

# Podcast Producer: Anatomy of a Workflow

Podcast Producer, an integrated part of Leopard Server, provides a complete publishing and management system for audio and video podcasts. Your users can submit raw material into the Podcast Producer system and have the files automatically processed into a suitable format so that the podcast can be published to iTunes and Leopard's built-in blog services.

*Workflows* are the cornerstone of the Podcast Producer system. Workflows define the process to convert the incoming raw material into the final content, including translating and recoding audio and video for different targets, and publishing and creating the environments, blog pages and other materials that make your podcast available to a variety of different targets. Because the workflow instructions are independent of the content, you can publish podcasts repeatedly using the same workflow to achieve consistently high quality results.

Understanding the anatomy of a workflow, from the contents and definition of the workflow environment, through to the methods of interaction with other components in Leopard Server such as Xgrid, will help you plan new workflows and organize and structure your Podcast Producer environment more effectively.

In this article, designed for Podcast Producer administrators, we will examine Podcast Producer from the perspective of a submitted job. By examining the contents and processing of a workflow, Podcast Producer administrators will learn about the significance of the different elements, and how they work together to help produce the final podcast or video cast.

## Anatomy of a Workflow

Podcast Producer creates a podcast, which is the result of a workflow of the individual atoms of structured and unstructured data. When creating a podcast, Podcast Producer uses a number of different applications and Mac OS X utilities to convert a podcast into the target format and structure that you want. The method of conversion, conversion settings, target formats, structure, and how the information integrates into the rest of the Leopard Server environment is controlled by a workflow. The workflow defines the different parameters passed to the command-line applications, and controls the processing progress of the podcast from the source material to the published product.

Workflow Definition Files are stored as a directory structure, called a bundle, that groups related files required to operate with the workflow. This includes the files that make up the resources (such as watermarks, HTML pages, and graphics) used during publication.

Core to Podcast Producer is a suite of command-line applications that are used to help format, convert and assemble the different components. The definition for how these systems work together is contained in the Workflow Definition Files, the template that defines the methods and steps required to take the raw content and produce the target material.

An instance of Podcast Producer running on Leopard Server receives the raw material from the client and supplies the raw material and the workflow to the Xgrid controller, which distributes the work across Xgrid nodes. The node follows a workflow task, processing the files and assembling of the content, before sending the resulting output back to the Podcast Producer server for publication.

Within Podcast Producer you can use and configure multiple workflows so that you can apply a different set of rules and processes to different jobs. For example, you might configure a Podcast Producer workflow that accepts an audio-only podcast and encapsulates it into a specially formatted blog post to appear on your website. For video podcasts, you might prepend and append videos that show your company logo and production information and then submit it to iTunes.

## Workflow Structure

Although the term workflow is used to describe a sequence that is followed when work is submitted to Podcast Producer, the workflow itself is actually a physical file (a bundle) that contains the definition for processing (a task list), any additional files required to achieve the processing and target formats, and some metadata about the workflow (the description, license and version data).

The structure of the Workflow Definition File Bundle is a series of steps, each of which call a number of command-line applications to perform a particular action. Through each step, the podcast is processed until you have produced all the desired target components.

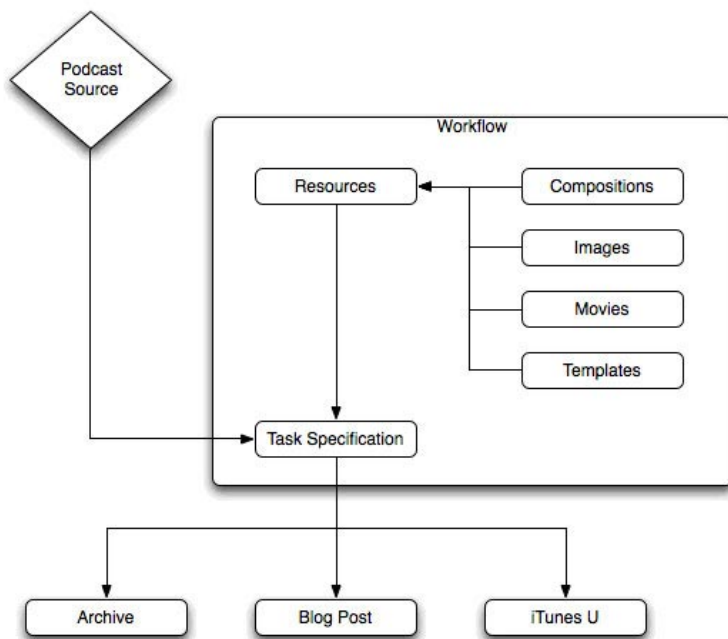
To understand the physical contents of a workflow, it is worth examining the logical structure of the workflow when it is used to process the raw source file into one or more of the target formats. You can see a sample of this in Figure 1.

### Related Articles

- Podcast Producer: Writing Actions
- Podcast Producer: Using the Command Line
- Podcast Producer: Publishing on YouTube
- Podcast Producer: Scheduling Podcasts
- Leopard Technology Series for Developers: Server Overview

### Resources

- Reference Library: Mac OS X Server
- Mac OS X Server Podcast Producer Workflow Tutorial (PDF, 1MB)
- Mac OS X Server Podcast Producer Administration (PDF, 2.1 MB)
- Leopard Server QuickTours (in iTunes)
- Quick Tips Videos
- Podcast Producer Mailing List
- Podcast Producer Technology Brief (PDF, 1.2 MB)



**Figure 1:** Workflow Definition File Bundles in Podcast Producer

The source material is combined with the workflow resources using a Workflow Task Specification File that combines and merges the content into the target format. For example, in a simple "audio podcast workflow" that is converted into a blog post, the Workflow Task Specification File would define the following steps:

1. Convert the audio from the source format suitable for publication through a blog post.
2. Use the predefined template (from the Templates directory of the Resources directory of the Workflow) to format a blog post that will publish the audio content.
3. Submit the blog post into the Leopard blog system.

The standard elements of a Workflow Definition File Bundle consist of these files and configuration details that are used in combination to produce the target content.

### Workflow Definition File Bundle Contents

You can display the contents of an existing Workflow Definition File Bundle (from /System/Library/PodcastProducer/Workflows) by clicking on the file with the right-mouse button and selecting "Show Package Contents" from the context menu. This will show the files that make up the workflow and their structure. The typical contents of a Workflow Definition File Bundle include:

- Info.plist**—an XML-based property list file containing the base information about the workflow, including the description, name and other data used to identify and describe the workflow within Podcast Producer.
- LICENSE**—the license information for using the workflow.
- Resources**—the directory containing the related resources for the workflow
- version.plist**—an XML-based property list file containing the version information for the Workflow Definition File Bundle

The Resources directory is critical, as it defines the different elements and files that make up the workflow, including the property list file (template.plist) that defines the execution of the tasks required to produce the desired output. The contents of a typical Resources directory include:

- Compositions**—the Quartz Composition Files used with or applied to podcast video sources. For example, you can use Quartz Compositions for filtering video and changing the output to be black & white or sepia toned, or for providing transitions between video fragments.
- lang.lproj**—the language-specific directory containing the localized data. There should be one directory for each language.
- template.plist**—the main property list that defines all of the parameters, configuration options, and tasks that control the processing of the source material.
- Templates**—the templates used during processing. These include HTML templates (used when posting to a blog) and platform specific files, such as specific templates to be used when a blog post is viewed on an iPhone.
- Tools**—any special tools required to perform any processing. Standard tools do not need to be included as they will already be installed on both the Leopard Server and the Xgrid clients used to perform the processing.

The Resources directory within the Workflow Definition File Bundle contains only the workflow-specific resources. You can add global resources to Podcast Producer by adding files and templates to the

/Library/PodcastProducer/Resources. Standard organization watermark, image and video material should be stored in these global resource locations so that Workflow Definition File Bundles contain only workflow-specific information.

## Using Properties

There are a number of configuration and property lists that define the workflow information. The primary property file is the `template.plist` file within the Resources directory, and this holds the main specification information for the workflow. Although at first glance the definition seems very simple, it is also very powerful, as it allows for a very flexible execution structure and sequence:

- 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.
- name**—this entry specifies the name of the Xgrid job. This is used to identify the job when the workflow is submitted to the Xgrid controller.
- notificationEmail**—specifies the mail address that the Podcast Server to notify the administrator about the status of workflow jobs submitted into the Xgrid system.
- taskSpecifications**—this is the key to the Workflow Definition File, as it is this element that enables you to specify the individual steps that will be executed to produce the final file.

The property lists and configuration within the workflow interact with the general properties configured through the Podcast Producer administration interface. For example, the `notificationEmail` property in the workflow property list can be configured to use the value of a general email address property within the Podcast Producer. Using this, you can have both individual workflows with their own specifications, and workflows that make use of the generic properties.

You can edit the global properties within the Podcast Producer administration tool. You can see an example of the default settings provided with the tool in Figure 2.

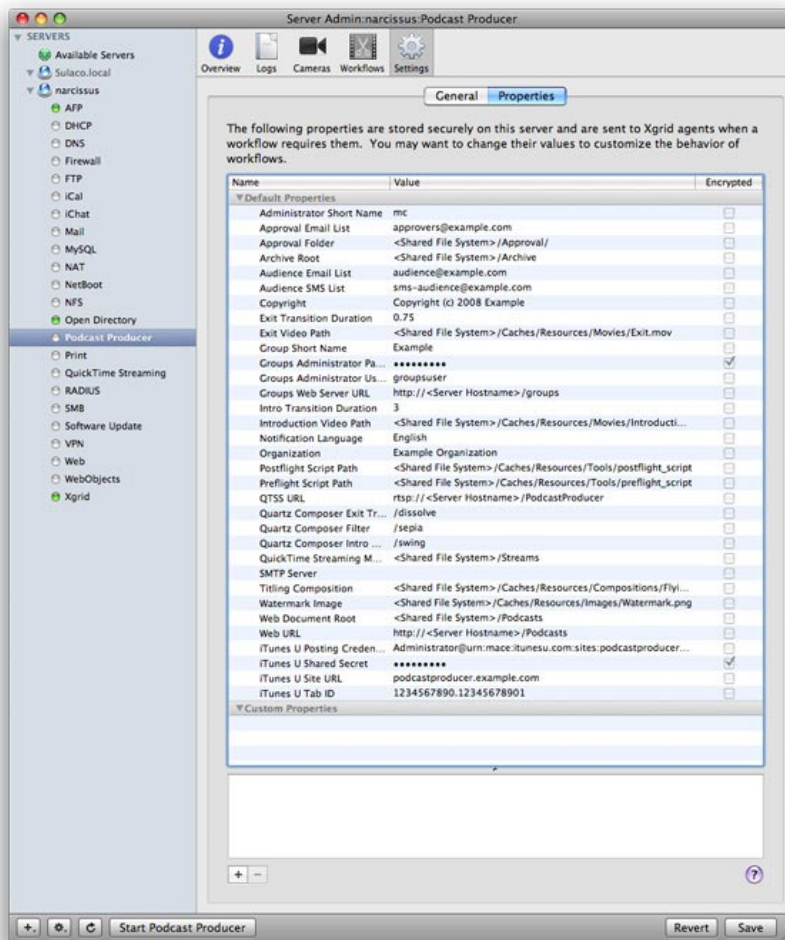
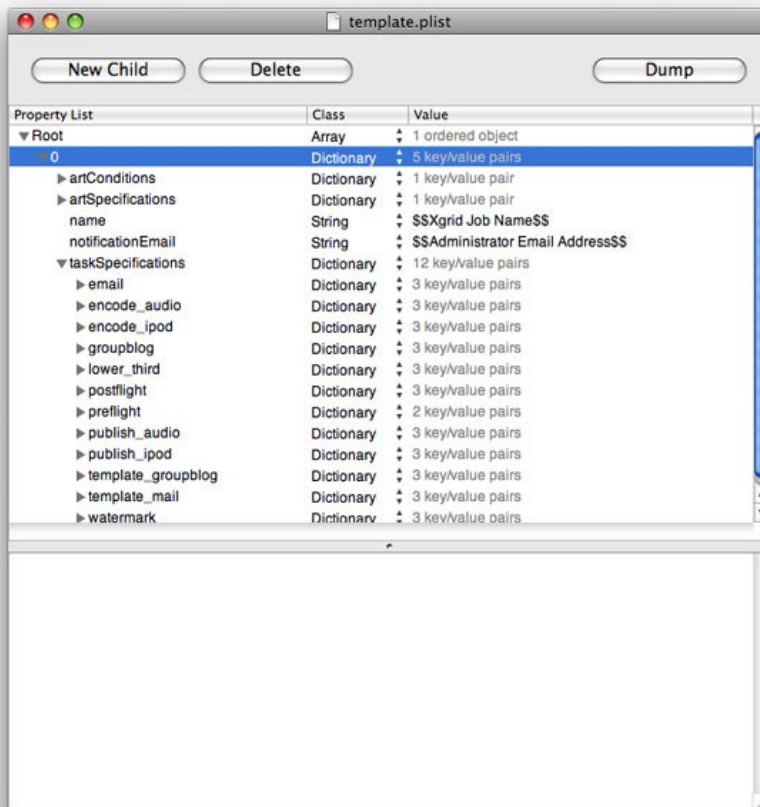


Figure 2: Editing Global Podcast Producer Properties

For the workflow-specific properties, you can use any text editor and edit the raw XML or use the Mac OS X Property List editor, as seen here in Figure 3.



**Figure 3:** Editing Workflow-specific Properties

You can access any property from the workflow or the global property specification by placing the name of the property within double-dollar sign qualifiers. For example, the standard Web URL property can be used within your specification by quoting `$$Web URL$$`. You can use these property values throughout your Workflow Definition File, both within other properties and within the task specification. It is these pre-defined properties and values that enable you to identify the source material during processing.

### Defining the Workflow Execution

The `template.plist` file, as mentioned, contains the bulk of the configuration information about the workflow. The properties defined within this file are used in combination with a number of standard properties configured globally within Podcast producer. Specifically within each workflow is a set of properties called the `taskSpecifications`.

The Task Specification File in the workflow controls the commands that are executed on the source material and associated resources. These definitions form the action part of the process. For each action specification, you have three properties: the command name, the arguments to the command and the dependencies.

For example, many of the standard workflows include the ability to watermark your productions to help protect your productions once they have been published. Watermarking is achieved by using the `pcastaction` tool with the `watermark` command. You can see the definition of this action within one of the standard workflows by examining the XML source:

```
<dict>
  <key>arguments</key>
  <array>
    <string>watermark</string>
    <string>--basedir=$$Base Directory$$</string>
    <string>--input=$$Content File Basename$$$$Content File Extension$$</string>
    <string>--output=$$Content File Basename$$-watermarked.mov</string>
  </array>
  <key>command</key>
  <string>/usr/bin/pcastaction</string>
  <key>dependsOnTasks</key>
  <array>
    <string>unpack</string>
  </array>
</dict>
```

You can also define and use custom actions by including scripts and applications within the Resources/Tools directory of your Workflow Definition File Bundle.

The `dependsOnTasks` property within the specification controls the execution sequence of the processing by listing which tasks should have been executed before this task starts. For each task, it is the dependency of previous tasks to describe the sequence of the overall process.

When defining tasks for a workflow, the Task Definition File must be designed so that the input and output files match for each step of the process. A failure in any single step of the task process will cause the entire workflow to fail.

## Standard Workflows

Podcast Producer comes with a number of standard workflows designed to address the most common target outputs and formats for a typical podcast environment. Although the workflows are generic, you can customize these workflows by editing the standard properties applied to all workflows, without having to edit the workflows directly.

Understanding the existing standard workflows can give you a good idea of the type of processing and manipulation available through Podcast Producer and how you can adapt and modify your own workflows to suit your environment.

The blog-based workflows accept an incoming Podcast audio or video file and prepare the file for posting into a blog configured within the Leopard Server blog environment. The different workflows perform the following actions:

**Blog.** This action archives the source material, encodes to audio/video format and publishes the result to the group blog with announcement email.

**Blog and iTunes with intro.** This action is similar to Blog, but adds a watermark and introductory before encoding to both Blog and iPod formats. It also notifies the iTunes podcast directory of the new post.

**Blog and iTunes with intro and effects.** This action is identical to the Blog with iTunes with intro, with the addition of a Quartz filter before encoding.

**Blog and iTunes with intro and effects.** This action is identical to Blog and iTunes with intro and effects, but also adds an Apple TV encoding format.

**Blog and iTunes with watermark.** This action is similar to Blog, but adds a watermark before encoding, and also provides iTunes version.

**Blog with streaming.** This action is identical to Blog, but adds an encoding format suitable for 3GPP phone streaming.

**Blog with streaming an intro.** This action is identical to Blog with streaming, with the addition of a watermark and intro material before encoding.

**Blog with streaming and no archive.** This action is identical to Blog with streaming, but without keeping an archive of the source material.

**Blog with streaming, intro and effects.** This action adds an effect to the source material before encoding.

**Blog without archive.** This action is identical to blog but does not keep an archive of the source material.

## Creating New Workflows

You can create additional workflows within the `/Library/PodcastProducer/Workflows/` directory rather than the System directory to prevent the workflows from being deleted during a software upgrade.

Creating new Workflow Definition File Bundles enables you to customize and expand the targets and formatting of podcasts as they are submitted through the system. The easiest way to create a new workflow is to copy one of the standard workflows and adapt it for your requirements.

To create your new workflow, copy an existing Workflow Definition File Bundle into the `/Library/PodcastProducer/Workflows/` directory and give the new bundle a new name. Before extending and expanding the definition of the workflow, the following elements must be modified:

Change the name of the Workflow Definition File Bundle to reflect the content.

Change the `CFBundleIdentifier` and `CFBundleName` within the Info.plist file to reflect the new name.

Alter the version.plist information so that you can track changes to your Workflow Specification File.

With these core modifications in place, you can now edit the workflow to add your desired customizations. You can do this by adding new templates and media elements, and then editing and modifying the Task Specification File to perform the required processing to convert the submitted podcasts into your target design.

## Execution and Distribution of Work

The execution of a workflow is very simple. As you have already seen, the different steps in the workflow call a number of command line applications to perform each step until the workflow reaches the target state.

In order to distribute the processing work among multiple machines using Xgrid, the core elements from Podcast Producer (source material and shared resources) are shared using standard file sharing. A job is then submitted to an Xgrid agent enclosing the appropriate workflow that the media was submitted under, and the details about the job specifics (such as the name and location of the source files). The unique job

information is submitted as properties that are then used during the property replacement in the various actions executed by the Xgrid agent.

The ability for Podcast Producer to distribute the work to Xgrid agents is due to the standalone nature of the Workflow Definition File Bundle and the sharing mechanism and systems built into Leopard Server.

During the final part of the workflow, the final file to be submitted into one or more of the destination systems. Podcast Producer makes use of other elements within Leopard Server, including the built-in blog system, the mail server and iTunes U service, all of which also make use of the Web publishing and hosting environment. The configuration of these processes is handled through a combination of workflow-specific configuration data and the global Podcast Producer properties.

## **What You've Learned, and What's Next**

Understanding the anatomy and components of a typical Podcast Producer workflow enables you to develop and configure your own Workflow Definition File Bundles. By leveraging the power of the Leopard Server environment, including the blog system and the ability to distribute the processing of work through Xgrid, it is easy to set up and configure customized Podcast Producer workflows and publishing systems. In future articles we will be looking in more detail at how to develop and create your own workflows and how to use the Workflow Definition File to help plan your Podcast Producer environment.

**Updated:** 2008-10-22