Mac OS X Server
Podcast Producer Administration

For Version 10.6 Snow Leopard

Apple

1 Infinite Loop
Cupertino, CA 95014-2084
408-996-1010
www.apple.com

Simultaneously published in the United States and Canada.

019-1419/2009-05-29

Contents

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 publishing podcasts of lectures, training, and 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, "Upgrading to Podcast Producer 2" describes how to upgrade your computer running Podcast Producer 1 to Podcast Producer 2.
- Chapter 4, "Setting Up Podcast Producer for High Availability" describes how to configure Podcast Producer for failover.
- Chapter 5, "Podcast Library" describes the Podcast Producer library.
- Chapter 6, "Managing Workflows" provides a high-level overview of workflows and how to manage them using Server Admin.
- Chapter 7, "Managing Cameras" describes how to manage and monitor camera usage using Server Admin.
- Chapter 8, "Managing Feeds" describes how to manage Podcast Library feeds.
- Chapter 9, "Customizing Workflows" describes how to customize workflows.
- Chapter 10, "Deploying Scalable Podcast Producer Solutions" describes how to plan the deployment of Podcast Producer solutions.
- Chapter 11, "Podcast Producer Command-Line Tools" describes the Podcast Producer command-line tools.

- Chapter 12, "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 Help Viewer while you're managing Mac OS X Server. You can view help on a server, or on an administrator computer. (An administrator computer is a Mac OS X computer with Mac OS X Server administrator software installed on it.)

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

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

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

To see the most recent server help topics:

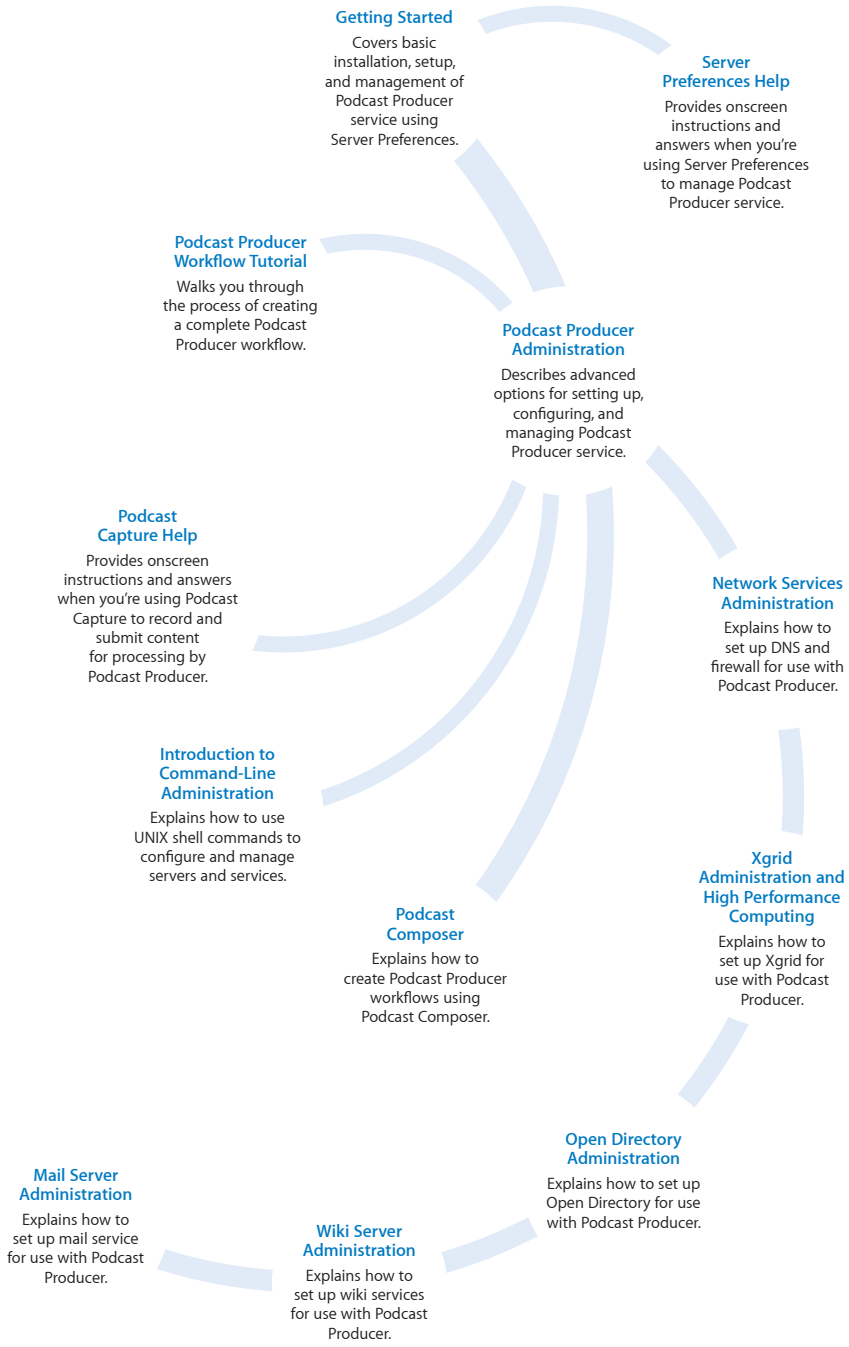
- Make sure the server or administrator computer is connected to the Internet while you're getting help.

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

Documentation Map

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

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



Viewing PDF Guides Onscreen

While reading the PDF version of a guide 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 Resources website at: www.apple.com/server/resources/

- An RSS feed listing the latest updates to Mac OS X Server documentation and onscreen help is available. To view the feed, use an RSS reader application such as Safari or Mail and go to: feed://helposx.apple.com/rss/snowleopard/serverdocupdates.xml

Getting Additional Information

For more information, consult these resources:

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

Overview of Podcast Producer

1

This chapter introduces Podcast Producer and describes its architecture.

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.

What's New in Podcast Producer

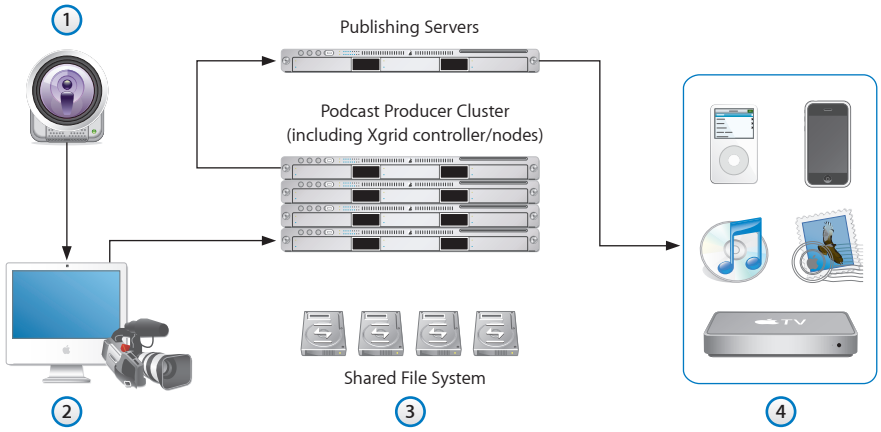
Mac OS X Server v10.6 includes a new version of Podcast Producer, Podcast Producer 2, which offers major enhancements:

- **Podcast Producer Setup Assistant.** This program simplifies the process of setting up Podcast Producer Server. This assistant asks you for the information it needs and takes care of setting up Podcast Producer Server and related services.
- **Podcast Composer.** You can now create workflows effortlessly using this new application (in /Applications/Server/). Podcast Composer graphically leads you through creating workflows that control how Podcast Producer generates and distributes podcasts.
- **Dual video recording.** Podcast Producer 2 lets users record dual video sources using the Podcast Capture application on a Mac or the new Podcast Capture web application on a Mac, iPhone, or Windows computer. Apple provides several picture-in-picture templates, or you can create your own.
- **Podcast Library.** Podcast Library lets your server store podcasts and deliver them to viewers through RSS and Atom feeds. For example, your podcasts can feed directly from your server through iTunes U. Atom feeds simplify distributing multiple podcast versions, such as iPod, Apple TV, and audio only, because each Atom feed can contain multiple versions and the viewer's playback device picks the best version.
- **High availability.** Camera agents and Podcast Capture clients can now fail over to other Computers running Podcast Producer Server. Failover is optional and requires Xsan.

- **Compressor and Qmaster support.** Apple Compressor is a powerful tool that gives you complete control over encoding settings and allows you to create production-quality encoders. Apple Qmaster lets you distribute rendering tasks across a Mac OS X network. Podcast Producer 2 supports Compressor and Qmaster.
- **Podcast Capture Web Application.** This is a web version of Podcast Capture, which you can use to remotely capture and upload audio and video QuickTime movies to a Podcast Producer server for encoding and publishing. You can also use this application to upload documents that are compatible with Apple's Quick Look technology. This application runs from a browser window on Mac, Windows, and UNIX computers.

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 Users use Podcast Capture or Podcast Capture Web Application to record and upload audio or video to a Podcast Producer server for processing.

Podcast Capture is a utility included in Mac OS X v10.6 and Mac OS X Server v10.6 or later. The Podcast Capture Web Application is a web-based application that can run on different platforms.

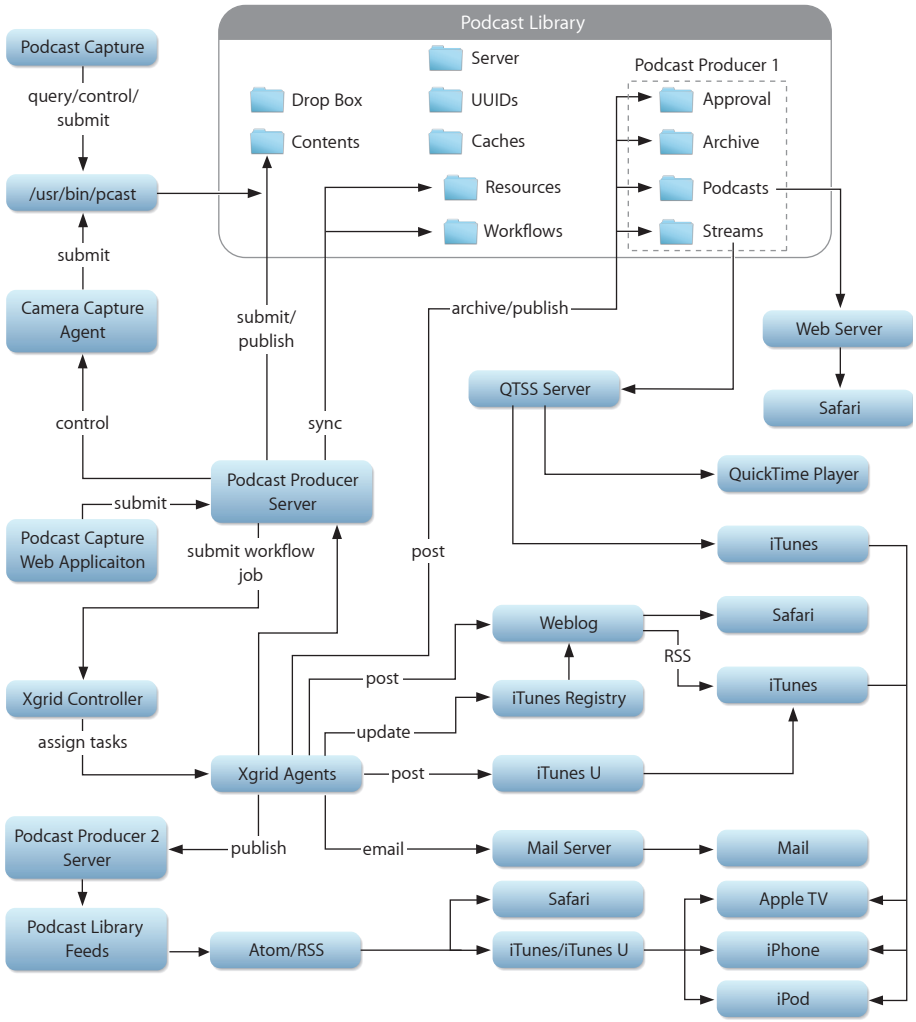
2 If remotely recording audio or video, the recording system submits the recording to the Podcast Producer server.

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 sends the finished podcasts to the Podcast Library for archiving and delivery. 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

The Podcast Producer server (`pcastserverd`) 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 and files compatible with Apple's Quick Look technology to be processed on an Xgrid cluster.

You can use the Podcast Producer server to:

- Specify the location that the Podcast Library uses to store content, resources, and metadata associated with workflow submissions.
- 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 Library

Podcast Library is a repository that stores everything that goes through the Podcast Producer server. This includes workflows, job submissions, original content, intermediate files, all versions of published podcasts, metadata, and any other information needed for the successful processing and publishing of podcasts. Podcast Library uses RSS and Atom feeds to distribute podcasts.

Podcast Library requires a shared file system for storing information and content. The supported shared file systems are Xsan, HFS, and NFS.

For more information about Podcast Library, see Chapter 5, "Podcast Library," on page 40.

Podcast Capture

Podcast Capture (in /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 Mac OS X v10.5 and Mac OS X Server v10.5 or later.

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

Podcast Capture Web Application

Podcast Capture Web Application provides a subset of the functionality of Podcast Capture. The web application allows you to remotely record and submit audio and video recordings to a Podcast Producer server using a web browser. You can also use this application to upload documents that are compatible with Apple's Quick Look technology.

However, you cannot use this application to record screen activity or bind a Mac to the Podcast Producer server.

You can run the Podcast Capture web application from any system running the Mac OS X, Windows, or UNIX operating system, including the iPhone and iPod Touch.

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

Podcast Composer

Podcast Composer (in /Application/Server/) is a graphical workflow editor that leads you through the steps of defining video-based Podcast Producer workflows. You graphically choose the intro, title, and exit videos; specify different transitions and effects between videos; and view real-time titles and effects.

You can add watermarks and overlays to your Podcast content. Your workflow also specifies encoding formats and targets distribution via wiki, iTunes U, or Podcast Library for your finished podcast.

For more information about how to use Podcast Capture, see *Podcast Composer User Guide*.

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 79.

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.

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 content 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.

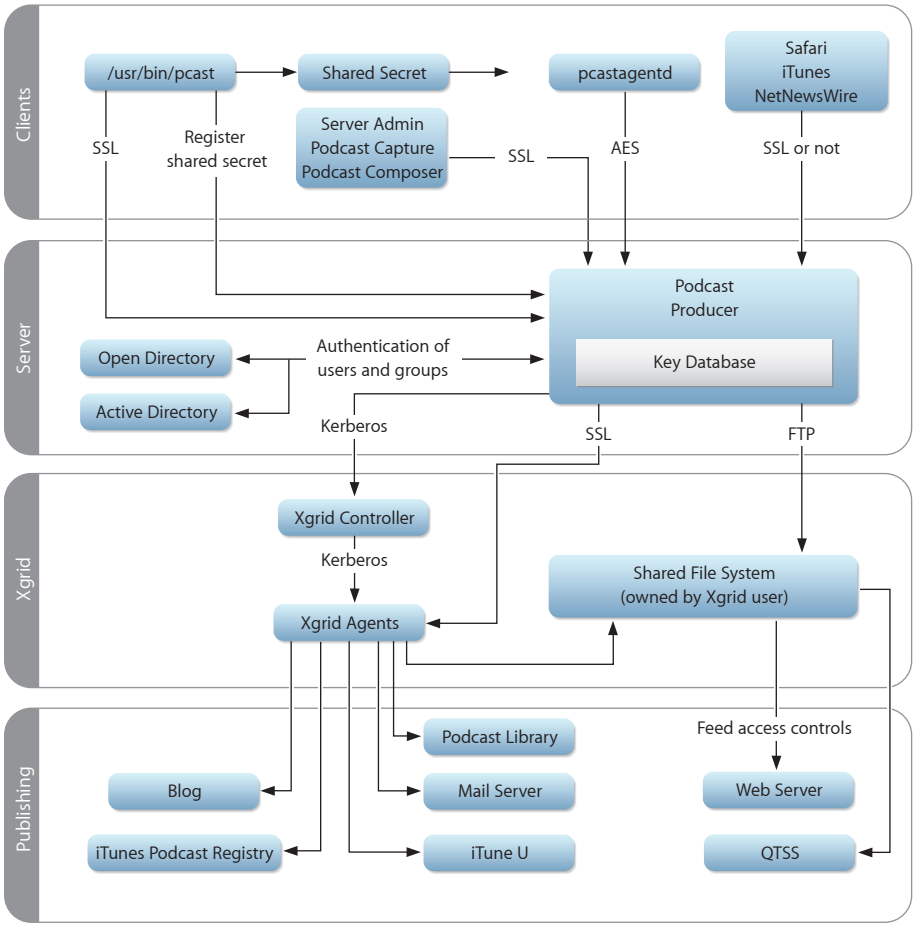
Xgrid

The Podcast Producer server lets you specify an Xgrid controller for processing content movies on Xgrid agents. Podcast Producer 1 workflows can run on Xgrid agents running Mac OS X v10.5. However, Podcast Producer 2 workflows run only on Mac OS X v10.6 Xgrid agents.

Using Xgrid agents for processing content 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 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 support Active Directory digest authentication.

The Podcast Producer server can support any combination of basic, digest, and Kerberos (Negotiate or SPNEGO) authentication.

The authentication settings for Podcast Producer are stored in the file /Library/Preferences/com.apple.pcastserverd.plist by the `http_auth_type` key. You must have root access to edit this file.

By default, the basic, digest, and Kerberos authentication methods are enabled and the property list file contains this entry:

<key>http_auth_type</key> <array> <string>basic</string> <string>digest</string> <string>kerberos</string> </array>

To disable an authentication method, remove the line containing its `<string>` entry and save the file. Then stop and restart the Podcast server, either in Server Admin or by using the command `pcastctl server restart`.

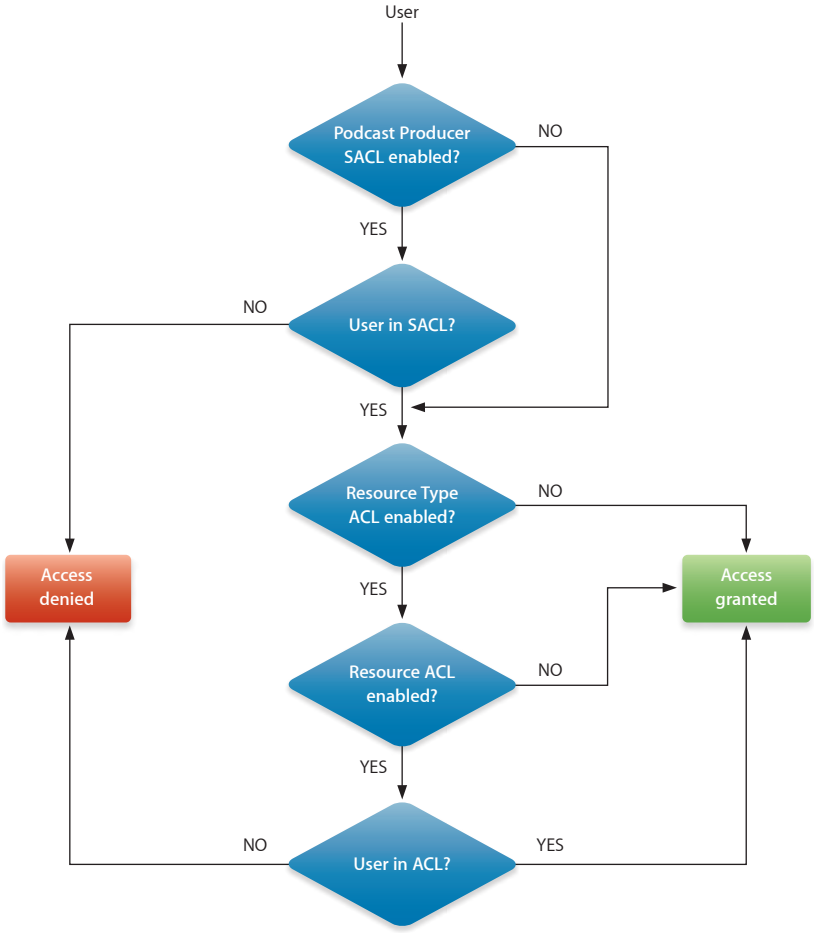
Podcast Producer supports standard HTTP digest authentication for local and Open Directory users and custom digest authentication for Active Directory users.

Access Control

The Podcast Producer server uses Open Directory or Active Directory to authenticate users and groups specified in the Service Access Control List (SACL) in Server Admin and the camera, workflow, and feed Access Control Lists (ACLs) maintained by the Podcast Producer server.

The Podcast Producer server stores the ACLs in the same database it uses to store shared secrets and other sensitive information.

The following diagram illustrates how Podcast Producer uses the Server Admin SACL and the Podcast Producer server ACLs to restrict access to resources: cameras, workflows, and feeds.



Xgrid

The Podcast Producer server uses Kerberos to communicate with the Xgrid controller using the `xgrid` command-line tool.

The Podcast Producer server also uses 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 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 1 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 relevant response.

In the Podcast Producer 2 security model, the Podcast Library, which has valid credentials, publishes podcasts.

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, Compressor, 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.

  You can create professional codecs using Compressor.

  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 v10.6. This tool is used by some default Podcast Producer workflows and is described in “Customizing Workflows” on page 64
- All hosting technology used by Podcast Producer is based on Internet standards like HTTP, Atom, 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)

Setting Up Podcast Producer

2

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 v10.6. For instructions on setting up Mac OS X Server, see *Getting Started* (included on the Mac OS X Server v10.6 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 Chapter 10, "Deploying Scalable Podcast Producer Solutions," on page 67.

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 Capture Requirements

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

- Any Mac running Mac OS X v10.5 or later
- 20 GB of free disk space
- Network connectivity (100 Mbps)
- Some podcast encoding operations require a compatible graphics card
- iSight camera (built-in or external) or FireWire DV camcorder

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 v10.6
- Network connectivity (100 Mbps)
- Xsan for optional cluster file services

Podcast Producer Xgrid Rendering Requirements

Following is a list of the minimum requirements for the Podcast Producer Xgrid system:

- Any Intel-based Macintosh Server or Intel-based desktop Mac (a Mac Pro, for example).
- Mac OS X v10.6 or Mac OS X Server v10.6.
- At least 1 GB of memory (RAM) plus 512 MB of additional RAM per processor core.
- At least 50 GB of available disk space.
- Xsan for optional cluster file services.
- Quartz Extreme-enabled video chipset. (You can verify Quartz Extreme support in the Graphics section of Apple System Profiler.)

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

Setting Up Podcast Producer

To set up the Podcast Producer server, follow these steps:

Step 1: Ensure proper DNS configuration

Before you can configure the Podcast Producer server, be sure that you have access to a properly configured DNS server. The DNS server can be an external server or a server you set up locally on your server.

For more information about ensuring proper DNS configuration, see "Ensuring Proper DNS Configuration" on page 25.

Step 2: Enable Podcast Producer server
Before you can configure and start the Podcast Producer server, you must enable Podcast Producer Server administration in Server Admin. This allows Server Admin to start, stop, and change settings for Podcast Producer Server.

For more information about enabling Podcast Producer server, see "Enabling Podcast Producer Server Administration in Server Admin" on page 27.

Step 3: Configure Podcast Producer server
Use the Podcast Producer Setup Assistant to configure the Podcast Producer server and other required services.

For more information about configuring Podcast Producer server, see "Configuring Podcast Producer Server Using the Podcast Producer Setup Assistant" on page 27.

Step 4: Configure default workflow properties
Use the Server Admin to configure the default workflow properties of Podcast Producer.

For more information about configuring the default workflow properties of Podcast Producer server, see "Configuring the Default Workflow Settings" on page 30.

Ensuring Proper DNS Configuration

If you don't have access to a DNS server on your network, then set up one locally on your server as described in this section. However, if you're using an existing DNS server, then make sure that forward and reverse DNS is working for your server. If not, contact your network administrator.

Important: To set up a DNS server locally, your computer must be connected to a network router or switch.

- "Configuring the DNS Server" on page 25
- "Verifying Forward and Reverse DNS Operation" on page 26

Configuring the DNS Server

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

To configure the DNS server:

1. Open Server Admin and connect to the server.
2. Click Settings.
3. Click Services.
4. Select the DNS checkbox.
5. Click Save.

6. Click the triangle at the left of the server.

   The list of services appears.

7. From the expanded Servers list, select DNS.
8. Click Zones.
9. Click Add Zone > Add Primary Zone (Master).
10. Click the new zone (its name is example.com.).
11. In the Primary Zone Name field, replace example.com with "*hostname*.private."

    Replace *hostname* with the name of your computer.

12. In the Admin Email field, enter "*username*@*hostname*.private."

    Replace *username* with the name of your administrator account.

13. Make sure that there is a reverse zone that resolves the IP address of your computer to "*hostname*.private."
14. Click Save.
15. Start the service by clicking Start DNS.

    If the DNS server is running, click Stop DNS first.

16. To make sure that DNS is running, ping server.pcast.private and client.pcast.private using Network Utility or from the command line.

Verifying Forward and Reverse DNS Operation

To ensure proper DNS setup, you must verify that forward and reverse DNS is working for your server.

To verify proper DNS configuration:

1. Open Terminal.
2. To test whether forward and reverse DNS is working, enter the following command:

```
sudo changeip -checkhostname
```

3. Verify that the output indicates success and looks like the following example:

```
Primary address     = 10.0.0.80

Current HostName    = meichen.example.com
DNS HostName        = meichen.example.com

The names match. There is nothing to change.
dirserv:success = "success"
```

If this test fails, do the following and try this test again:

- If you're using your own local DNS server, revisit its configuration as described in "Configuring the DNS Server" on page 25.

- If you're using a DNS server on your network, ask the network administrator to fix the problem.

Enabling Podcast Producer Server Administration in Server Admin

Before you can configure and start the Podcast Producer server, you must enable Podcast Producer Server administration in Server Admin. This allows Server Admin to start, stop, and change settings for Podcast Producer Server.

To enable Podcast Producer Server for administration:

1. Open Server Admin and connect to the server.
2. Click the Settings button in the toolbar.
3. Click the Services tab.
4. Select the checkbox for Podcast Producer Server.
5. Click Save.

Configuring Podcast Producer Server Using the Podcast Producer Setup Assistant

Podcast Producer Server provides the Podcast Producer Setup Assistant, which leads you through configuring Podcast Producer Server and all other services that are needed by it.

To configure Podcast Producer Server using the Podcast Producer Setup Assistant:

1. Open Server Admin.
2. In the Servers list, select the server that will run Podcast Producer Server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer Server.
4. Click the Overview button in the toolbar.
5. Click Configure Podcast Producer.
6. In the Introduction screen, click Continue.
7. In the Express or Standard screen, select a setup method and then click Continue:
   - Express setup—To automatically configure and enable all required network services locally on the same server as Podcast Producer, select this option. Then proceed to "Using the Express Setup Configuration Option" on page 27to complete the configuration process.
   - Standard setup—To manually specify the network services required by Podcast Producer Server, select this option. Then proceed to "Using the Standard Setup Configuration Option" on page 28to complete the configuration process.

Using the Express Setup Configuration Option

Use the express setup option to quickly get Podcast Producer Server up and running.

To perform an express setup of Podcast Producer Server:

1. In the Directory screen:
   - If your server is hosting an Open Directory master, click Continue.
   - If your server is bound to a directory and has been Kerberized, click Continue.
   - If your server is not hosting an Open Directory master and is not bound to a directory, set up an Open Directory master:

     a. In the Name field, enter the name of the directory administrator.

     b. In the Short Name field, enter the short name of the directory administrator.

     c. In the User ID field, enter a unique ID if you don't want to use the default ID.

     d. In the Password and Verify fields, enter the password for the directory administrator.

     e. Click Continue.
   - If your server is bound to a directory but has not been Kerberized, provide the directory administrator name and password to Kerberize your server:

     a. In the Name field, enter the name of the directory administrator.

     b. In the Password field, enter the password for the directory administrator.

     The password should be the same as the password you created when you installed Mac OS X Server v10.6.

     c. Click Continue.
2. In the Confirm screen, read the summary of the settings that the setup assistant will configure, then click Continue.
3. If prompted, enter the name and password of your Open Directory administrator account, then click Continue.

   If the express setup succeeds, the setup assistant displays a message informing you that Podcast Producer Server is configured and ready for use. Otherwise, an error message appears describing the problem.
4. If setup succeeds, click OK to dismiss the message.
5. If setup fails, record the error, click OK, fix the problem, and try again.
6. To start Podcast Producer Server, click Start Podcast Producer.

Using the Standard Setup Configuration Option

Use the standard setup option to specify the network services that Podcast Producer Server requires to run.

To perform a standard setup of Podcast Producer Server:

1. In the Directory screen:
   - If your server is hosting an Open Directory master, click Continue.

- If your server is bound to a directory and has been Kerberized, click Continue.
- If your server is not hosting an Open Directory master and is not bound to a directory, set up an Open Directory master:

  a. In the Name field, enter the name of the directory administrator.

  b. In the Short Name field, enter the short name of the directory administrator.

  c. In the User ID field, enter a unique ID if you don't want to use the default ID.

  d. In the Password and Verify fields, enter the password for the directory administrator.

  e. Click Continue.
- If your server is bound to a directory but has not been Kerberized, provide the directory administrator name and password to kerberize your server:

  a. In the Name field, enter the name of the directory administrator.

  b. In the Password field, enter the password for the directory administrator.

  The password should be the same as the password you created when you installed Mac OS X Server v10.6.

  c. Click Continue.

2 In the Accounts screen, select one of the following options, then click Continue:

- Create workflow execution user automatically—Select to automatically create a user account that Podcast Producer Server uses to submit jobs to Xgrid, select this option.
- Enter credentials for an existing user—Select to specify an existing user account for Podcast Producer Server to use to submit jobs to Xgrid, select this option, then enter the user credentials in the Name and Password fields.

3 In the Library screen:

a Click Choose to specify the path to the Podcast Library or enter the path manually in the Podcast Library field.

Podcast Producer uses this location in the filesystem to store configuration information, recordings, and podcasts. This location must be in a shared file system if you want to use additional computers as rendering agents. Otherwise, only your computer can process Podcast Producer workflows.

If you plan to configure your server for failover, you must use Xsan for the shared file system.

b To share the Podcast Library using NFS, select "Enable NFS share."

The recommended shared file system is Xsan. However, if the location you have chosen is not on an Xsan volume, or if not all of your computers are attached to the Xsan volume, you can choose to share the Podcast Library using NFS.

c Click Continue.

4 In the Xgrid screen:

a To set up a local Xgrid controller and agent on your system, select "Set up a local Xgrid controller and agent."

b To use an existing Xgrid, controller, select "Use an existing Xgrid controller", and then enter the address of the controller and choose one from the list.

c Click Continue.

5 In the Confirm screen, read the summary of the settings that the setup assistant will configure, then click Continue.

6 If prompted, enter the name and password of your Open Directory administrator account, then click Continue.

If the standard setup succeeds, the setup assistant displays a message informing you that Podcast Producer Server is configured and ready for use. Otherwise, an error message appears describing the problem.

7 If setup succeeds, click OK to dismiss the message.

8 If setup fails, record the error, click OK, fix the problem, and try again.

9 To start Podcast Producer Server, click Start Podcast Producer.

10 Using Workgroup Manager, create users and groups if needed.

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 Servers list, select the server running Podcast Producer, then click the service disclosure triangle to show the services for administration.

3 In the service list beneath the server, select Podcast Producer Server.

4 Click Settings.

5 Click Properties.

6 Click the triangle next to Default Properties to display the properties.

7 If needed, change the default values of these properties.

Property	Value
Administrator Short Name	The default value is the short name of the administrator account you created when setting up your Mac OS X Server v10.6.
Copyright	This field is empty by default. Enter the copyright information you want to add to all workflow.
Drop Box Folder	This folder is used by Qmaster when processing Compressor encodings. By default, the DropBox folder is at the root level of the shared filesystem used by Podcast Library.
Drop Box Owner Group Shortname	The default value is the short name of the administrator account you created when setting up your Mac OS X Server v10.6.
Library Language	English is the default language used for the Podcast Library feed catalogs.
Notification Language	English is the default language used for sending notifications.
Organization	This field is empty by default. Enter the name of the organization you want added to podcasts.
Postflight Script Path	The preflight script is the first task that a workflow runs.
Preflight Script Path	The preflight script is the first task that a workflow runs.

If you have upgraded to Podcast Producer 2, the Default Properties list includes all default Podcast Producer 1 and Podcast Producer 2 default properties.

For more information about configuring default properties, see "Configuring Default Workflow Properties" on page 45.

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 v10.6 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.

Accessing the Podcast Capture Web Application

The Podcast Capture web application lets you use a web browser to remotely record and submit content to a Podcast Producer server for processing and publishing.

With the Podcast Capture web application, you can:

- Record and submit video from single and dual sources
- Record and submit audio
- Submit files compatible with Apple's Quick Look technology

To access the Podcast Capture web application:

1. In Server Admin, enable the Podcast Capture web application:
   a. Open Server Admin.
   b. In the Servers list, select the server running Podcast Producer, then click the service disclosure triangle to show the services for administration.
   c. In the service list beneath the server, select Podcast Producer Server.
   d. Click the Settings button in the toolbar, and then click the General tab.
   e. Select "Enable Podcast Capture web application."
   f. Click Save.
2. Open a web browser on your system.

   For example, open Safari on your Mac, iPhone, or Windows computer.
3. In the Address field, enter the following URL:

   https://*server*:8170/podcastproducer/capture

   Replace *server* with the IP address or DNS name of your server (for example, pcastserver.example.com).

   You can also launch the Podcast Capture web application by clicking its URL in the Overview pane of the Podcast Producer server in Server Admin (Server Admin > *server* > Podcast Producer > Overview).
4. Enter the server address or hostname and your user credentials to log in:
   a. In the address field, enter the DNS name or IP address of your server.
   b. In the Name and Password fields, enter valid user credentials.

      By default, you can use the administrator account you created when you configured your server for the first time.
5. Click Connect.

   To learn how to use the Podcast Capture web application, refer to its onscreen help.

Upgrading to Podcast Producer 2

3

Use this chapter to learn how to upgrade your Podcast Producer server to Podcast Producer 2.

The process of upgrading your Podcast Producer 1 server running on a computer with Mac OS X Server v10.5 to Podcast Producer 2 is fairly simple if you follow the steps in this chapter. Most of the work involved in upgrading the server and migrating the data to the Podcast Library is done automatically.

Upgrading Your Podcast Producer Server

During the upgrade process, Podcast Producer clients cannot submit jobs. Plan to perform the upgrade process when there is little or no Podcast Producer activity to minimize the disruption.

To upgrade your Podcast Producer server to Podcast Producer 2:

1. Update your computer to run the latest version of Mac OS X Server v10.5 (version 10.5.7 or later).
2. If you're using Xsan to store the shared filesystem, upgrade the computer running the Xsan metadata controller to Mac OS X Server v10.6.
3. Upgrade your computer by installing Mac OS X Server v10.6.
4. In Server Admin, check the status indicator of the Podcast Producer server.

   A yellow indicator means that the Podcast Producer data is still being migrated and will be available soon. It can also mean that there is a problem that you must fix before Podcast Producer 2 can run.

   If the indicator is green, it means that the upgrade and migration process was successful.

Troubleshooting Upgrade and Migration Issues

This section describes how to troubleshoot Podcast Producer upgrading and migration issues.

- "Adding a Service Principal" on page 35
- "Checking the Startup Log" on page 35

Adding a Service Principal

If a server is Kerberized with versions 10.5.0 to 10.5.5 of Mac OS X Server and is not updated to v10.5.7 or later, the system will not have a pcast service principal and the status indicator of the Podcast Producer server in Server Admin turns yellow.

To fix the problem, you must add the service principal.

To add the service principal:

1. Open Server Admin.
2. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Open Directory.
4. Click the Settings button in the toolbar, then the General tab.
5. Click Join Kerberos.

From the command line:

```
sso_util configure -r <REALM> -f <node> -a <diradmin> pcast
```

Checking the Startup Log

To check whether there was a problem during the upgrade process, check the Podcast Producer Server Startup log of the Podcast Producer server. This log is stored at /Library/Logs/pcastserverd/startup.log.

To check the Startup log:

1. In the service list beneath the server, select Podcast Producer.
2. Click the Logs button in the toolbar.
3. From the View pop-up menu, choose Podcast Producer Server Startup Log.
4. Check the log for problems.

Setting Up Podcast Producer for High Availability

4

Use this chapter to learn how to set up the Podcast Producer server for high availability.

Podcast Producer 2 supports failover of the Podcast Producer server process, which allows Podcast Producer clients to fail over to other Podcast Producer servers.

Podcast Producer 2 also supports failover of the Xgrid controller to ensure that Xgrid processing of workflows is not disrupted.

How Podcast Producer Failover Works

For Podcast Producer failover to work, you must configure the primary Podcast Producer server and the failover servers to be on the same authentication realm and configured to use the same Podcast Library location in the shared file system, which must reside on an Xsan volume.

When a Podcast Producer client (camera agent, Podcast Capture, or Podcast Capture Web Application) logs in to a Podcast Producer server, it receives a list of the IP addresses of the Podcast Producer servers it can fail over to.

This list is cached and keyed using the name of the server the client was connecting to. If the client attempts to connect to that server later and it is unavailable, the client looks in the cached list and attempts to connect to each server in turn until.

The client keeps trying until it succeeds in connecting to one of the server on the list. It will then cache the new list of failover hosts based on the response it gets from the new server.

Failover is not supported for web clients, such as Safari accessing the Podcast Library or Podcast Capture Web Application.

In addition, load balancing is not supported for Podcast Producer clients such as `podcast` and Podcast Capture or the camera agents. However, it is possible to use a hardware load balancer for web clients; such a hardware load balancer would also provide failover support.

If the primary Podcast Producer server or Xgrid controller fails, its clients fail over to one of the servers on the list. Xgrid supports transparent failover for clients and agents; Podcast Producer 2 is one of those clients that are supported.

The Xgrid configuration is stored in the shared file system. If you set up a secondary system and point it at the primary system's shared file system location, it uses the same Xgrid controller as the primary system.

Before Setting Up High Availability

Before you configure your Podcast Producer server for failover, note the following:

- The failover server must be bound or Kerberized into the same authentication realm as the primary Podcast Producer server.

  The failover server must not already be Kerberized into some other realm.
- The shared filesystem used by your server's Podcast Library must reside in an Xsan volume.
- The Xsan metadata controller must run on a computer with Mac OS X Server v10.6.

  For more information about setting up Xsan volumes, refer to *Xsan 2 Administrator's Guide* and related documentation.
- For best results, start with a fresh installation of Mac OS X Server v10.6 on the computer you intend to use as a failover server.
- Don't use the Podcast Producer Setup Assistant to configure the failover servers.

Configuring Podcast Producer for Failover

You can set up one or more computers as failover Podcast Producer servers.

Important: All clients (Podcast Capture, Podcast Capture Web Application, and camera agents) must connect successfully to the primary server at least once before they will be able to fail over.

To configure your Podcast Producer server for failover:

1. Install Mac OS X Server v10.6 on the computers you intend to use as failover servers.
2. Bind the failover servers into the same authentication realm as the primary Podcast Producer server.
3. In Server Admin, enable the Podcast Producer server:
   a. Connect to the server.
   b. Click Settings and then Services.
   c. Select the checkbox for Podcast Producer Server.
   d. Click Save.

4 In Server Admin, configure the Podcast Producer server:

   a Select the server, then click the service disclosure triangle to show the services for administration.

   b In the service list beneath the server, select Podcast Producer.

   c Click the Settings button in the toolbar, then click the General tab.

   d In the Podcast Library field, enter the same path used by the primary Podcast Producer server.

   e Click Save.

   The Podcast Producer server uses the information in the Podcast Library of the primary server to configure the other settings needed for failover, such as Xgrid settings.

5 If you're using an SSL certificate on the primary server, in Server Admin, configure the failover server to use its own certificate:

   a Select the server, then click the service disclosure triangle to show the services for administration.

   b In the service list beneath the server, select Podcast Producer.

   c Click the Settings button in the toolbar, then click the General tab.

   d From the Certificate pop-up menu, choose a certificate or create one.

   e Click Save.

6 To verify that your Podcast Producer is setup for failover, run the following command on the primary and failover servers:

```
podcast --listinfo --server <hostname>
```

   This command returns a property list file containing a list (under the `cluster_members` key) of the Podcast Producer servers that a client can fail over to.

Troubleshooting High Availability Issues

This section describes common troubleshooting issues and their solution.

Failover After First Login

All clients (Podcast Capture, Podcast Capture Web Application, and camera agents) must connect successfully to the primary server at least once before they will be able to fail over.

Xsan Redundancy

Use standard Xsan techniques to provide failover protection. Set up multiple Xgrid metadata controllers for failover.

Failover and Express Setup

Consider the following configuration of a Podcast Producer server:

- The server is set up using the Express setup option of the Podcast Server Assistant.
- The server is bound to a different authentication realm than the realm used by the primary Podcast Producer server. (Express setup always creates a standalone Open Directory master and uses it to Kerberize the server.)

To use this server as a failover Podcast Producer server, you must do the following:

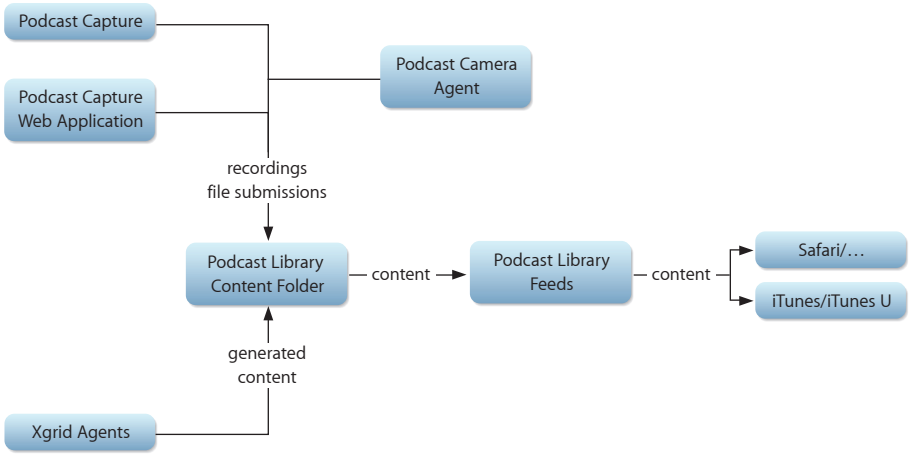
1. Stop the Podcast Producer server.
2. Destroy the Open Directory master on the server.
3. Bind the server to the same directory as the one used by the primary server.
4. Change the Podcast Library location to point to the Podcast Library location used by the primary server.
5. Start the Podcast Producer server.

Podcast Library

5

Use this chapter to learn how the Podcast Library works.

The Podcast Library is central to Podcast Producer. It is responsible for storing, organizing, and serving content to users.



Overview of Podcast Library

The Podcast Library fulfills the following functions:

- Stores information and content

  Podcast Library is a repository that stores everything that goes through the Podcast Producer server. This includes, workflows, job submissions, original content, intermediate files, all versions of published podcasts, metadata, and other information needed for the processing and publishing of podcasts.

- Organizes information and content

Podcast Library organizes information based on the metadata available to it. For example, Podcast Library uses information such as the camera used to record a video, the name of the user who submitted the video, the time of day the recording was submitted, the type of recording, and the name of the workflow that was processed to create folders or categories for organizing videos.

- Serves content

  Podcast Library serves content to users over RSS and Atom feeds. For example, your podcasts can be hosted on your server but accessed through the iTunes U, which can aggregate the feeds, but leaves the content on the server.

  Atom feeds simplify the distribution of multiple podcast versions, such as iPod, Apple TV, and audio only, because each Atom feed can contain multiple versions and the viewer's playback device picks the best version.

The Podcast Library consists of the following:

- Logic, embedded in pcastserverd, for storing, organizing, and serving content
- Shared file system where information and content is stored

Shared File System

Podcast Library requires a shared file system for storing information and content. The supported shared file systems are Xsan, HFS, and NFS.

Important: Don't modify files in the shared file system directly. Instead, use the `podcast` command-line tool. For example, to add a workflow, use the `podcast --installworkflow` command. To delete a workflow, use the `podcast --deleteworkflow`. For more information about `podcast` see its man page.

Podcast Library stores the following in its shared filesystem:

- Approval, Archive, Podcasts, and Streams folders
  - The Approval folder stores content that requires approval.
  - The Archive folder stores copies of the submitted or processed recordings for archival purposes.
  - The Podcast folder is where Podcast Producer stores the finished podcasts.
  - The Streams folder stores the QTSS streams produced by Podcast Producer.

  These folders exist to support Podcast Producer 1 workflows. None of the Podcast Producer 2 workflows use these folders.
- AgentPreviews

  For every remote camera, this folder stores the last preview frame received.
- Caches

This folder contains the workflow and resource caches used for processing workflows. This folder also contains caches of the web pages that the Podcast Library has published so that the next time a feed is requested, Podcast Library spends less time processing the request.

- Content

  Stores the submitted and published content.

- DropBox

  This folder allows Podcast Producer to share files with Qmaster when processing Compressor encodings.

  To protect content, Podcast Producer doesn't give Qmaster the filesystem permissions needed to look into the Content folder. Instead, Podcast Producer moves the files it needs to hand off to Qmaster into the DropBox folder.

  Podcast Producer and Qmaster have the needed permissions to access this folder. When Qmaster finishes processing the Compressor encodings, it saves the output into the DropBox folder. Podcast Producer moves the processed files from this folder into the Published folder of the Content folder.

- pcastserverd.plist

  This property list file defines the identity of the Podcast Library.

- Server

This folder stores the following:

  - The jobs that will be submitted to Xgrid for processing
  - All resources installed on the server
  - The workflows deployed by the Podcast Producer server
  - Cluster preferences (for example, the Podcast Producer Xgrid settings defined in Server Admin)
  - Member preferences for each member of the Xgrid cluster

- UUIDs

  This folder is an index of the Content folder. It stores aliases to published podcasts.

Managing Workflows

6

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. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Workflows.
5. Select a workflow in the Workflow list.
6. To restrict access to the workflow, click "Allow access to *workflow name* for the following users and groups."
7. 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 (-).
8. 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. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. 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. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Workflows.
5. 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.
6. 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.
7. 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. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Workflows.
5. Select a workflow in the Workflow list.
6. 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 45
- "Configuring Custom Workflow Properties" on page 46

Configuring Default Workflow Properties

Podcast Producer defines a set of default properties that are common to most workflows.

You can use Server Admin to configure a the following Podcast Producer default workflow properties.

Property	Value
Administrator Short Name	The default value is the short name of the administrator account you created when setting up your Mac OS X Server v10.6.
Copyright	This field is empty by default. Enter the copyright information you want to add to all workflow.
Drop Box Folder	This folder is used by Qmaster when processing Compressor encodings. By default, the DropBox folder is at the root level of the shared filesystem used by Podcast Library.

Property	Value
Drop Box Owner Group Shortname	The default value is the short name of the administrator account you created when setting up your Mac OS X Server v10.6.
Library Language	English is the default language used for the Podcast Library feed catalogs.
Notification Language	English is the default language used for sending notifications.
Organization	This field is empty by default. Enter the name of the organization you want added to podcasts.
Postflight Script Path	The preflight script is the first task that a workflow runs.
Preflight Script Path	The preflight script is the first task that a workflow runs.

To configure default workflow properties:

1. Open Server Admin.
2. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Settings.
5. Click Properties.
6. Click the triangle next to Default Properties to display the properties.
7. To change the value of a property, double-click the property's value and enter a new value.
8. 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" on page 46
- "Deleting Custom Workflow Properties" on page 47
- "Modifying Custom Workflow Properties" on page 47

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 Select the server, then click the service disclosure triangle to show the services for administration.
3 In the service list beneath the server, select Podcast Producer.
4 Click Settings.
5 Click Properties.
6 Click the triangle next to Custom Properties to display the custom properties.
7 Click the (+) button to add a custom property.
8 Double-click the name field of the property and enter its name.
9 Double-click the value field of the property and enter its value.
10 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 Select the server, then click the service disclosure triangle to show the services for administration.
3 In the service list beneath the server, select Podcast Producer.
4 Click Settings.
5 Click Properties.
6 Click the triangle next to Custom Properties to display the custom properties.
7 Select the property to delete and click the (-) button.
8 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 Select the server, then click the service disclosure triangle to show the services for administration.
3 In the service list beneath the server, select Podcast Producer.
4 Click Settings.
5 Click Properties.
6 Click the triangle next to Custom Properties to display the custom properties.
7 To change the value of a property, double-click the property's value and enter a value.

8 (Optional) To encrypt a property, click the property's checkbox.

9 Click Save.

Managing Cameras

7

This chapter describes how to manage and monitor camera usage.

Managing Cameras

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

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. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Cameras.
5. Select a camera in the Cameras list.
6. To restrict access to the camera, click "Allow access to *camera name* for the following users and groups."
7. 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 (-).

8 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 Select the server, then click the service disclosure triangle to show the services for administration.

3 In the service list beneath the server, select Podcast Producer.

4 Click Cameras.

5 Select a camera in the Cameras list.

6 Click Remove Camera.

7 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 Select the server, then click the service disclosure triangle to show the services for administration.

3 In the service list beneath the server, select Podcast Producer.

4 Click Cameras.

5 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.

6 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.

7 To list cameras based on their status, from the Search pop-up menu, choose Status and enter the name of the status.

Managing Feeds

8

Use this chapter to learn how to manage Podcast Library feeds.

Podcast Producer 2 serves content using RSS and Atom feeds. The use of feeds greatly simplifies the delivery process and conforms to industry standards.

Accessing Podcast Library Feeds

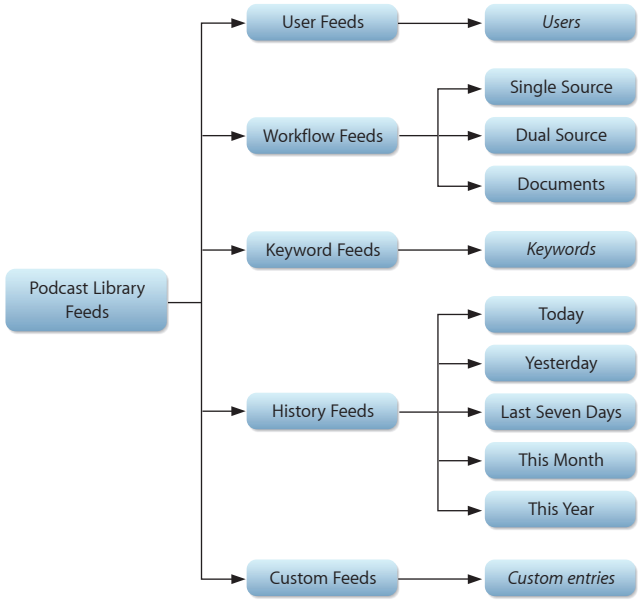
Podcast Producer server provides a URL link in its Overview pane in Server Admin that you can click to access the Podcast Library feeds.

By default, the URL is:

feed://server_address:8171/podcastproducer/catalogs

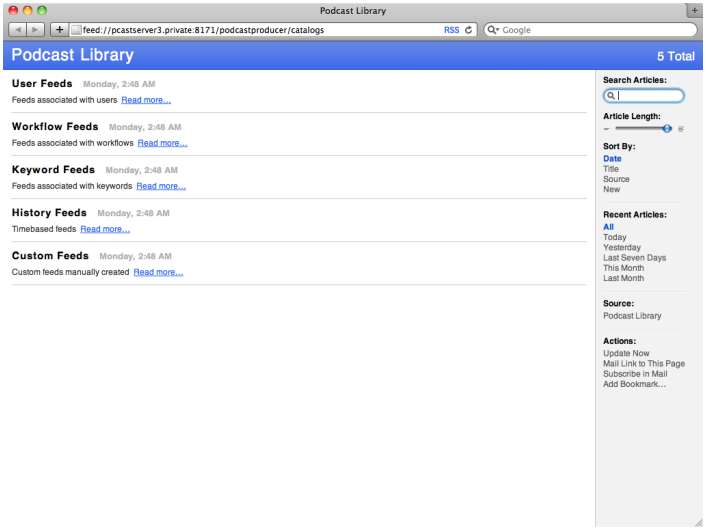
Podcast Library Feed Structure

The following diagram illustrates the hierarchy of the Podcast Library feeds.



Each box in the diagram represents a web page, except the Users, Keywords, and "Custom entries" boxes. Each one of these boxes represents one or more pages. By default, these pages are empty after you set up Podcast Producer server.

The main page, Podcast Library, is a super catalog of all feeds. It organizes feeds into five catalogs.



These catalogs are:

- User Feeds

  To open this catalog, click User Feeds. This is a catalog of all podcast episodes organized by submitting user. Each entry in this catalog represents a user and is a link to another page that lists the podcast episodes submitted by the user. Podcast Library adds new user entries after workflows are submitted to the Podcast Producer server for processing.

- Workflow Feeds

  To open this catalog, click Workflow Feeds. Each entry in this catalog represents a workflow and is a link to another page that lists the podcast episodes submitted by the workflow. Podcast Library adds new workflow entries after workflows are installed or deployed on the Podcast Producer server.

- Keyword Feeds

  To open this catalog, click Keyword Feeds. Each entry in this catalog represents a keyword associated and is a link to another page that lists the podcast episodes associated with the keyword. Podcast Library adds new keyword entries after workflows with keywords defined in them are installed or deployed on the Podcast Producer server.

- History Feeds

To open this catalog, click History Feeds. This catalog organizes podcast episodes based on when they were submitted. This catalog has five entries:

- Today

  Click this entry to see all podcast episodes published today.

- Yesterday

  Click this entry to see all podcast episodes published yesterday.

- Last Seven Days

  Click this entry to see all podcast episodes published in the last seven days.

- This Month

  Click this entry to see all podcast episodes published this month.

- This Year

  Click this entry to see all podcast episodes published this year.

  Podcast Library adds these entries after you start the Podcast Producer server.

- Custom Feeds

  To open this catalog, click Custom Feeds. Each entry in this catalog represents a custom grouping of podcast episodes.

You can use the toolbar at the right-hand side to search, sort, and manage feeds.

Controlling Access to Feeds

By default, anyone can access the Podcast Library feeds.

However, you can restrict access to feeds by enabling the Podcast Producer server SACL in server admin and specifying which users and groups have access.

You can also add another access control layer by enabling an ACL for feeds and specifying which users and groups have access. to control access to feeds using ACL, use the ACL administration commands of the `podcast` command.

For example, to enable ACL for all feeds, enter:

```
podcast --enableacls --resource_type Feed
```

To enable ACL for a feed, enter:

```
podcast --enableacl --resource_uuid UUID
```

For more information about ACL administration commands, see the man page of `podcast`.

Using Feed and Catalog Administration Commands

To manage catalogs and feeds, use the feed and catalog administration commands of `podcast`.

For more information about these commands, see the man page of `podcast`.

Customizing Workflows

9

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

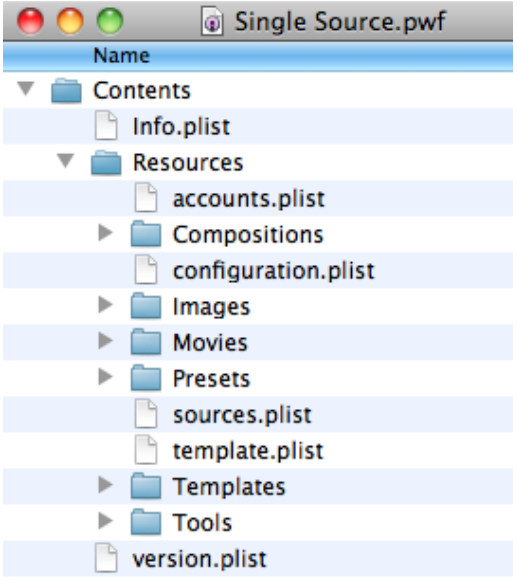
The workflows that ship with Mac OS X Server v10.6 are examples of the type of back-end processing you can perform on input files.

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.

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.)
Resources	(Optional) Contains all local resources used by the workflow. Global workflow resources are stored in /System/Library/PodcastProducer/Resources.

The following table describes the default contents of the Resources folder:

Element	Description
Compositions	(Optional) Stores the Quartz compositions referenced by the workflow.
Images	(Optional) Contains images used by the workflow.
Movies	(Optional) Contains movies used by the workflow.
Presets	(Optional) Contains property lists that specify the encoding presets used by the workflow.
sources.plist	(Optional) Specifies the type of content processed by the workflow.
template.plist	The workflow template that specifies the tasks to execute and the order of execution.
Templates	(Optional) 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	(Optional) Contains command-line tools used by the workflow.
version.plist	Contains workflow version information.

In addition to the default contents of the Resources folder, Podcast Composer uses additional property lists: accounts.plist, crypto.plist, and configuration.plist.

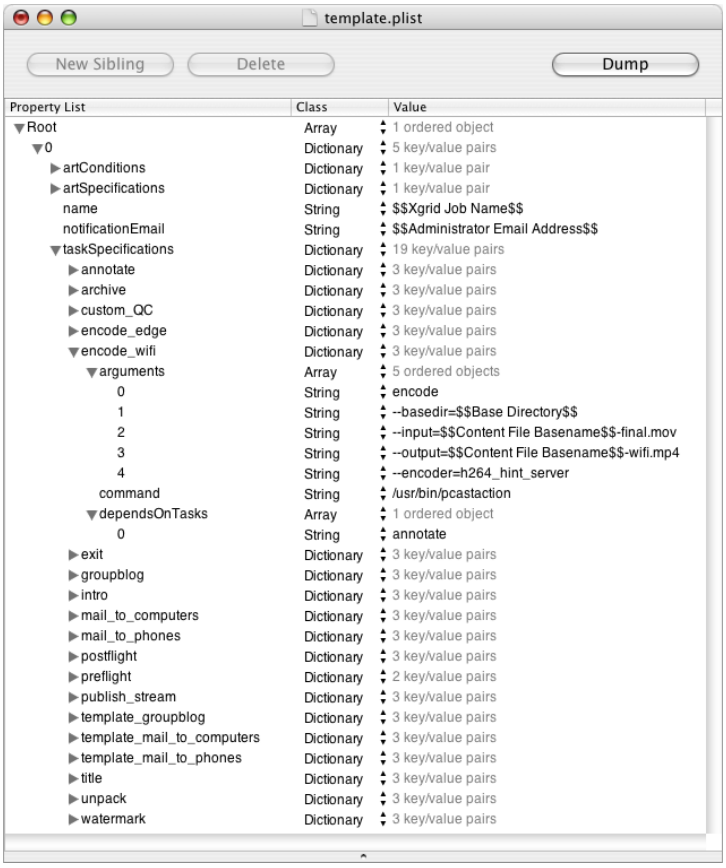
The Structure of a Workflow

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 *Command-Line Administration* guide or the `xgrid` 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 ($$Xgrid Job Name$$) 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 ($$Administrator Email Address$$) 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 taskSpecificiations 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 18.

Server-Generated Property Keys

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

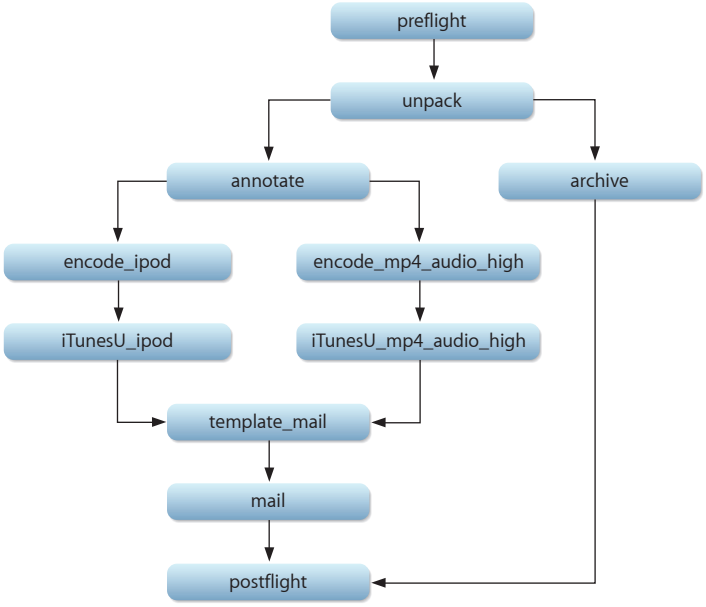
Property Key	Matching Property Name
$$Administrator Email Address$$	The mail address of the Podcast Producer administrator.
$$Administrator Full Name$$	The full name of the Podcast Producer administrator.
$$Base Directory$$	The base directory that Podcast Producer creates for storing submitted recordings (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.
$$Library Bundle Path$$	The path to the root level of the Podcast Library on the shared file system .
$$Podcast Producer URL$$	The URL of the Podcast Producer server.
$$Podcast Producer Wiki Content URL$$	The URL of the wiki.
$$Podcast Producer Wiki Content Document Root$$	The path to the wiki's root level folder.
$$Recording UUID$$	The UUID of the recording.
$$Server UUID$$	The UUID of the Podcast Producer server.
$$Shared Filesystem$$	The top level of the shared file system.
$$User Email Address$$	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.

Property Key	Matching Property Name
$$User ID$$	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.
$$Active Podcast Producer Cluster Members$$	The Podcast Producer Xgrid cluster members.

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.

For a listing of the `pcastaction` commands, see "The pcastaction Tool" on page 80.

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.

Podcast Producer Default Workflows

Podcast Producer ships with three sample workflows. These workflows were created using Podcast Composer.

To modify these workflows, you can use Podcast Composer or you can modify them manually. However, if you modify them manually, you won't be able to edit them again using Podcast Composer.

To explore and modify these workflows, open them using Podcast Composer.

To open a default workflow using Podcast Composer:

1. Open Podcast Composer.
2. Choose File > Open Remote.
3. In the dialog sheet that appears:
   a. In the Server field, enter the address of your Podcast Producer server.
   b. In the Name and Password fields, enter the user name and password assigned to you by the Podcast Producer administrator.
   c. Click Connect.
4. In the Open Remote Workflow window:
   a. Select one of the default workflows.
   b. Click Open.
   c. (Optional) In the Save As field enter a different name.
   d. Select a folder in which to store the workflow.
   e. Click Save.

5 Explore and modify the workflow as needed.

6 Save the workflow (File > Save).

7 To Deploy the workflow to the server, choose File > Deploy to Server.

8 In the dialog sheet that appears:

   a In the Server field, enter the address of your Podcast Producer server.

   b In the Name and Password fields, enter the user name and password assigned to you by the Podcast Producer administrator.

   c Click Deploy.

9 If prompted, click Overwrite to replace the existing workflow or Duplicate to create a new workflow.

   Every workflow has a unique identifier (UUID). Even if you change the name of the workflow in Podcast Composer, the workflow retains its UUID, unless you instruct Podcast Composer to deploy a new workflow with a new UUID.

10 When prompted, click OK.

Single Source

This workflow:

- Takes as input one QuickTime movie (video recording, audio-only recording, or screen recording)
- Adds introduction, title, and exit movies to the recording
- Adds a watermark to the recording
- Generates 3 versions of the podcast (iPod/iPhone, Apple TV (SD), and Audio)
- Publishes the podcast versions to the Podcast Library

Dual Source

This workflow:

- Takes as input two QuickTime movies (two video recordings or a video recording and a screen recording)
- Combines the two movies using the Overlay Quartz composition
- Adds introduction, title, and exit movies to the combined movie
- Adds a watermark to the combined movie
- Generates 3 versions of the podcast (iPod/iPhone, Apple TV (SD), and Audio)
- Publishes the podcast versions to the Podcast Library

Documents

This workflow:

- Takes as input one or more documents (video recording, audio-only recording, or screen recording)
- Uses Quick Look to convert all the pages to images
- Combines all the images into a master QuickTime movie
- Adds introduction, title, and exit movies to the master movie
- Adds a watermark to the master movie
- Generates 3 versions of the podcast (iPod/iPhone and Apple TV (SD))
- Publishes the podcast versions to the Podcast Library

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 45.

- 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 46. 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.

- Use Podcast Composer to modify a workflow.
- Duplicate and modify existing workflows by manually adding, removing, or modifying workflows tasks. This also includes adding custom properties.

  This level of customization gives you complete control over what workflows do.

Important: If you manually modify a workflow you initially created using Podcast Composer, you won't be able to open it or edit it using Podcast Composer.

Tools for Editing Workflows

The primary tool for editing workflows is Podcast Composer. However, if you want to manually modify a workflow, 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.
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. Install the workflow using the `podcast --installworkflow` command.
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.

2 Add, delete, and modify tasks and custom properties as needed.

For information about workflow tasks, see "Workflow Task Specifications" on page 58.

3 Configure the value of the new properties in Server Admin.

4 Deploy the workflow using the `podcast --installworkflow` command.

5 Verify that you can see the new workflow in Server Admin.

Deploying Scalable Podcast Producer Solutions

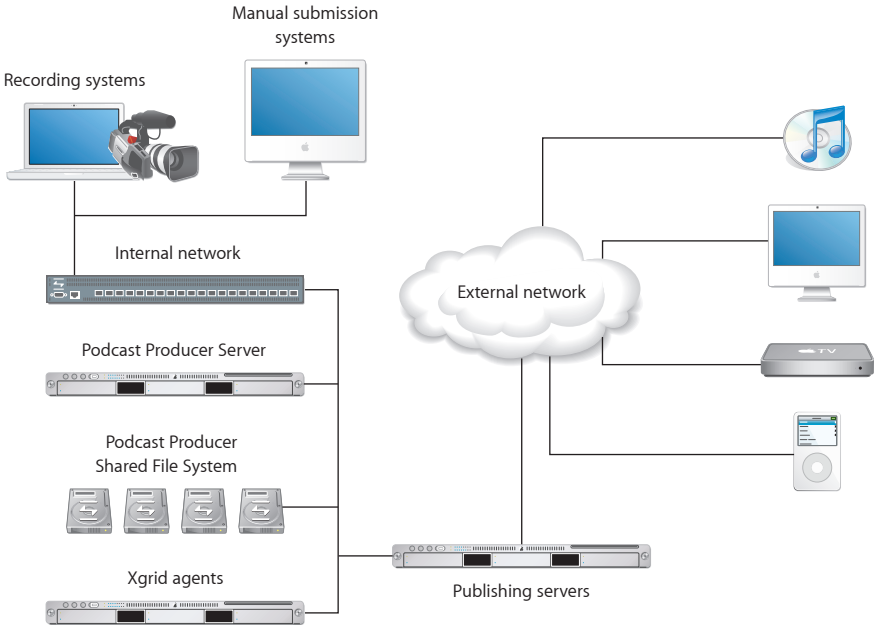
10

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 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 v10.6. Users with systems running Mac OS X v10.6 can upload video content using their systems.

Any system capable of running Mac OS X v10.6 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 v10.6 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 podcast Tool" on page 79.

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 so that recordings are directly stored on an 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:

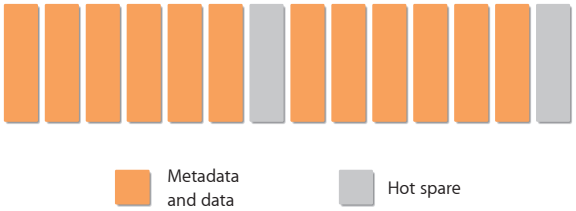
- Reconfigure your Podcast Producer settings
- Migrate the data from the old shared file system to a new one
- Possibly update your workflows to take advantage of additional processing power
- Possibly reconfigure your publishing services

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.

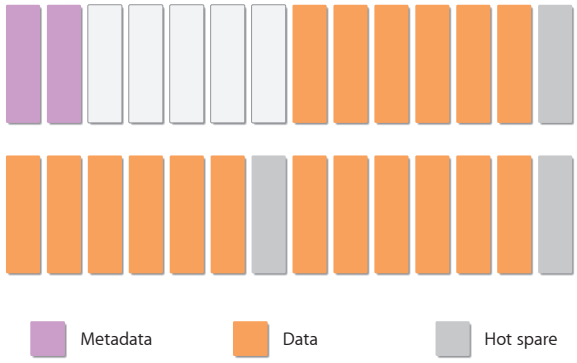
This configuration is a good starting point for organizations that do not have all the resources needed for a highly scalable deployment. It should handle a good amount of Xgrid nodes before slowing the entire system.

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 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: • 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.

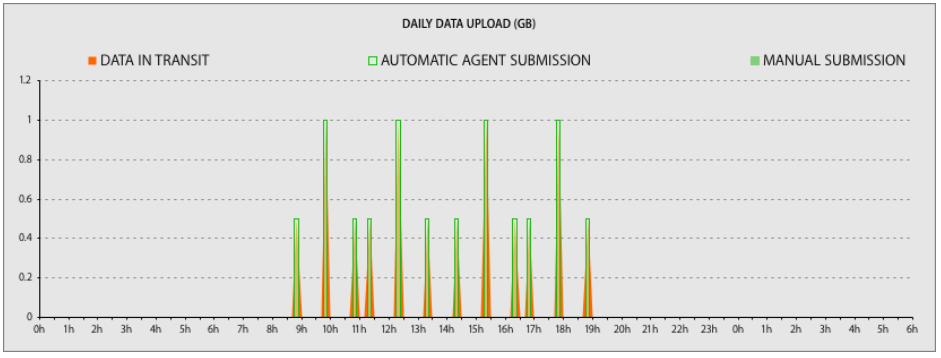
The colored cells help you see the recording pattern.

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

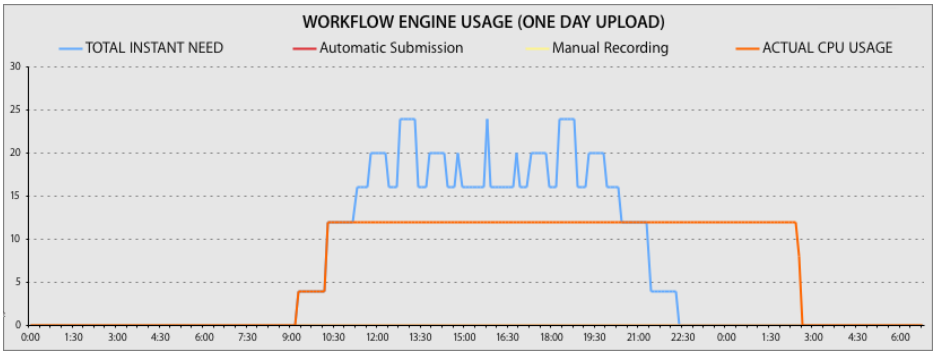
The following chart illustrates the daily upload times.



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.

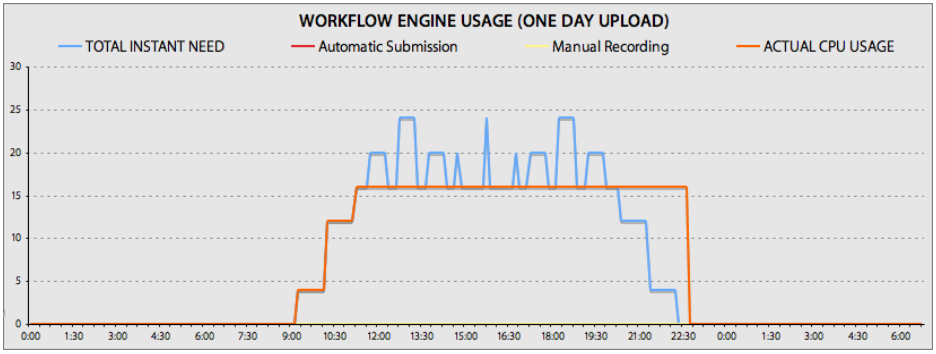


In the chart above, the blue line shows the instant computing need if unlimited computing resources were available. The orange line shows the actual CPU usage based on the number of Xgrid nodes available. In this case, there are three Xgrid nodes, each with four CPUs.

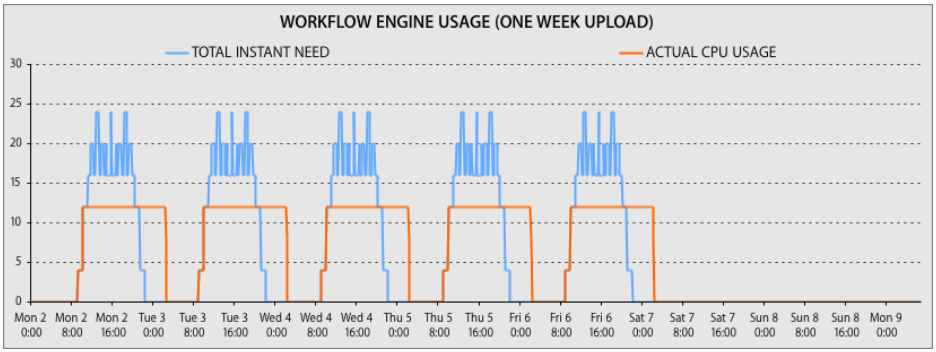
As show in the chart above, at 3:00 in the morning, all workflows completed and all content was published.

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

Podcast Producer Command-Line Tools

11

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 provide 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 allows you to interface with Podcast Producer server. For example, you can use this tool or command to get status information from the server, bond a local device to the server, and control the capture and submission of media for processing by Podcast Producer.

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.

Syntax

The following is the syntax of the `podcast` command:

```
podcast [-s server] [-u username] [-p password] [--auth {Password |
Kerberos}] [--checksslcert] [--timeout seconds] command [command-options]
```

Command Options

The `podcast` command provides the following categories of command options:

- Information commands
- Agent commands
- Camera control commands
- Submission commands
- Workflow administration commands
- Access Control List (ACL) commands
- Feed administration commands
- Catalog administration commands

For more information about `podcast`, see its man page.

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.

For more information about `pcastconfig`, see its man page.

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.

For more information about `pcastctl`, see its man page.

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.

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

Command	Description
`pcastaction annotate`	Adds annotations to the input movie.
`pcastaction approval`	Submits content for approval.
`pcastaction archive`	Archives the input movie at the specified location.
`pcastaction chapterize`	Adds chapters to an input movie by detecting scene changes.
`pcastaction compressor`	Submits an encoding job via Compressor.
`pcastaction documents2movie`	Takes a file, a folder of documents or a zipped archive of documents and generates a chaptered movie with the content of these files.
`pcastaction encode`	Encodes the input movie using the specified codec.
`pcastaction getposterimage`	Takes a movie and generates a poster image (PNG) by grabbing a frame from a specific time frame.
`pcastaction getpreviewmovie`	Generates a preview movie from the input movie.
`pcastaction groupblog`	Posts the content to the specified group blog.
`pcastaction iTunes`	Instructs the iTunes Store to check the specified RSS feed for new episodes.
`pcastaction iTunesU`	Posts the input video at the specified iTunes U tab.
`pcastaction jabber`	Sends a message via Jabber.
`pcastaction mail`	Sends a notification mail to the specified user using the mail template in the workflow's Resources/Templates folder.
`pcastaction merge`	Merges two movies with a fade transition between them.
`pcastaction notify_itunesu`	Notifies iTunes U that new content is available.
`pcastaction pip`	Takes 2 movies (main and secondary) and creates a Picture-in-Picture reference movie using a Quartz Composer composition.
`pcastaction preflight`	Runs the preflight script (System/Library/PodcastProducer/Resources/Tools/preflight_script) with the specified arguments.

Command	Description
`pcastaction postflight`	Runs the postflight script (System/Library/PodcastProducer/Resources/Tools/postflight_script) with the specified arguments.
`pcastaction publish`	Publishes the input file to a web or QTSS server.
`pcastaction publish2finalcutserver`	Publishes file to a folder watched by Final Cut Server.
`pcastaction publish2folder`	Publishes file to a folder.
`pcastaction publish2library`	Publishes file to the Podcast Library.
`pcastaction qceffect`	Applies a Quartz Composer composition to the input video file.
`pcastaction qtimport`	Prepares a QuickTime movie.
`pcastaction qceffect`	Applies a Quartz Composer effect (composition) to a movie.
`pcastaction shell`	Runs the specified shell script with the specified arguments.
`pcastaction template`	Processes a web or mail template into a localized content block to be used in mail or web postings.
`pcastaction title`	Adds the supplied title to the input video.
`pcastaction unpack`	Unpacks folder archives before running the main part of a workflow.
`pcastaction upload`	Submits content for approval.
`pcastaction userblog`	Unpacks folder archives before running the main part of a workflow.
`pcastaction watermark`	Superimposes the specified image as a watermark over the input video.
`pcastaction wikiserver`	Posts the content to the specified posting URL destination.
`pcastaction workflow`	Submits file to another workflow.

For more information about `pcastaction` and its commands, see its man page. You can also view help information about the commands of `pcastaction` by entering:

```
$ pcastaction help command
```

Monitoring Podcast Producer

12

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 (`pcastserverd`) activity.
Podcast Producer Server Error Log	Records error messages generated by the Podcast Producer server.
Podcast Producer Server Startup Log	Records error messages generated by the Podcast Producer server during startup.
Podcast Producer HTTP Access Log	Records all requests processed by the Apache server instance (`httpd`) used by Podcast Producer.
Podcast Producer HTTP Error Log	Records HTTP error messages generated by Podcast Producer's `httpd` instance.
Podcast Producer Application Log	Records external HTTP requests to the Podcast Producer server.

In addition, Podcast Producer server stores additional error logs in /Library/Logs/pcastserverd/DiagnosticReports/. Anytime a workflow fails, Podcast Producer server adds an error log to this folder.

The DiagnosticReports folder exists only on the computer running Podcast Producer server.

To view Podcast Producer logs using Server Admin:

1. Open Server Admin.
2. Select the server, then click the service disclosure triangle to show the services for administration.
3. In the service list beneath the server, select Podcast Producer.
4. Click Logs.
5. From the View pop-up menu, choose the log to view.

You can also view the Podcast Producer logs using Console (in /Applications/Utilities/).

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.

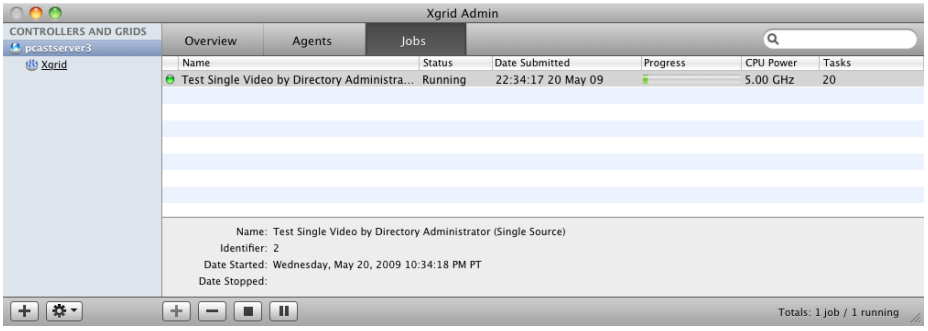
You can also look at the system log on the Podcast Producer server for Podcast Producer uploader entries.

Monitoring Xgrid Job Progress

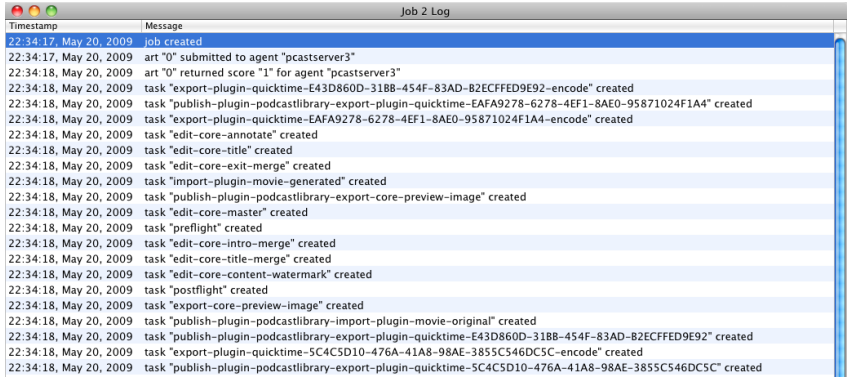
The Podcast Producer server sends a notification message when a Podcast Producer Xgrid job completes successfully, but you may want to actively monitor the progress of an Xgrid job.

Using Xgrid Admin

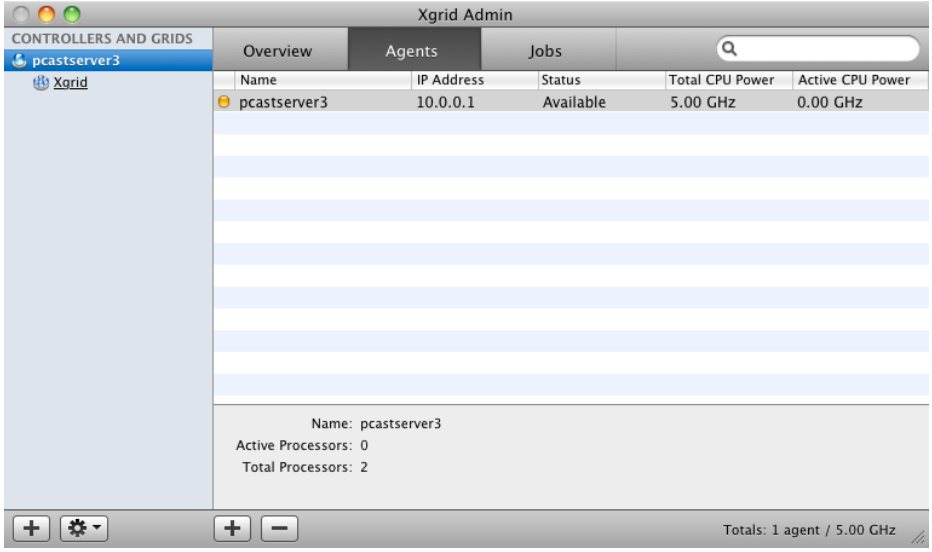
To actively monitor Xgrid job progress, use Xgrid Admin. Xgrid Admin shows you the progress of Xgrid jobs and whether they succeeded or failed, as shown in the following figure.



In the figure above, Xgrid Admin shows a job in progress. This job has 20 tasks and is currently using 12.40 GHz of CPU power. To drill down and find more information about the status of tasks, double-click the job. A window appears listing the tasks that were processed and providing more details about each task.



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 Feb 23 13:20:33 -0700 2009 -- 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 Feb 23 13:20:34 -0700 2009 -- 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 = 2009-02-23 13:34:01 -0700;
dateStarted = 2009-02-23 12:22:52 -0700;
dateStopped = 2009-02-23 13:23:10 -0700;
dateSubmitted = 2009-02-23 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 its man page.

Index