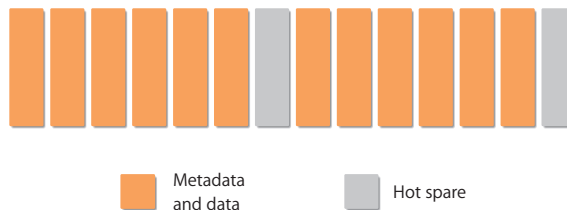
- Reconfigure your Podcast Producer settings
- Migrate the data from the old shared file system to a new one
- Possibly update your workflows to take advantage of additional processing power
- Possibly reconfigure your publishing services

Partially Scalable Deployment

The minimum configuration for a scalable deployment is the following (assuming that DNS, Mail, Open Directory, and Web services are already available):

- 1 Xserve server with Podcast Producer
- 1 Xgrid system (controller and agent)
- 1 Metadata Controller (MDC) and 1 RAID array for the Xsan system

In this configuration, you set up the metadata within the data storage pool using RAID 5 and leave two drive modules as hot spares, as shown here.



This configuration allows you to scale the computing elements easily by adding new Xserve systems with Xgrid agent enabled and Xsan component installed.

However, the storage element in this configuration does not scale because the metadata is set up within the data storage pool. This configuration optimizes the maximum usage of storage space on one RAID array but does not provide the highest scaling flexibility.

To scale the storage element of this configuration, you'll first need to migrate the data from the old RAID array to a new system where the storage area network (SAN) metadata pool and the SAN data pool are separate. Then, you can add logical unit numbers (LUNs) to the storage pool to increase the storage size and get better performance. You'll also need to reconfigure your Podcast Producer shared file system settings.

This configuration is a good starting point for organizations that do not have all the resources needed for a highly scalable deployment. It should handle a good amount of Xgrid nodes before slowing the entire system.

Highly Scalable Deployment

The perfect minimum setup that gives you maximum flexibility for scaling the computing and storage element for your Podcast Producer solution is the following (assuming that DNS, Mail, Open Directory, and Web services are already available):

- 1 Xserve server with Podcast Producer
- 1 Xgrid system (controller and agent)
- 1 MDC and 2 RAID arrays for the Xsan system

In this configuration you set up the metadata separately from the data storage pool using RAID 1, and you use RAID 5 for the data storage pool with three drive modules as hot spares, as shown here.



To scale your solution, just add very new Xgrid and RAID systems as needed.

Case Study

To better understand the importance of planning and the resource demands of a Podcast Producer solution, this section discusses a sample deployment.

This case study takes the following into consideration:

- Network bandwidth
- Content uploading
- Typical recording-day schedule with different workflow benchmarks

In addition, this case study provides charts to illustrate the results of the study.

Note: The charts in this section were generated by a special tool developed by Apple engineers. The numbers in these charts are based on established benchmarks for a set of custom workflows. These numbers may not apply to your setup.

Recording System Configuration

The Podcast Producer system discussed in this case study has the following configuration for recording video:

- 4 recording systems (no manual submission systems) to record and upload content
- A default network bandwidth of 100Mbit/s from the recording systems to the Podcast Producer server
- A recording quality of H.264 (1 GB/h)

Workflow Benchmarks

This sample deployment uses four workflows. The following table lists the benchmarks established for these workflows after testing them:

Workflow	Description	Ratio	CPUs	In (MB/h)	Out (MB/h)
1	This workflow takes in a 720x576 movie (less than 90 minutes long) and produces three podcasts: <ul style="list-style-type: none">• AAC Audio• iPod video (H264)• High-quality video (H264/native resolution)	100%	2	250	500
2	This workflow is the same as workflow 1 except that it takes in a movie more than 90 minutes long.	150%	2	250	500
3	This workflow takes the same input and generates the same output as workflow 1, but also generates additional podcasts.	100%	4	250	1,050
4	This workflow is the same as workflow 3 except that it takes in a movie more than 90 minutes long.	150%	4	250	1,050

The Ratio column in the table lists the ratio of recording time to processing time. For example, if the recording time is 1 hour, a ratio of 150% means that it takes 1.5 hours to produce and publish the podcast. The CPUs column lists the number of CPUs (not systems) required by the workflow for best performance. The In column lists the size per hour of the recorded movie and the Out column lists the per hour size of the resulting podcasts.

Recording Schedule

A typical recording day (from 7 a.m. to 7:00 p.m.) for this case study is illustrated below.

Room	Recording Times (in Half Hour Increments)																					
	7	8	9	10	11	12	1	2	3	4	5	6										
106		1	1	1	1		2	2	2	2		3	3	3	3		4	4	4	4		
110				1	1	1	1		2	2	2	2		3	3	3	3		4	4	4	4
112			1	1	1	1		2	2	2	2		3	3	3	3		4	4	4	4	
114	1	1	1	1		2	2	2	2			3	3	3	3		4	4	4	4		

The numbers in the colored cells represent the workflow being used. In this example, workflow 1 is used to record the first sessions in all rooms, workflow 2 is used for recording the following sessions in all four rooms, and so on.

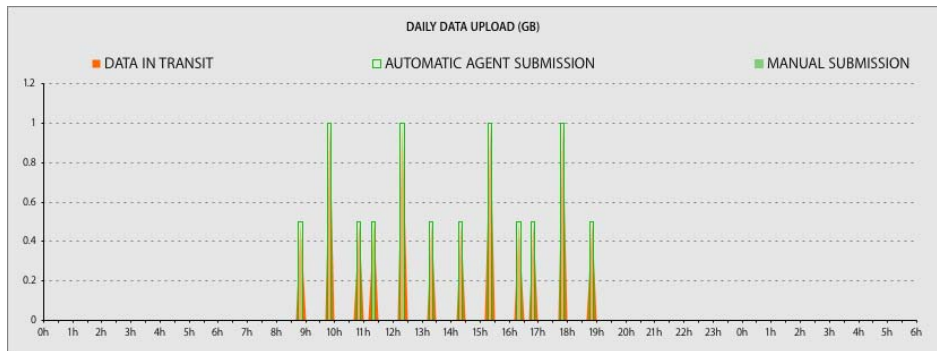
The colored cells help you see the recording pattern.

Performance

With one system for uploading the recorded movies to the shared file system and three Xgrid nodes, this section describes the performance of the system.

Daily Data Upload

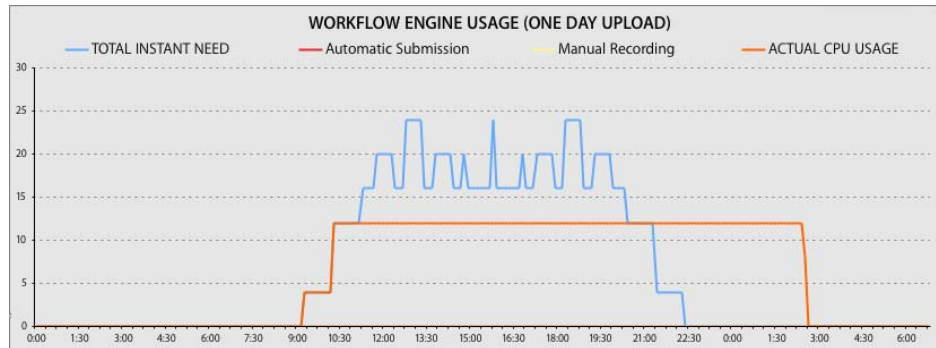
The following chart illustrates the daily upload times.



In this case, the size of recorded movies and the spacing of recording times allows for a smooth uploading pattern of 15 minutes per submission.

CPU Usage

The following chart illustrates the CPU usage pattern when using three Xgrid nodes.

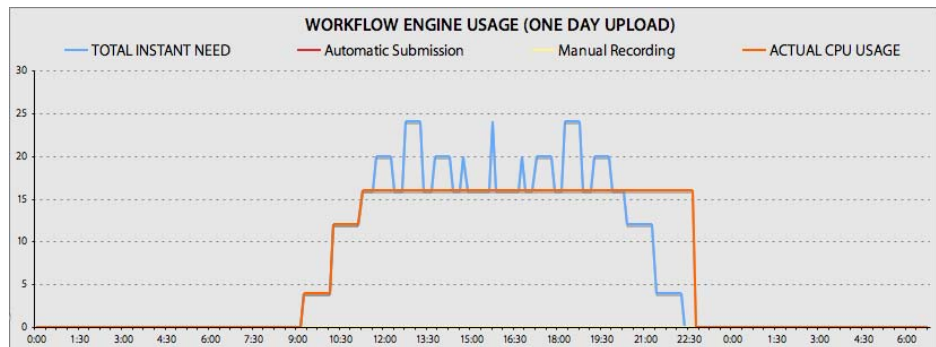


In the chart above, the blue line shows the instant computing need if unlimited computing resources were available. The orange line shows the actual CPU usage based on the number of Xgrid nodes available. In this case, there are three Xgrid nodes, each with four CPUs.

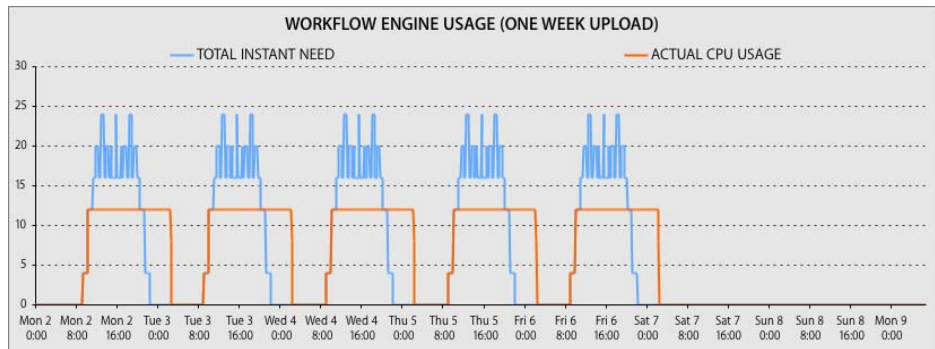
As shown in the chart above, at 3:00 in the morning, all workflows completed and all content was published.

Note: From a mathematical point of view, the area under the blue line is equal to the area under the orange line.

If the number of Xgrid nodes is now four (16 CPUs), the performance will improve as shown in the following chart.



However, using only two Xgrid nodes will do the job if fast delivery of content is not required, as shown in the following chart.



As indicated in the chart, all podcasts will be available six days after the recording started.

The choice of how many Xgrid nodes to use depends on delivery time requirements and cost.

Storage Usage

The following shows the required Xsan setup for an academic year (40 weeks) to accommodate the recording schedule used in this case study:

- Uploaded data per day: 8,500 MB
- Processed data per day: 35,700 MB per day

The estimated total storage need for 40 weeks is 8,840,000 MB, or about 8.6 TB.

Summary

Based on this case study, the following factors play a big role in the planning and deploying of a Podcast Producer solution:

- Number of recording and manual submission systems
- Type of workflows to be used
- Typical recording-day schedule for each recording and manual submission system
- Estimated time to complete a workflow job under best conditions
- Number of Xgrid nodes and number of CPUs that can be used in parallel
- Required delivery time and system optimization needed to meet requirements
- Required Xsan storage
- Required Xsan bandwidth per Xgrid agent to ensure that tasks execute at full speed
- Required network bandwidth

This chapter describes Podcast Producer command-line tools.

Podcast Producer command-line tools provide you with the flexibility you need to customize Podcast Producer for your needs. This chapter provides a high-level overview of these commands.

For more information about the Podcast Producer command-line tools, see the *Command-Line Administration* guide or the corresponding man page.

The `podcast` Tool

The `/usr/bin/podcast` tool provides complete scriptable control over Podcast Producer client-side functionality. It allows you to perform any function that Podcast Capture offers, and more. For example, you can use `podcast` for listing and setting audio, video, and screen recording quality.

The following are ways in which you can take advantage of the `podcast` tool:

- Wrapping the `podcast` tool.

You can wrap the `podcast` tool to add a layer of functionality.

For example, you can write a GUI application that authenticates users and provides only one option for starting and stopping screen recording.

- Creating Podcast Producer widgets.

For example, you can create a widget for viewing Podcast Producer presets.

- Scheduling recording jobs.

You can create `cron` jobs to automate the recording and submission of audio and video content to the Podcast Producer server.

For example, you can write a `cron` job to record lectures. You can also write `cron` jobs for submitting daily Podcast Producer jobs after 11:00 p.m.

The pcastconfig Tool

This is the server configuration command-line tool (`/usr/bin/pcastconfig`). It provides all the functionality offered by the Podcast Producer service in Server Admin.

The pcastctl Tool

Use the `/usr/sbin/pcastctl` tool to start, stop, and restart the Podcast Producer server or agent. Also use this tool to display the status of running daemons.

The pcastaction Tool

The `/usr/bin/pcastaction` tool is used in workflows and provides a rich set of commands for processing and producing audio and video podcasts.

For example, the `pcastaction watermark` command imposes a watermark image on the input video. Also, the `pcastaction encode` command outputs an encoded version of the input file.

This chapter describes how to monitor and troubleshoot Podcast Producer issues.

Podcast Producer provides several ways for monitoring Podcast Producer activity.

Viewing Podcast Producer Logs

You can use the Logs pane of the Podcast Producer server to view Podcast Producer logs.

The logs help you monitor and troubleshoot Podcast Producer issues. You can even write scripts that look for certain log entries to alert you of possible issues.

Podcast Producer provides the following logs:

Log	Description
Podcast Producer Server Log	Records Podcast Producer server (<code>pcastserverd</code>) activity.
Podcast Producer Server Error Log	Records error messages generated by the Podcast Producer server.
Podcast Producer HTTP Access Log	Records all requests processed by the Apache server instance (<code>httpd</code>) used by Podcast Producer.
Podcast Producer HTTP Error Log	Records HTTP error messages generated by Podcast Producer's <code>httpd</code> instance.
Podcast Producer Application Log	Records external HTTP requests to the Podcast Producer server.

To view Podcast Producer logs using Server Admin:

- 1 Open Server Admin.
- 2 In the Computers and Services list, select Podcast Producer.
- 3 Click Logs.
- 4 From the View pop-up menu, choose the log to view.

Monitoring Movie Transfers

Podcast Capture users can monitor the progress of movie submissions to the Podcast Producer server using the Transfers window, as described in the onscreen help for Podcast Capture.

You can also look at the system log on the Podcast Producer server for Podcast Producer uploader entries.

Monitoring Xgrid Job Progress

The Podcast Producer server sends a notification message when a Podcast Producer Xgrid job completes successfully, but you may want to actively monitor the progress of an Xgrid job.

Using Xgrid Admin

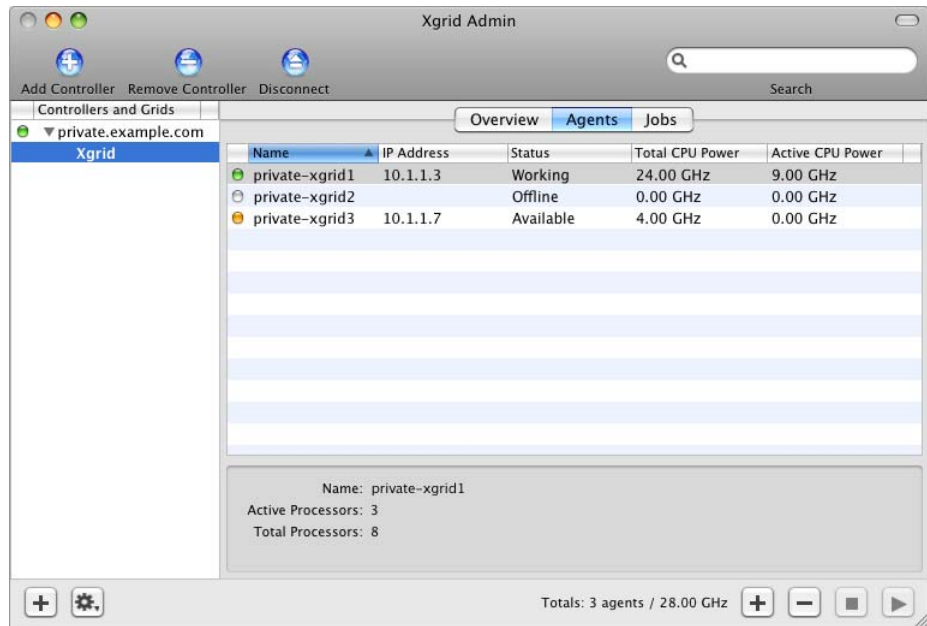
To actively monitor Xgrid job progress, use Xgrid Admin. Xgrid Admin shows you the progress of Xgrid jobs and whether they succeeded or failed, as shown in the following figure.



In the figure above, Xgrid Admin shows that the job titled History 1, which was submitted by Anne Johnson, has been completed. The text between parentheses is the name of the workflow selected by her in Podcast Capture before submitting the job.

Also in the figure, Xgrid Admin shows that the job submitted by John Clark is still running and is about 40% done. This job has 22 tasks and is currently using 12.40 GHz of CPU power.

In addition to monitoring job progress, you can use Xgrid Admin to see the status of Xgrid agents, as shown below.



For more information about Xgrid Admin, see *Xgrid Administration and High Performance Computing*.

Using the Command Line

You can use the `xgrid` command-line tool to monitor Podcast Producer Xgrid job progress.

For example, when the Podcast Producer server submits a job to the Xgrid controller, an entry similar to the following appears in the Podcast Producer Server Log:

```
Thu Aug 23 13:20:33 -0700 2007 -- Wrote Xgrid job batch file: /var/pcast/  
server/xgrid_jobs/9A93EA1B-1A59-43DE-B786-1537EC0CD479_job.xml
```

This entry tells you that the Podcast Producer server created an Xgrid job file based on a workflow template and stored it in the `/var/pcast/server/xgrid_jobs/` directory.

Following this entry is another entry indicating the job number:

```
Thu Aug 23 13:20:34 -0700 2007 -- Kicked off Xgrid Job: 594 for Podcast  
Producer Job: 598
```

In this example, the job's ID is 594.

You can then use the Xgrid command-line tool to find out the results of running the tasks specified in the job using the job's ID, as in the following example:

```
$ xgrid -h private.example.com -auth Kerberos -job results -id 594
Page created at: http://private.example.com/groups/podcasts/blog/68912
Running /Volumes/Podcast/Caches/Resources/Tools/preflight_script
```

You can even narrow the Xgrid job's status results to show a specific task:

```
$ xgrid -h private.example.com -auth Kerberos -job results -id 594 -tid
groupblog
Page created at: http://private.example.com/groups/podcasts/blog/68912
```

In this example, the output indicates that the groupblog task created a page at the specified URL.

You can also view Xgrid job status as in the following example:

```
$ xgrid -h private.example.com -auth Kerberos -job attributes -id 594
{
  jobAttributes = {
    activeCPUPower = 0;
    dateNow = 2007-08-23 13:34:01 -0700;
    dateStarted = 2007-08-23 12:22:52 -0700;
    dateStopped = 2007-08-23 13:23:10 -0700;
    dateSubmitted = 2007-08-23 12:22:52 -0700;
    jobStatus = Finished;
    name = "History 1 by Anne Johnson (Blog and iTunes with intro and
effects)";
    percentDone = 100;
    taskCount = 22;
    undoneTaskCount = 0;
  };
}
```

For more information about using the Xgrid command-line tool, see *Command-Line Administration*.

AFP Apple Filing Protocol. A client/server protocol used by Apple file service on Macintosh-compatible computers to share files and network services. AFP uses TCP/IP and other protocols to communicate between computers on a network.

address A number or other identifier that uniquely identifies a computer on a network, a block of data stored on a disk, or a location in a computer memory. See also **IP address**, **MAC address**.

administrator A user with server or directory domain administration privileges. Administrators are always members of the predefined “admin” group.

alias Another email address at your domain that redirects incoming email to an existing user.

Apple Filing Protocol See **AFP**.

automount To make a share point appear automatically on a client computer. See also **mount**.

bit A single piece of information, with a value of either 0 or 1.

Bonjour A protocol developed by Apple for automatic discovery of computers, devices, and services on IP networks. This proposed Internet standard protocol is sometimes referred to as “ZeroConf.” For more information, visit www.apple.com or www.zeroconf.org. To see how this protocol is used in Mac OS X Server, see **local hostname**.

client A computer (or a user of the computer) that requests data or services from another computer, or server.

command line The text you type at a shell prompt when using a command-line interface.

command-line interface A way of interfacing with the computer (for example, to run programs or modify file system permissions) by entering text commands at a shell prompt.

daemon A program that runs in the background and provides important system services, such as processing incoming email or handling requests from the network.

DHCP Dynamic Host Configuration Protocol. A protocol used to dynamically distribute IP addresses to client computers. Each time a client computer starts up, the protocol looks for a DHCP server and then requests an IP address from the DHCP server it finds. The DHCP server checks for an available IP address and sends it to the client computer along with a lease period—the length of time the client computer may use the address.

directory Also known as a folder. A hierarchically organized list of files or other folders, or probably both.

directory domain A specialized database that stores authoritative information about users and network resources; the information is needed by system software and applications. The database is optimized to handle many requests for information and to find and retrieve information quickly. Also called a directory node or simply a directory.

DNS Domain Name System. A distributed database that maps IP addresses to domain names. A DNS server, also known as a name server, keeps a list of names and the IP addresses associated with each name.

DNS domain A unique name of a computer used in the Domain Name System to translate IP addresses and names. Also called a **domain name**.

DNS name A unique name of a computer used in the Domain Name System to translate IP addresses and names. Also called a **domain name**.

domain Part of the domain name of a computer on the Internet. It does not include the Top Level Domain designator (for example, .com, .net, .us, .uk). Domain name “www.example.com” consists of the subdomain or host name “www,” the domain “example,” and the top level domain “com.”

domain name See **DNS name**.

Domain Name System See **DNS**.

drop box A shared folder with privileges that permit other users to write to, but not read, the folder’s contents. Only the owner has full access. Use only AFP to create drop boxes. When a folder is shared using AFP, the ownership of an item written to the folder is automatically transferred to the owner of the folder, thus giving the owner of a drop box full access to and control over items put into it.

everyone Any user who can log in to a file server: a registered user or guest, an anonymous FTP user, or a website visitor.

export In the Network File System (NFS), a way of sharing a directory with clients on a network.

file server A computer that serves files to clients. A file server may be a general-purpose computer that's capable of hosting additional applications or a computer capable only of serving files.

file system A scheme for storing data on storage devices that permits applications to read and write files without having to deal with lower-level details.

File Transfer Protocol See **FTP**.

FTP File Transfer Protocol. A protocol that enables computers to transfer files over a network. FTP clients using any operating system that supports FTP can connect to a file server and download files, depending on their access privileges. Most Internet browsers and a number of freeware applications can be used to access an FTP server.

group A collection of users who have similar needs. Groups simplify the administration of shared resources.

guest user A user who can log in to your server without a user name or password.

home directory A folder for a user's personal use. Mac OS X also uses the home directory, for example, to store system preferences and managed user settings for Mac OS X users.

host Another name for a server.

host name A unique name for a server, historically referred to as the UNIX hostname. The Mac OS X Server host name is used primarily for client access to NFS home directories. A server determines its host name by using the first name available from the following sources: the name specified in the `/etc/hostconfig` file (`HOSTNAME=some-host-name`); the name provided by the DHCP or BootP server for the primary IP address; the first name returned by a reverse DNS (address-to-name) query for the primary IP address; the local hostname; the name "localhost."

Internet Generally speaking, a set of interconnected computer networks communicating through a common protocol (TCP/IP). The Internet (note the capitalization) is the most extensive publicly accessible system of interconnected computer networks in the world.

Internet Protocol See **IP**.

IP Internet Protocol. Also known as IPv4. A method used with Transmission Control Protocol (TCP) to send data between computers over a local network or the Internet. IP delivers packets of data, while TCP keeps track of data packets.

IP address A unique numeric address that identifies a computer on the Internet.

IP subnet A portion of an IP network, which may be a physically independent network segment, that shares a network address with other portions of the network and is identified by a subnet number.

Kerberos A secure network authentication system. Kerberos uses tickets, which are issued for a specific user, service, and period of time. Once a user is authenticated, it's possible to access additional services without retyping a password (this is called single sign-on) for services that have been configured to take Kerberos tickets. Mac OS X Server uses Kerberos v5.

LDAP Lightweight Directory Access Protocol. A standard client-server protocol for accessing a directory domain.

Line Printer Remote See **LPR**.

local hostname A name that designates a computer on a local subnet. It can be used without a global DNS system to resolve names to IP addresses. It consists of lowercase letters, numbers, or hyphens (except as the last characters), and ends with ".local" (e.g, bills-computer.local). Although the name is derived by default from the computer name, a user can specify this name in the Network pane of System Preferences. It can be changed easily, and can be used anywhere a DNS name or fully qualified domain name is used. It can only resolve on the same subnet as the computer using it.

LPR Line Printer Remote. A standard protocol for printing over TCP/IP.

Mac OS X The latest version of the Apple operating system. Mac OS X combines the reliability of UNIX with the ease of use of Macintosh.

Mac OS X Server An industrial-strength server platform that supports Mac, Windows, UNIX, and Linux clients out of the box and provides a suite of scalable workgroup and network services plus advanced remote management tools.

mount (verb) In general, to make a remote directory or volume available for access on a local system. In Xsan, to cause an Xsan volume to appear on a client's desktop, just like a local disk.

Network File System See **NFS**.

network interface Your computer's hardware connection to a network. This includes (but isn't limited to) Ethernet connections, AirPort cards, and FireWire connections.

NFS Network File System. A client/server protocol that uses Internet Protocol (IP) to enable remote users to access files as though they were local. NFS exports shared volumes to computers according to IP address, rather than user name and password.

nfsd daemon An NFS server process that runs continuously behind the scenes and processes read and write requests from clients. The more daemons that are available, the more concurrent clients can be served.

Open Directory The Apple directory services architecture, which can access authoritative information about users and network resources from directory domains that use LDAP, NetInfo, or Active Directory protocols; BSD configuration files; and network services.

open source A term for the cooperative development of software by the Internet community. The basic principle is to involve as many people as possible in writing and debugging code by publishing the source code and encouraging the formation of a large community of developers who will submit modifications and enhancements.

oplocks See **opportunistic locking**.

opportunistic locking Also known as oplocks. A feature of Windows services that prevents users of shared files from changing the same file at the same time. Opportunistic locking locks the file or part of the file for exclusive use, but also caches the user's changes locally on the client computer for improved performance.

owner The owner of an item can change access permissions to the item. The owner may also change the group entry to any group in which the owner is a member. By default the owner has Read & Write permissions.

password An alphanumeric string used to authenticate the identity of a user or to authorize access to files or services.

pathname The location of an item within a file system, represented as a series of names separated by slashes (/).

permissions Settings that define the kind of access users have to shared items in a file system. You can assign four types of permissions to a share point, folder, or file: read/write, read-only, write-only, and none (no access). See also **privileges**.

print queue An orderly waiting area where print jobs wait until a printer is available. The print service in Mac OS X Server uses print queues on the server to facilitate management.

port A sort of virtual mail slot. A server uses port numbers to determine which application is to receive data packets. Firewalls use port numbers to determine whether data packets are permitted to traverse a local network. "Port" usually refers to either a TCP or UDP port.

privileges The right to access restricted areas of a system or perform certain tasks (such as management tasks) in the system.

process A program that has started executing and has a portion of memory allocated to it.

protocol A set of rules that determines how data is sent back and forth between two applications.

QTSS QuickTime Streaming Server. A technology that lets you deliver media over the Internet in real time.

QuickTime A set of Macintosh system extensions or a Windows dynamic-link library that supports the composition and playing of movies.

QuickTime Streaming Server See **QTSS**.

queue An orderly waiting area where items wait for some type of attention from the system. See also **print queue**.

server A computer that provides services (such as file service, mail service, or web service) to other computers or network devices.

share point A folder, hard disk (or hard disk partition), or CD that's accessible over the network. A share point is the point of access at the top level of a group of shared items. Share points can be shared using AFP, Windows SMB, NFS (an "export"), or FTP protocols.

short name An abbreviated name for a user. The short name is used by Mac OS X for home directories, authentication, and email addresses.

single sign-on An authentication strategy that relieves users from entering a name and password separately for every network service. Mac OS X Server uses Kerberos to enable single sign-on.

SLP DA Service Location Protocol Directory Agent. A protocol that registers services available on a network and gives users easy access to them. When a service is added to the network, the service uses SLP to register itself on the network. SLP/DA uses a centralized repository for registered network services.

TCP Transmission Control Protocol. A method used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. IP takes care of handling the actual delivery of the data, and TCP takes care of keeping track of the individual units of data (called packets) into which a message is divided for efficient routing through the Internet.

ticket, Kerberos A temporary credential that proves a Kerberos client's identity to a service.

Transmission Control Protocol See **TCP**.

UDP User Datagram Protocol. A communications method that uses the Internet Protocol (IP) to send a data unit (called a datagram) from one computer to another in a network. Network applications that have very small data units to exchange may use UDP rather than TCP.

UID User ID. A number that uniquely identifies a user within a file system. Mac OS X computers use the UID to keep track of a user's directory and file ownership.

URL Uniform Resource Locator. The address of a computer, file, or resource that can be accessed on a local network or the Internet. The URL is made up of the name of the protocol needed to access the resource, a domain name that identifies a specific computer on the Internet, and a hierarchical description of a file location on the computer.

USB Universal Serial Bus. A standard for communicating between a computer and external peripherals using an inexpensive direct-connect cable.

User Datagram Protocol See **UDP**.

user ID See **UID**.

user name The long name for a user, sometimes referred to as the user's "real" name. See also **short name**.

volume A mountable allocation of storage that behaves, from the client's perspective, like a local hard disk, hard disk partition, or network volume. In Xsan, a volume consists of one or more storage pools. See also **logical disk**.

WebDAV Web-based Distributed Authoring and Versioning. A live authoring environment that enables client users to check out webpages, make changes, and then check the pages back in while a site is running.

WINS Windows Internet Naming Service. A name resolution service used by Windows computers to match client names with IP addresses. A WINS server can be located on the local network or externally on the Internet.

A

- access
 - ACLs 18
 - camera 37
 - workflow 31
- ACLs (access control lists) 18
- administrator 33
- AES (Advance Encryption Standard) 18
- agents
 - Podcast Producer 16
 - Xgrid 16, 17, 18, 41, 68
- annotations, movie 51, 52, 53, 55, 58, 59
- Apple TV format 53
- approval 33
- archiving
 - file system organization 64
 - iTunes U workflow 54, 55
 - QuickTime movies 49, 50, 51
 - root folder 33
- arrays, disk. *See* RAID
- audience, properties for 33
- authentication 18
 - See also* passwords

B

- bandwidth for QuickTime movies 22, 68
- batch processing, QuickTime movies 69
- BEEP (Blocks Extensible Exchange Protocol) 18
- binding, Mac to Podcast Producer 29
- Blocks Extensible Exchange Protocol. *See* BEEP
- Blog and iTunes with intro and effects for Apple TV workflow 53
- Blog and iTunes with intro and effects workflow 52
- Blog and iTunes with intro workflow 51
- Blog and iTunes with watermark workflow 50
- blogs. *See* blog service
- blog service
 - Blog and iTunes with intro and effects for Apple TV workflow 53
 - Blog and iTunes with intro and effects workflow 52
 - Blog and iTunes with intro workflow 51

- Blog and iTunes with watermark workflow 50
- Blog without archive workflow 48
- Blog with streaming, intro and effects workflow 59
- Blog with streaming and intro workflow 58
- Blog with streaming and no archive workflow 56
- Blog with streaming workflow 57
- Blog workflow 49
 - introduction 25
- Blog without archive workflow 48
- Blog with streaming, intro and effects workflow 59
- Blog with streaming and intro workflow 58
- Blog with streaming and no archive workflow 56
- Blog with streaming workflow 57
- Blog workflow 49
- browsers, web 14

C

- caches, file system organization 64
- camera 15, 37, 38, 66
- capture, video. *See* Podcast Capture
- clients and secure server communication 18
 - See also* users
- clusters, Xgrid 14
- codecs 19, 51, 53
- command-line tools
 - podcast** 15, 18, 66, 77
 - tool summary 77, 78
 - workflow tools 46
 - Xgrid 18
- configuration
 - default workflow 28
 - DNS 23
 - general settings 26, 28
 - preliminaries 22
 - services 23, 24, 25, 26
 - users 24
 - verifying 30
 - workflow properties 33, 35, 36
 - Xgrid 26
- controllers, Xgrid 16, 18, 27
- copyright notice 33

D

dependencies, task 45
directory services 18, 22, 24
disk arrays. *See* RAID
disks. *See* storage considerations
DNS (Domain Name System) service 23
documentation 8, 10
Domain Name System. *See* DNS
domains, directory, Open Directory 18, 24

E

email. *See* mail service
email list 33
encodings, workflow 40
encryption 18

F

files
 plist files 41, 61
 shared secret 34
 size considerations 68
 storage considerations 68
file sharing 16, 27, 34, 63, 64
filters
 camera 38
 workflow 32
folder 33
folders
 archive root 33
 shared 27, 64
 web root 34

G

groups 18, 33

H

hardware requirements 21
help, using 8
hosts. *See* servers

I

images 19, 34, 52
iPod formats 51
iTunes 14, 34
iTunes U with intro workflow 55
iTunes U workflow 54

J

jobs, Xgrid 16, 41

K

KDC (Kerberos Key Distribution Center). *See* Kerberos
KEK (Key Exchange Key) 18
Kerberos 18

Key Exchange Key. *See* KEK
keys, property 43, 44

L

localization support 62
logical unit numbers. *See* LUNs
logs 79
LUNs (logical unit numbers) 70

M

Mac OS X, manual submission systems 66
mail service 23, 33, 34
manual submission systems 66
movies. *See* QuickTime movies

N

network traffic and recording quality 67
notification, email 23, 34, 41

O

Open Directory 18, 24

P

passwords 18, 33, 43
pcastaction tool 78
pcastagentd daemon 18
pcastconfig tool 78
pcastctl tool 78
plist files 41, 61
Podcast Capture
 binding operation 29
 computer requirements 21
 manual submission 66
 overview 15
 shared file system 27
 verifying setup 30
 workflow access control 31
podcast folder 64
Podcast Producer
 architecture 14, 15
 customization overview 19
 deployment scenarios 69, 71, 73, 74, 75
 logs 79
 overview 7, 13, 17
 resource planning 65, 66, 67, 68
 starting 28
 transfer monitoring 80
 troubleshooting 80
 See also configuration
podcast tool 15, 18, 66, 77
properties
 custom 35, 36, 61
 list of workflow 33, 35, 36, 60
publishing, podcast 19, 68

Q

QTSS. *See* QuickTime Streaming Server

Quartz Composer 19, 34, 52

QuickTime movies

- archiving 49, 50, 51
- bandwidth considerations 22, 68
- batch processing 69
- capabilities 16
- manual submission 66
- path locations 33
- and Podcast Producer 13, 15, 16
- quality and file size 68
- Xgrid capacity considerations 22

QuickTime Streaming Server (QTSS)

- blog workflow options 56, 57, 58, 59
- file storage 64
- properties 34
- setup 26

R

RAID (Redundant Array of Independent Disks) 68, 70, 71

recording

- file storage 64
- quality of 66, 67
- and resource planning 66, 72

Redundant Array of Independent Disks. *See* RAID requirements, hardware and software 21

S

Safari 14

SANs (storage area networks) 67, 68, 70, 71, 75

scalable deployment scenarios 69, 71

scripts, shared file 34

Secure Sockets Layer. *See* SSL

security

- access control 18, 31, 37
- authentication 18
- overview 17, 18, 19
- passwords 18, 33, 43
- SSL 18

Server Admin 18

servers

- computer requirements 21
- hosting 19
- Podcast Producer 13, 15
- secure communication 18
- web 33, 34

See also QuickTime Streaming Server

setup procedures. *See* configuration

shared files. *See* file sharing

shared secret files 34

shell scripts 42

Short Message Service. *See* SMS

short name 33

Simple Mail Transfer Protocol. *See* SMTP

small deployment scenario 69

SMS (Short Message Service) 33

SMTP (Simple Mail Transfer Protocol) 34

software requirements 21

SSL (Secure Sockets Layer) 18

storage area networks. *See* SANs

storage considerations

- RAID 68, 70, 71
- SANs 67, 68, 70, 71, 75
- workflow options 39
- and Xgrid 68
- See also* archiving

streaming media. *See* QuickTime Streaming Server

Submit for Approval workflow 60

T

task specifications

- introduction 42, 45
- iTunes U with intro workflow 55
- iTunes U workflow 54
- Submit for Approval workflow 60
- See also* blog service

templates, workflow 41, 42, 43

The 78

transfers, monitoring movie 80

U

UNIX shell script 19, 42

usernames, group administrator 33

users

- authentication 18
- creating 24
- groups 18, 33
- posting credentials 34
- secure communication 18

V

video capture system. *See* Podcast Capture

videos. *See* QuickTime movies

W

watermarks 34, 50, 51

web browsers 14

web server 33, 34

web services 25

See also blog service

workflows

- access control 31
- benchmarking of 69, 72
- bundle structure 39, 40
- commands list 46
- customizing 60, 61, 62
- default settings 28
- default types 47

- deployment planning 69
- editing tools 60
- encodings 40
- management of 31, 32, 33, 35, 36
- overview 16, 22
- pcastaction** tool 78
- properties 33, 35, 36, 60
- resources 40, 61
- storage options 39
- templates 41, 42, 43
- See also* task specifications
- Workgroup Manager 24

X

- Xgrid
 - agents 16, 17, 18, 41, 68
 - capacity considerations 22
 - clusters 14
 - controllers 16, 18, 27
 - jobs 16, 41
 - setup 26
 - and storage considerations 68
 - task dependencies 45
 - workflow overview 16
- xgrid** tool 18
- Xsan 67, 68, 71, 75