

# fpgg - Feline Picture Gallery Generator

Tommi Saviranta

wnd@iki.fi

Copyright © 2006-2007 Tommi Saviranta

## 1. Introduction

fpgg stands for *Feline Picture Gallery Generator* -- and that's exactly what it is. What makes fpgg to stand out from other similar programs? I'm not sure, but it allows me to do exactly what I want to do.

## 2. Requirements

fpgg should be pretty much happy with anything that runs Perl. In addition to Perl, some sort of command line invoked utility for resizing pictures is needed. convert from ImageMagick is a safe choice. fpgg can also invoke external application such as exiv2 to reduce file size by stripping EXIF data (e.g. thumbnail and MarkerNote) out.

## 3. Features

As a gallery generator fpgg is obviously expected to resize pictures and create (X)HTML pages. In detail, fpgg can:

- Use templates to index, picture category, and picture view page.
- Use user given information about each picture (e.g. title and description).
- Create index page and picture category index page with automatically generated thumbnails to all pictures (within that group)
- Include list of picture categories to index page
- Create view page for each picture
  - Automatically resize original picture to given size for view page
  - Include links to next and previous picture
  - Include picture information such as user given title, description.

- Include link to original picture
- Include links to categories this picture belongs to
- Use any command line invoked application to resize and otherwise process pictures.
- Use any command line invoked tool to strip unwanted EXIF data from pictures.
- Generates valid XHTML 1.0

## 4. Installation

Installing fpgg isn't really needed, unless you want to have it in your path.

## 5. Configuration

Currently there is no general configuration file for fpgg. Instead, all configuration is done through command line options and *template files*.

In addition to template files, fpgg uses *info* files to provide user given information about pictures.

## 6. Command line options

### 6.1. Basic options

`--title title`

Use given *title* as gallery title. This value will find its way to the template, and is redundant as such.

### 6.2. View picture options

`--view-size widthxheight`

Defined view picture size in pixels. View picture never exceeds *width* and *height* in picture width and height, respectively.

`--view-quality quality`

Define view picture quality. This value is given to picture resizing application. If resize application does not take quality value, this option is ignored.

`--view-convert command`

Define application to resize original picture into view picture. *command* may and should includes some of the following tags:

`%s`

Source file.

`%d`

Destination file

`%w`

Target width

`%h`

Target height

`%q`

Target quality

Do *not* place quotes around `%s` or `%d`, these are added automatically.

**Example:** `--view-convert "convert %s -resize %wx%h -quality %q %d"`

`--view-postprocess command`

Use given *command* to post-process view picture after resizing. For example this command can be used to strip unwanted EXIF-data from the pictures to reduce file size. Multiple commands can be given by separating them with a semicolon. All occurrences of `%s` are expanded with the filename. Default value uses **exiv2** to strip MakerNote and thumbnail data. Give empty string to disable post-processing view image.

## 6.3. Thumbnail options

`--thumb-size widthxheight`

See `--view-size`.

`--thumb-quality quality`

See `--view-quality`.

`--thumb-convert command`

See `--view-convert`.

`--thumb-postprocess command`

See `--view-postprocess`.

## 6.4. Miscellaneous options

`--img-basepath URI`

Define hardcoded basepath for images. This option allows to have images on external server without having to rely on `.htaccess`.

`--list-all-files`

List all files that are needed for this gallery.

`--ssh-all-files host`

Like `--list-all-files`, but prints out a full command to copy the files to remote *host* using `ssh` and `tar`. *host* can include *username* and *path* on remote host, if given in form of *user@host:path*, *host:path*, or *user@host*.

## 7. Template files

Template files are used and required to build (X)HTML pages. `fpgg` needs a template file for index, and view page, and if picture categories are defined, for category index. `fpgg` expects to find `index.template`, `view.template`, and `category.template` in work directory, respectively.

Template files usually contain valid (X)HTML base for generated pages, but may contain anything. Each recognized tag is expanded according to the tag. If unrecognized or unsupported tags are used in templates, they will also appear in final document. All tags are written as `<TAG:name:option...>` or `<*TAG:name:option...>`, where *name* is the tag name, and each *option* is separated from tag name or previous option with a colon (:). If tag name in template file is preceeded with an asterisk (\*) and the tag has no value set (by `fpgg`), the whole line will be deleted from output. For example, if picture has no map link set, the following line in template file will not be included in output at all.

```
<div class="map"><*TAG:PICTURE-MAP></div>
```

### 7.1. General tags

`TITLE`

Title given by command line option. `--title`.

`LINK-INDEX:label`

Will create a link to main index page. If given, *label* is used as link's label.

TOTAL-PICTURES

Number of pictures in whole gallery.

CREATED

Date of (X)HTML page creation.

ALL-THUMBS

Replaced with links and thumbnails to every picture in gallery. Can generate a big chunk of (X)HTML. Usually only used on an index page.

ALL-CATEGORIES

Replaced with links to each picture category. Usually only used on an index page.

## 7.2. Category index page tags

CATEGORY

Category name.

CATEGORY-THUMBS

Replaced with links and thumbnails to every picture in category. Can generate a big chunk of (X)HTML.

## 7.3. View page tags

VIEW-LINK-PREV:*label*

Link to previous picture in gallery. If given, *label* is used as link label.

VIEW-LINK-NEXT:*label*

Link to next picture in gallery. If given, *label* is used as link label.

PICTURE-FULLPATH

Full path to picture.

PICTURE-FILE

Filename of picture.

PICTURE-PATH

Path to picture (without picture file name).

PICTURE-TITLE

Picture title from user given information. See `info-file` tag `title`.

#### PICTURE-DESCRIPTION

Picture description from user given information. See `info-file` tag `text`.

#### PICTURE-MAP:*label1*:*label2*

User given link to online map. If given, *label1* is used as label for precise map, and *label2* for approximate map. See `info-file` tag `map` and `map-ap`.

#### PICTURE-CATEGORIES

Links to categories this picture belongs to. See `info-file` tag `category`.

#### PICTURE-I

Number of this picture. Always within range of 1 -- TOTAL-PICTURES.

#### PICTURE-VIEW

Replaced with the view picture.

#### PICTURE-VIEW-LINK-ORIG

Replaced with the view picture, with link to original (unprocessed) picture.

#### PICTURE-VIEW-LINK-NEXT

Replaced with the view picture, with link to next picture in gallery.

#### PICTURE-LINK-ORIGINAL:*label*

Link to original (unprocessed) picture. If given, *label* is used as link label.

## 8. `info` file tags

`info` files are used to provide information about pictures in gallery. `fpgg` tries to find `info` file in each directory with pictures. `info` file does not have to include information about every picture -- if no information is not given, corresponding template tag will remain empty.

All options in `info` file except `text` begin with tag name, followed by a colon (:), a space, and value. (E.g. `title: Example picture`.) Value must fit in one line, which can be arbitrary long. Each picture entry must begin with a `file` option, and end with an empty line, or end of file.

#### `file`

Each picture entry must begin with this option. The value of this option is simply the name of the file, which must be in the same directory as `info` file or it has no effect.

#### `title`

Title for a picture.

`category`

Lists picture categories, separated with commas (, ). Category name may contain any characters except a comma.

`map`

Link to online map.

`map-ap`

Link to approximate online map.

`text`

Full (X)HTML formatted description. Unlike other options, `text` can be more than just one line. Description is then ended with an empty line, which also ends current `file` entry, and as such, `text` should always come last after other options.

## 9. Future plans

There is no masterplan or roadmap of any kind of `fpgg`. `fpgg` evolves as author's personal needs change. Found bugs and issues will be fixed, new features may be added, but there are not plans any more specific than this.

## 10. Contact

Tommi Saviranta

`<wnd@iki.fi>`

For up to date information see <http://wnd.katei.fi/contact.html>.