

JetPhoto Server

Developer's Notes

Document Version 1.0

JetPhoto Software
© 2010 Atomix Technologies

1. Understand Property List

While browsing JetPhoto Server file directories, you'll find there are many files have the same filename extension '.plist'. Actually, all of JetPhoto Server's website configuration, album metadata and template profiles are saved in Property List files. The Property List file is a special formatted XML file with '.plist' filename extension.

As the original, Property List is firstly applied on Mac OS X platform, constructed for saving user data with strings, numbers, dictionaries and arrays. For more detail information about Property List on Mac OS X, you can also refer to Apple's developer website <http://developer.apple.com>. Due to being encoded in XML, the plist file can be opened by any XML or text editor, developer can also handle it by the general XML parsers on any platform.

JetPhoto adopts Property List file to save website information and album metadata. However the Property List files used for JetPhoto Server all conform to a given 'Head-Items' structure. In other words, the JetPhoto Server property list contains one Head data object and an array of several Item data objects. Each Head or Item object contains a series of tags, and each tag is a key-value pair. The key field indicates the tag name, and the value field must be a string expression. Presently, JetPhoto only accepts the property lists with UTF-8 encoding.

As example, the XML file below shows a typical property list used for JetPhoto Server:

```
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>Head</key>
  <dict>
    <key>Format</key>
```

```

    <string>com.atomixtech.jetphoto.users.datalist</string>
    <key>Version</key>
    <string>1.0</string>
  </dict>
  <key>Items</key>
  <array>
    <dict>
      <key>ID</key>
      <string>1</string>
      <key>Password</key>
      <string>guest</string>
      <key>UserName</key>
      <string>guest</string>
    </dict>
    <dict>
      <key>ID</key>
      <string>2</string>
      <key>Password</key>
      <string>supervisor</string>
      <key>UserName</key>
      <string>supervisor</string>
    </dict>
  </array>
</dict>
</plist>

```

All property lists used for JetPhoto Server must have two tags 'Format' and 'Version' in Head segment, by which to declare the unique information type. And each Item must have a tag named 'ID' to indicate its position saved in the Items array.

The property list with 'Head-Items' structure has another alias that is called DataList, perhaps you'll also find this term in other JetPhoto technical documents. To open DataList file by Web programming, the developer can use the PHP class DataList that is located in 'jpserver/system/frameworks/datalist.php'. If you are a Windows programmer, you can also try to use the DataList COM DLL to handle DataList file in Windows programming environment, such as ASP, VB. datalist.dll is an enclosed component installed with JetPhoto Studio Windows version.

2. Album Data Package Structure

All album data packages are stored in folder 'data'. User creates album data package by JetPhoto Studio on Mac or PC, and uploads them to JetPhoto Server's data folder through WebSync action, FTP or other network file sharing tools.

Every album data package is named as the album's unique name with a .data suffix. There are three picture folders and an album.plist file saved in each album data package.

Normally, JetPhoto Studio resizes all original photos to generate three copies with different sizes for Web publishing. These three photo copies are Thumbnail image, Preview image and Standard image. All resized photo files are saved as JPEG format, the filename of them will be kept same as that of their original. The first photo copy is for thumbnail, whose bounds size is 128x128 pixels. The bounds size of the preview image is 256x256 pixels. The bounds size of the standard image is not fixed, user can set it while exporting or publishing album in JetPhoto Studio, its recommended size is 640x640 pixels. These three photo copies are separately stored in three different folders named 'size_0', 'size_1' and 'size_2'. Sometimes user can even keep original-size pictures in the album data package, and then these original photo files are stored in another picture folder named 'size_full'.

File 'album.plist' is a property list XML file to save the additional album information and the description, keywords and other properties of each photo belongs to this album. Here explains some major tags saved in a typical 'album.plist':

Head Tags

Key	Sample Value	Description
CanvasHeight0	128	The bounds size of picture files saved in

CanvasWidth0	128	folder 'size_0'.
CanvasHeight1	256	The bounds size of picture files saved in folder 'size_1'.
CanvasWidth1	256	
CanvasHeight2	640,	The bounds size of picture files saved in folder 'size_2'.
CanvasWidth2	640	
Title	My Album	Album title
Identification	my_album	Album short name (only allows to contain English alphabetic, numeric characters and underline mark)
Description		Album description
Keywords	Apple;Berry;Cherry	All keywords referred in this album (separated by semicolon)
ItemCount	18	The photos count number in this album.

Item Tags

Key	Sample Value	Description
File	IMG_10221.JPG	Photo file name
Created	2004-12-31 21:36:38	The time when the photo file was created (yyyy-mm-dd hh:mm:ss)
Modified	2004-12-31 21:36:38	The time when the photo file was modified (yyyy-mm-dd hh:mm:ss)
Archived	2004-12-31 21:36:38	The time when the photo file was imported into JetPhoto Studio (yyyy-mm-dd hh:mm:ss)
Height	1200	The original picture size (pixels)
Width	1600	
Size	501832	The original picture file size (bytes)
CameraMaker	FUJIFILM	Camera maker and model
CameraModel	FinePix S5000	
Aperture	f 4.0	Aperture (f number), shutter speed, focal length, ISO speed, flash mode, etc.
ShutterSpeed	1/135 sec	
FocalLength	8.4 mm	
ISOSpeed	400	
Flash	Off	
Longitude	121.514089	The GPS location where the photo was captured. (-180 to 180, -90 to 90)
Latitude	31.277497	
CaptureDate	2004-12-31 21:00:30	The time when the photo was captured by the camera (yyyy-mm-dd hh:mm:ss)
Description		Photo description
Keywords	Apple;Berry;Cherry	Photo keywords (separated by semicolon)

3. Design Appearance Templates

The folder 'view' stores all album appearance templates (.view packages). A template contains several sub-views to represent a photo album in different layout. Normally, the sub-view can be Thumbnail (thumbnail view), Picture Story (gallery view), Slideshow (slideshow view), Poster (detail view), Album Cover (cover view) or other user-defined view.

Each .view package is a folder that includes some appropriate template files. JetPhoto Server's album appearance template is a kind of smart template that includes PHP scripts. To represent photo albums in a series of visual web pages, such as Thumbnail, Slideshow, the PHP program of JetPhoto frameworks loads album metadata and pictures from the .data package, and fill all dynamic tags with album data elements in the page template saved in the .view package. In other words, JetPhoto converts the uniform album data package to the deluxe web pages by .view package.

For any .view package, it has an uniform file structure, which contains Template profile, Glue program and some Page template files. In JetPhoto Server's template package, each sub-view needs a separate glue program and a special page template. As example, for the Thumbnail sub-view, here two files are required, 'thumbnail.php' is the glue program, and 'thumbnail_template.php' is the page template.

Template profile:

Two files are included. To load the template profile is needed for some external tools, example, JetPhoto Studio's Website Manager reads the template profiles while organizing the remote site based on JetPhoto Server.

view-info.plist ----- a XML plist file to save the template information

view-preview.gif ----- a preview image for the template (200 pixels x 150

pixels)

File view-info.plist' is a property list XML file to save the information of the template. Here explains some major tags saved in a typical 'view-info.plist':

Head Tags

Key	Sample Value	Description
Name	Flash Gallery 2010	Template name
Description		Template description
Language	English	Template interface language
HasCover	1	Indicates the template contains a Cover sub-view if the value is 1, otherwise the value is 0.
HasThumbnail	1	Indicates the template contains a Thumbnail sub-view if the value is 1, otherwise the value is 0. (Thumbnail sub-view is required in any template)
HasGallery	0	Indicates the template contains a Gallery sub-view if the value is 1, otherwise the value is 0.
HasDetail	1	Indicates the template contains a Detail sub-view if the value is 1, otherwise the value is 0. (Detail sub-view is required in any template)
HasSlideshow	0	Indicates the template contains a Slideshow sub-view if the value is 1, otherwise the value is 0.
DefaultPage *	cover	Page name of the sub-view opened by clicking the album item in JetPhoto Server home page. The value of DefaultPage tag can be cover, thumbnail, gallery, detail or slideshow (cover is the default).
OpenInMainWindow *	1	Indicates the web album pages need be shown in the same window of JetPhoto Server home page if the value is 1, or shown in a new popup window if the value is 0 (0 is the default).

* added in JetPhoto Server 1.5 / Pro 2.5

Glue program:

JetPhoto Server's template is based on PHP scripting. The glue program is the entrance of all internal scripts. It provides an uniform procedure to load, process album data and represent it in web page, as the glue binding together album data and page templates with system frameworks. As example, for the Thumbnail sub-view, the below codes are saved in glue program 'thumbnail.php'.

```
$encode = "UTF-8"; // web page encoding, in normal case, it is UTF-8
$number_rows = 5; // the row numbers of the thumbnail grid
$number_cols = 6; // the column numbers of the thumbnail grid

// load album information from the specific XML plist file
// that is saved in album data package
@require("../system/frameworks/read_data.php");

// pre-process album data for thumbnail representation
@require("../system/frameworks/thumbnail_prep.php");

// represent web page using page template
@require("thumbnail_template.php");
```

Page template files:

The page template files are a set of HTML pages with embedded PHP script tags. Normally, one page template file is designed for one sub-view. For example, file 'thumbnail_template.php' is the page template of Thumbnail sub-view, and file 'slideshow_template.php' is the page template of Slideshow sub-view. The page template file is required in the assorted glue program, which loads and pre-processes the album data for the assorted page template.

Page template file is a HTML page with PHP scripts. Firstly, the designer can create or edit this page conveniently in some visualized web design tools, such as Macromedia DreamWeaver, Adobe GoLive or Microsoft FrontPage.

Once a flat HTML page is available, you should add PHP tags at some specific positions in this web page, to output some album data elements dynamically. For example, to set web page's title as the album's title, you can edit your template web

page as the below:

```
<head>
<title><?= $array_head["Title"] ?></title>
.....
```

Here is another sample to output the description tag of the first photo saved in this album:

```
<?= $array_items[1]["Description"] ?>
```

According to the two samples above, you already knew how to output a specific album's Head tag or Item tag in web page by using two system arrays `$array_head[]` and `$array_items[][]`.

Besides these two major arrays, actually, more public variables, functions that are defined in glue program or the system frameworks included, can be called in page template files. For the detail, please refer to the table below:

	Name	Scope	Description
Arrays	\$array_head	All	\$array_head[KEY] Get an album head tag with the specific KEY name.
	\$array_items	All	\$array_items[ID][KEY] Get an album item tag with the specific ID number and KEY name.
Variables	\$name	All	Get the name of current album data package
	\$encode	All	Get the text encoding of current web page. (Set this variable in glue program.)
	\$page	T,G,D,S	Get the current page number.
	\$autoplay	D	Determine whether the 'auto play' mode is on.
	\$number_rows	T,G	Get the row numbers of thumbnail grid. (Set this variable in glue program.)
	\$number_cols	T,G	Get the column numbers of thumbnail grid. (Set this variable in glue program.)
	\$number_items	T,G	Get items count in current album.
	\$number_items_perpage	T,G	Get items count in one page.
	\$number_items_lastpage	T,G	Get items count in the last page.
	\$number_items_thispage	T,G	Get items count in current page.
	\$number_pages	T,G,D	Get pages count.
	\$number_thispage	T,G,D	Get the current page number (= \$page).
\$number_firstitem	T,G	Get the first item id in current page.	
\$number_lastitem	T,G	Get the last item id in current page.	
Functions	chtml()	All	string chtml(string text) Convert special characters to HTML entities in the given text.
	cfile()	All	string cfile(string file, string type) Get the full path of the picture file with the specific filename and type. Type can be "size_0", "size_1", "size_2" or "size_full".
	page_navigator()	T,G,D	page_navigator(string previous, string next) Make a simple page navigator with the specific captions.

T=thumbnail view, G=gallery view, D=detail view, S=slideshow view, C=cover view

As a frequent asked question, lots of JetPhoto users expect to customize their own appearance template. To design the customized templates, first we don't suggest you directly change the page templates saved in the pre-installed .view packages. Instead, you should duplicate a .view package, and rename the new one as a new package. In normal case, you only need to modify the page template files (xxxx_template.php) while designing the new templates. Of course, it'll also be a good idea to make a preview image and write a short description in 'view-info.php' before you decide to deploy this new template.

Non-UTF8 Encoded Templates

To design appearance template for current JetPhoto version, we strongly suggest the template web page need to be encoded in UTF-8, whatever the language is. However, to be compatible with the early version of JetPhoto Server, the JetPhoto Server also supports the page templates with other text encoding, such as EUC-JP, GB2312 or BIG5. To enable the non-UTF8 template, firstly, you need change the \$encode setting in glue program. The original album metadata is encoded in UTF-8, it must be converted to a new copy in the target encoding before generate the web pages. JetPhoto Server uses 'iconv' to automatically convert plist files in different text encoding. iconv is a free UNIX command-line tool, that is pre-installed in most Linux distributions and Mac OS X. JetPhoto Server will save the converted album data plist files in site's cache directory (jpserver/site/cache), therefore, on UNIX-based operation system, you should also set the access mode of the cache directory as writable for your web server .

4. Configure Web Site

The folder 'site' stores web site information and cache data. There are fixed three plist files listed below:

site-info.plist ---- stores site related information, like "Title", "NeedLogin";

list.plist ---- stores all albums' listing information;

users.plist ---- stores user accounts to access this web site.

To change site-related information, you can find site-info.plist in 'site' folder, and change properties to implement different function easily. For example, you can change the "Title" property, then you browse the website, it is changed following your action. If you want to only publish your website to appointed member, you can change "NeedLogin" to "1", and add usernames and passwords to users.plist to allow them to access. However, as a JetPhoto Studio user, you usually just need to use Website Manager to do it.

5. RSS for JetPhoto Server

RSS Feed Specification

Following are the JetPhoto Server related RSS 2.0 tags:

Required tags:

title	-- Respond title of album or photo
link	-- Respond link of album or photo
description	-- Respond description of album or photo

Optional tags:

managingEditor	-- Editor of the album or photo
webMaster	-- Webmaster of the JetPhoto Server
generator	-- The RSS generator, default is JPServer with version
image	-- Specifies a GIF, JPEG or PNG image that can be displayed with the channel. The latest album cover small image.
docs	-- A URL that points to the documentation for the format used in the RSS file.
ttl	-- The time to live, it's a number of minutes that indicate how long the channel can be cached before refreshing from the source.
item	-- Content of RSS
language	-- Define language of RSS
pubDate	-- The date of the album generated
lastBuildDate	-- The updating date of the album or photo
category	-- The album belong to category
image	-- The album cover image
rating	-- The rating of the album or photo

comments -- User comments about the album or photo
copyright -- Copyright of the content in the channel

RSS for JetPhoto Server 1.0.5 Sample:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>Website Title</title>
  <link>JetPhoto server linkage URL</link>
  <description>The description about the website.</description>
  <managingEditor>The editor of site's contents.</managingEditor>
  <webMaster>Web master of the JetPhoto Server</webMaster>
  <generator>JetPhotoServer1.0.5,http://www.jetphotosoft.com</generator>
  <ttl>60</ttl>
  <image>
    <title>Title of album</title>
    <url>http://www.jpserver.com/jpserver/data/picture.jpg</url>
    <link>http://www.yoursite.com</link>
    <width>128</width>
    <height>96</height>
  </image>
  <item>
    <title>Shigongshan, Suzhou</title>
    <link>http://www.site.com/view/Classic.view/cover.php</link>
    <description>description about the album</description>
    <pubDate>Wed, 22 Sep 2004 12:35:56 +0800</pubDate>
  </item>
  <item>
    <title>Travel in Shanghai, China</title>
    <link>http://www.site.com/view/Orange.view/cover.php </link>
    <description>description about the album</description>
    <pubDate>Sun, 28 Sep 2005 10:15:50 +0800</pubDate>
  </item>
</channel>
</rss>
```