Podcast Generator - Documentation

http://podcastgen.sourceforge.net

Table of Contents

Overview

Podcast Generator is an open source podcast publishing script which allows users to upload audio and video files through a simple, yet powerful, web administration interface and automatically generates a podcast w3c-compliant feed.

Some technical details

- Database Environment: XML-based
- License: GNU General Public License (GPL)
- Operating System: OS Independent (Written in an interpreted language)
- Programming Language: PHP
- User Interface: web-based

Main features of Podcast Generator

- Extremely easy to install (Web Setup Wizard);
- Free and open source;
- Supports any media filetype (mp3, ogg, mpg, m4v, mov, etc...) and allows to create mixed audio and video podcasts;
- The XML feed automatically generated is fully compatible with aggregators (e.g. Juice and iTunes), meets the w3c standards and supports iTunes specific tags;
- Categories support: each category features its own podcast feed;
- User friendly web administration interface to upload, edit, delete episodes and fully customize the script;
- Dynamically created "Recent podcasts" and "Podcast Archive" web pages;
- Themes Support;
- Multilanguage support;
- Web or FTP upload of audio/video episodes;
- Web mp3 streaming player;

Internal structure of the script

As a matter of fact, Podcast Generator not only handles the automatic generation of the podcast feed, the script also offers a full-featured web administration and

produces dynamic web pages showing the "Recent Podcasts" and "Podcast Archive", generating a detailed page for each single episode (permalink).
The data associated to episodes is included in the XML feed on one hand, while, on the other hand, it is also regarded during the dynamic web pages generation process. The RSS feed is managed by the client aggregator, while, on the web side, the whole process is handled by the main file (index.php) which includes all the php modules and renders the web pages.

Database

Podcast Generator relies on XML to store episodes and categories data.
On uploading or editing every episode, an XML 1.0 file containing information provided by the user (e.g. title, description, related categories, keywords, etc...) is generated and associated to the uploaded media file. All this data is eventually merged into a single XML file (i.e. the RSS feed).

The adoption of XML to store episodes data offers several advantages:

- XML fields structure is flexible and can be updated incrementally;
- XML is based on international standards;
- XML allows metadata exchange between applications;
- XML is platform-independent and assure forward and backward compatibility;
- Data stored into XML structure can be used to develop further web ontologies schemas and semantic models.

Moreover the employment of XML as database allows to keep low server requirements to run the program: in order to install Podcast Generator a server with PHP support is needed and no DBMS (e.g. MySQL) is required.

The following table shows the structure of Podcast Generator episodes database:

Tag in Podcast Generator XML DB	Corresponding tag in the generated RSS feed	Description	Notices
<episode>	<item>	Root tag which defines a single episode.	
<titlePG>	<title>	Episode Title.	CDATA section.
<shortdescPG>	<itunes:subtitle> and <description>	Short Description.	CDATA section.
<longdescPG>	<itunes:summary>	Long Description. A summary of the episode content.	CDATA section (up to 4000 characters). In iTunes the contents of this tag are shown in a separate window that appears when the "circled i" in the Description column is clicked.

<imgPG>		Filename of the image associated to the episode.	This tag is currently not applied to the RSS feed. The image is currently displayed on "Recent Podcast" and "Permalink" pages.
<categoriesPG>		Categories associated to a single episode.	This tag is currently not applied to the RSS feed. It may currently contain up to 3 nested tags: <category1PG>, <category2PG> and <category3PG>. The categories are listed on the "Podcast Archive" page.
<keywordsPG>	<itunes:keywords>	Keywords associated to a single episode.	In iTunes the content of this tag is not visible but can be searched.
<explicitPG>	<itunes:explicit>	Specifies the presence of explicit content in the episode.	The 3 possible values are: "yes", "no", and "clean". If this tag is populated with "yes", an "explicit" parental advisory appears next to the podcast artwork on the iTunes Music Store, and in the Name column in iTunes.
<authorPG>	<itunes:author> and <author>	Author of the episode.	This tag contains 2 nested tags: <namePG> (author's name) and <emailPG> (author's email address). According to the RSS 2.0 specification[1], the tag <author> must contain a valid email address. In iTunes the content of the <itunes:author> tag is shown in the Artist column.

According to the iTunes technical specification by Apple[2], all the above values are limited to 255 characters, except for *<longdescPG>* (i.e. *<itunes:summary>*) which can be up to 4000 characters.

The categories structure is equally stored into an XML file (categories.xml) which contains the name of categories along with a corresponding unique ID automatically generated by the script .
The categories database structure is quite simple but effective: the <category> tag defines a single category and contains two nested tags: <id> (unique ID) and <description> (i.e. The category name).

Server limits and workarounds: the "FTP Feature"

When uploading media files through Podcast Generator web admin interface, a decisive limit users can encounter could consist in the maximum upload size allowed for each single file. This value doesn't depend on Podcast Generator configuration, yet it is set in the server setup. A default limit of 8MB per single file, for example, could be adequate to upload audio files but not suitable for video podcasting.
This value can be easily increased editing the *upload_max_filesize*, *post_max_size* and *memory_limit* variables in the "php.ini" configuration file.
However, if a user is not allowed by any means to modify the server configuration (e.g. in some shared or free hosting environments), Podcast Generator offers the

---

1   http://cyber.law.harvard.edu/rss/rss.html
2   http://www.apple.com/itunes/store/podcaststechspecs.html

"FTP Feature", an alternative approach to upload large files. Files can be alternatively uploaded via FTP without any size limitation and processed through the "FTP feature" (available in the web admin interface of the script), which scans the media directory and indexes all the uploaded episodes: if available, titles and authors data is read from the ID3 tags, while users can eventually edit the new episodes adding additional information, such as keywords and long description.

External components

Podcast Generator package includes third-party widespread components aimed at specific tasks.

GetID3()

GetID3() (http://getid3.sourceforge.net) is a PHP script that extracts useful information from MP3s and other multimedia file formats. Podcast Generator makes use of this script in order to extract duration, filetype, bitrate and frequency from audio and video episodes. As a further matter, the "FTP Feature" of Podcast Generator calls this script to extract title and artist from mp3 ID3 tags.

GetID3() is free software released under the terms of the GNU General Public License.

Last RSS

LastRSS (http://lastrss.oslab.net/) is a simple RSS parser written in PHP. This component is used by Podcast Generator to read its official news feed on Sourceforge.net (http://feeds.podcastgenerator.net/podcastgenerator) and display the latest news (new releases, updates) in the main page of the web admin interface.

LastRSS is free software released under the terms of the GNU General Public License.

Audio Player

Audio Player (http://www.1pixelout.net/code/audio-player-wordpress-plugin/) is a highly configurable single track mp3 player designed in Adobe Flash[3]. This player was originally conceived as a Wordpress[4] plugin and is released under the terms of the GNU General Public License.

---

3   http://www.adobe.com/products/flash/
4   http://wordpress.org

Prototype and Scriptaculous

Prototype (http://www.prototypejs.org) is a cross-browser javascript frameworks aimed to ease development of dynamic web applications. Scriptaculous (http://script.aculo.us) is an addition to the Prototype framework which allows to implement animations, graphic effects and ajax controls on web pages. Podcast Generator adopts this frameworks in order to simplify the user interaction with the administration interface allowing an elegant showing and hiding of menu items and options on demand.

Scriptaculous and Prototype can be used free of charge under the terms of a MIT-style licence[5].

XMLParser

XMLParser (http://www.criticaldevelopment.net/xml/) is an XML parser which works in both PHP 4 and PHP 5 despite the difference in DOM extensions management between these two versions. This component assure high compatibility in Podcast Generator XML manipulation.
XMLParser is released under the terms of the GNU General Public License.

Tiny MCE

TinyMCE (http://tinymce.moxiecode.com) is a platform independent web based Javascript HTML WYSIWYG editor. It has the ability to convert HTML "textarea" fields or other HTML elements to editor instances. This component is adopted by Podcast Generator to provide users with a visual editor to the episodes "long description" field (which supports HTML code) and to the "Freebox", an HTML box that allows to add links, text and other HTML tags and displays them on the web pages rendered by the script.

TinyMCE is released under the terms of the GNU Lesser General Public License.

Themes handling

Podcast Generator doesn't adopt any existent template engine (such as SMARTY[6]) to handle themes and "skins". A simple template engine based on string replacing has been developed.

---

5   Scriptaculous License: http://github.com/madrobby/scriptaculous/wikis/License
Prototype License: http://dev.rubyonrails.org/browser/spinoffs/prototype/trunk/LICENSE?format=raw
6   http://www.smarty.net

The design of a new theme for Podcast Generator requires a basic knowledge of (x)HTML.
In order to convert any (x)HTML layout into a Podcast Generator theme, some special text strings must be included within the document. These strings actually consist in simple text that Podcast Generator will automatically recognize and replace with the related content.

The string "-----PG_PODCASTTITLE-----", for example, will be dynamically replaced with the title of the podcast by Podcast Generator template engine.

As soon as a new theme is ready, this must be simply placed in a folder (along with any related image, stylesheet, etc...) under the "themes" directory in the script root. The new theme will automatically appear among the available themes list in Podcast Generator administration interface.

The following table shows the list of the text strings that should be included in every theme, their recommended placement within the (x)HTML document and the expected results:

String	Recommended placement	Result
-----PG_PAGETITLE-----	<head>	Display the title of the page
-----PG_METATAGS-----	<head>	Generate Meta Tags (Author, Copyright, Description, Keywords)
-----PG_JSLOAD-----	<head>	Generate Javascript requests (<script src="...">) when needed (e.g. Admin pages with WYSIWYG editor etc..)
----PG_PODCASTTITLE-----	<body>	Display podcast Title
-----PG_PODCASTSUBTITLE----	<body>	Display podcast subtitle
-----PG_PODCASTDESC----	<body>	Display podcast description
-----PG_MENUADMIN-----	<body>	Display "Admin" menu
-----PG_MENUHOME-----	<body>	Display "Home" menu
-----PG_MENUARCHIVE-----	<body>	Display "Podcast Archive" menu
-----PG_RIGHTBOX-----	<body>	Display box containing feed address and iTunes subscription button
-----PG_FREEBOX-----	<body>	Display "Freebox", an HTML box users can customize through a visual editor from the admin section.
-----PG_MAINBODY-----	<body>	Display the main content generated by the script (latest episodes, archive, admin interface etc...)
-----PG_FOOTER-----	<body>	Display footer

In addition, a file called *functions.php* can be added in each theme directory: this will

allow to define new strings which will be replaced or will replace the default text string with custom content.

Security

Podcast Generator administration login is handled through "sessions", a PHP native feature that assigns a unique session ID to the user and allows to preserve certain data across subsequent accesses. The administrator's password is stored in config.php file and encrypted with MD5 algorithm. Furthermore the script performs multiple checks on variables passed through the POST and GET methods in the pages URL in order to prevent cross site scripting attacks.
Use of Podcast Generator with Register Globals set to ON is strongly deprecated due to potential security issues.

Additional information

Nearly 12000 lines of code written from scratch[7].
Podcast Generator has been downloaded over 44000 times from Sourceforge.net servers[8] since April 2005.

—————————————

This documentation has been written by Alberto Betella, developer of Podcast Generator and is released under Creative Commons Attribution - Non Commercial - Share Alike license (http://creativecommons.org/licenses/by-nc-sa/3.0/)

---

7 External components are excluded in line counting.
8 http://podcastgen.sourceforge.net/download.php