# falconG Responsive Gallery generator

Version 1.3

## description

The falconG program generates a multi-language, responsive (mobile first) hierarchical WEB gallery from

1. from a special gallery structure file and a series of images.
2. any directory (folder) and its sub-directories which contain JPG, PNG, GIF, BMP image files and helper files describing the order, the title and the description of these images. It can process a JAlbum ([www.jalbum.net](http://www.jalbum.net)) generated gallery source[[1]](#footnote-1). OR

The user can specify all aspects of the galleries.

The resulting gallery may contain a virtual hierarchy of image galleries for all of the specified languages, Google Analytics code, Facebook like/share links can be added to the galleries and watermarks to the images.

## features

1. Generates a multilingual, hierarchical and embeddable responsive photo gallery with freely modifiable colors and fonts.
2. Any number of languages may be used. Generates one HTML file for each language, so the download only contains texts from one language.
3. Each album and image may have a title, which - when given - is always displayed and a description which is only shown when requested.
4. Image and gallery names are not the original file or directory names.
5. The hierarchy is virtual as each HTML files are in the same physical directory on the server and the same is true for all image and thumbnail files as well.
6. A simple index file (named *\_index.html*) is generated for each languages, but you can use your own index page and can put the gallery in an IFRAME.
7. The number of images and sub-galleries are only restricted by free disk space, although with tens of thousands of images or sub-galleries the access speed may be low (this depends on the operating system.)
8. A thumbnail image is generated for each source image. These thumbnails are smaller to make download faster and to use fewer resources. The user can specify image and thumbnail dimensions. Thumbnails are always resized, but images may be marked as not to be resized.
9. Only one copy of each image and thumbnail is stored on the server.
10. Any image may appear in any number of galleries. It may have the same title and description text in all galleries, but those can be overwritten in a sub-album.
11. A watermark may be applied to all images in a user specified font, color, transparency and position.
12. The user may set a Google analytics code for each pages
13. The galleries can link to externally created ‘About’ pages and index page.

## How to Use

You can create a gallery

1. from a hierarchical directory structure with images and special description files (e.g. from the file the program *Jalbum* creates), or
2. using a single gallery description file (*gallery.struct)* which contains all language album and image data (The source image file can be anywhere on the source machine) or
3. using only the GUI (does not work yet)

You may set the path to the **source gallery** and the **destination gallery** (which must be different from the source gallery and and may not be contained inside the source gallery tree). The content of the destination gallery is the one to upload to the server machine.

To generate a gallery for any number of languages you need either language files or an existing *gallery structure* file, which contains all of these languages.

A *gallery structure* file is an UTF-8 encoded text file with the name *<source directory name>.struct*.

If any other text files are used (e.g. generated by JAlbum) they can be either native or UTF-8 encoded. The resulting gallery structure files are always UTF-8 encoded.

An American English language file called *en.lang* is provided in the program directory and you can translate its content to any language you need[[2]](#footnote-2). The names of the language files are not important, but its extension must be ‘*.lang*’. Each language file must start with the line: *falconG Language file* followed by lines in format *name=text.* Only the *text* part should be translated. Please look up your country code on the net and please translate the copyright text too[[3]](#footnote-3). The order in which language texts are searched for is:

1. gallery structure file
2. *.lang* file(s) in source directory
3. *.lang* file(s) in destination directory
4. *.lang* file(s) in program directory

Do not use more *.lang* files then the number of languages you want to use! For instance if you only want to use German, then only put the single *de.lang* file into the source directory.

The next step depends on whether you want to use

1. A single UTF-8 coded *gallery structure* file, or
2. a hierarchical directory structure with images and optionally text files for file ordering, image/album title and description.

### Case #1

The *gallery struct* ure file are text files[[4]](#footnote-4). They must have the name of the source gallery root directory with an extension *.struct[[5]](#footnote-5)*, or it can be named as *gallery.struct.* Itmust be encoded in UTF-8. It can be created by any UTF-8 aware text editors. Do not use document editors (Word, OpenOffice, etc) to edit your files!

The *gallery.struct* file contains the common language texts and the names and paths of all albums and images. Paths can be relative to the source directory set in the GUI or can be absolute paths[[6]](#footnote-6). Any part of any line starting with the character ‘#’ is a comment and will not be used (except for the first line). Empty lines are required to separate galleries, but nowhere else.

The *gallery.struct* file contains the following sections:

Header section  
Language section  
Album section

*Header section*

starts with the file type line

*# falconG Gallery Structure file 1.3*

followed by a copyright line. Example:

*# © - András Sólyom (2019)*

followed by the source and destination directory as comments

*Language section*

contains blocks for the compulsory texts for all languages. It starts with

*[Language count:<number of languages>*

(no closing square bracket here) followed by the language blocks.

Each block starts with a line with the number of the block (1,…) at the first character position, followed by the text definition lines and ends with a ‘]’ at the first character position

Example:

[Language count:2

0

abbrev=en

name=English

icon=

images=Images

albums=Albums

toAlbums=To Albums

homePage=Home

about=About

contact=Contact

captions=Captions

share=Share

latestTitle=Sampling of Most Recent Uploads

latestDesc=This gallery contains some of the latest uploads of this site. Of course all photographs presented here can be found in their corresponding galleries too. You just have to find them...

countryCode=en\_US

countOfImages=%1 image(s) and %2 sub-album(s) in this album

falconG=Site generated by <strong>falconG</strong> - © A.Sólyom 2018

1

abbrev=hu

name=Magyarul

icon=

images=Képek

albums=Albumok

toAlbums=Albumokhoz

homePage=Főoldal

about=Rólam

contact=Kapcsolat

captions=Képaláírás

share=Megosztás

latestTitle="Kedvcsináló"

latestDesc="Ebben a galériában található néhány a legutoljára feltöltött képeim közül, amelyek természetesen a saját galériájukban is megtekinthetőek. Persze azt meg kell keresni..."

countryCode=hu\_HU

countOfImages=%1 kép és %2 album van ebben az albumban

falconG=Site generated by <strong>falconG</strong> - © A.Sólyom 2018

]

*Album section*

The structure of the album section is the same for hand made structure files and automatically generated ones, but the actual album and image definition lines may be different. These types of lines can be mixed in the same file as it is also possible to add images and or sub-albums to a generated structure file or to modify one manually outside of falconG.

What is common:

* album lines for the same level albums or sub-albums inside another album are separated from each other by an empty line.
* album hierarchy is determined by the indentation level of lines, i.e. the number of SPACE characters at front of the text[[7]](#footnote-7). Image lines and sub album lines in an album start with one space more at the front then the album line has.
* title, description and icon lines are at the same level as the album or image they belong to.
* title, description and icon lines are optional. (icon lines are only used with albums!)

Differences between *falcon* generated and manual entries *(notation: text that can/must be changed is between angled brackets <>, text literals – including parentheses - are texts outside of <>, conditional (either one of two) texts are separated by a pipe (|) character*):

* generated (G) album lines has the structure:  
  *<sub-album name>(A:<album id>)<album path relative to source| nothing>*
* manual (M) album lines  
  *<relative/path/name/of/thisAlbum>*from this the program will generate the following line:  
  *<album>(A:< ID of thisAlbum >)< relative/path/name/of >*
* image lines - G:  
  *<source image name>(<image ID,<original width>x<original height>,<resized width>x<resized height>,<upload date>, original file size)<relative path>*  
  or  
  *<!!source image name>(<image ID,<original width>x<original height>,<resized width>x<resized height>,<upload date>)<relative path>*or, when a manually entered path name was wrong   
  *<image name or path> # not found*
* image lines - M  
  *<relative/or/absolute/path/image name>|<image name w.o. path>*  
  or  
  *<relative/or/absolute/path/>!!<image name>|!!<image name w.o. path>*  
  Here the double exclamation mark signals that this image must not be resized by the program. When no path is given the path name and name of the parent album is used, unless there was a path name on the previous image in the same sub-album, in which case its path will be used.
* Title and description lines may follow any image and album. (see after the description of the icon definition line.)

The last non-image definition line after an album definition line is an icon definition line.

* generated file:  
  *[Icon:<image ID>]*
* in file edited manually by hand this line may be missing. In that case the id of the first image following the album definition line is used as an icon. In other cases:  
  *[Icon:path/of/icon/file.jpg]*

Any album or image may have a title and a description text. These are put into the gallery structure file either inside a   
 *[Title-<language code>:<text> ] or [Title-<language code>=<text> ]*  
construct or a  
 *[Descr-<language code>:<text>] or [Descr-<language code>=<text>]*construct. If any of these is given in any language then it must be given for all languages used, although the *<text>* may be left empty.

Each text will have a text ID in the gallery structure. The same text may be used in any number of places. The first time a text is found with a ‘:’ before it will be the default text for that image or album. To override a text in an album for an image use an equal sign (=) instead of the colon. If the program finds a text after a colon (:) for the same image later on that text will replace the previous default one. After saving the structure file again this changed text will only appear at the first image and all later cases it will be replaced by a ‘\*’<text ID> inside the [Text-..:…] or [Descr-..:….] definition

If the text follows an equal sign it will only be valid in that folder.

If you change any text manually in a generated file then to force a recalculation of the text ID delete the \*<text ID> from the first title/description line definition.

*In gallery structure files texts will appear in one of these forms:*

[<Title>|<Descr>-<language code #1>:|=<text in language #1>|\*]  
[Title-<language code #2>:<text in language #2>|\*]  
further lines for the other languages  
or[<Title>|<Descr>-<language code #1>:|=<text in language #1>]\*<text ID>  
[Title-<language code #2>:<text in language #2>|\*]  
further lines for the other languages  
or  
[<Title>|<Descr>-<language code #1>:\*<text ID>]  
[<Title>|<Descr>-<language code #21>:\*]  
further lines for the other languages

Examples:   
In the structure file an album named *flowers* contains image *thisImage.jpg* from the source directory *Source* in the source tree*.* It has a title and a description:

thisImage.jpg(987654321,5000x1000,500x100, 2019-07-09,1062363)Source  
[Title-en:First title of this image]\*12345  
[Title-hu:Első képcím ehhez a képhez]  
[Descr-en:First description of this image]\*678901234  
[Descr-hu:Első képleírás ehhez a képhez]

Because a colon ‘:’ is used, these texts, are the default texts for *thisImage* and will be used to any consecutive occurances for the same image, unless another text is specifically asked for. The ‘\*’ after the closing square bracket *on the first* title and decription linesignals the presence of a text ID, so falcon need not generate one. If you modify the text by hand remove the text id from the line.

Let’s suppose in album named *spring* *thisImage.jpg* appears again[[8]](#footnote-8):

*thisImage.jpg(987654321,5000x1000,500x100, 2019-07-09,1062363)Source  
[Title-en: \*]   
[Title-hu:\*]  
[Descr-en: \*]   
[Descr-hu:\*]*

The ‘\*’ character instead of a text means: use the default text for this image.

Then the same image appears in album *Best images* again with the same title, but with a different description:

*thisImage.jpg(987654321,5000x1000,500x100, 2019-07-09,1062363)Album1  
[Title-en:First title of this image]\*12345  
[Title-hu:Első képcím ehhez a képhez]  
[Descr-en=Another description of this image]\*345701234  
[Descr-hu=Másik képleírás ehhez a képhez]*

The ‘=’ sign signals that we do not redefine the default but use a new text. Now let’s suppose the same image is in a fourth album named *Lillies* with a new title and same description as in the previous folder:

*thisImage.jpg(987654321,5000x1000,500x100, 2019-07-09,1062363)Album1  
[Title-en:Second title of this image]\*12345  
[Title-hu:Második képcím ehhez a képhez]  
[Descr-en:\*345701234]   
[Descr-hu=\*]*

In this case either of ‘:’ and ‘=’ can be used as we give the ID of the description we want to appear here explicitely.

Example #1   
hand created gallery.struct file (descriptiona in *blue italics*) with two languages: English and Hungarian.:

# Album structure:

(A:1) *album ID line for root album*

[Title-en*:<title in English>*]

[Title-hu*:<title in Hungarian>*]

[Descr-en*:<descriptin in English>*] *arbitrarily long line. Put*

[Descr-hu*:<description in Hungarian>*] *\n in the text for line feed*

[Icon:<path of icon image>] *relative or absolute path name*

*relative/path/of/image.jpg first image in album w.o. a title  
or description. Notice the space at the start!*

relative/path/of/second\_image.jpg *second image in album*

[Title-en*:<title in English>*] *with a title*

[Title-hu*:<title in Hungarian>*] *but no description*

empty line before first sub-album

path/of/first/sub-album *at the same level where images*

[Title-en*:<title in English>*] *were in the parent album*

[Title-hu*:<title in Hungarian>*] *because this is also inside it*

*relative/path/of/another/image.jpg notice the 2 spaces before it!*

*relative/path/of/another*/!!notResized\_image.jpg *!! : do not resize*

In this example the name of the first sub album is *sub-album*. This can be the name of a (real) directory, but also any other name. If it is the name of a directory and the images and sub-directories inside this has no path name given, then they will be looked for in this directory. The two exclamation marks at the beginning of the image name signals that the image will not be resized.

Example #2

The same album as before but now with concrete values. The top level album has no name , the one below it have no images, just sub albums and the first sub album has two images in it. (Because of WinWord the long lines are wrapped here, but they are one line in the file):

# Album structure:

(A:1)

[Title-en:Andreas Falco Photography]

[Title-hu:Andreas Falco Photography]

Magyarorszag

[Title-en:Hungary]\*3634818861

[Title-hu:Magyarország]

[Descr-en:Hungary is a relatively small landlocked country in the Pannonian Basin, (or Carpathian Basin) in Central Europe]]

[Descr-hu:Hazánk Európa 18. legnagyobb országa]

[Icon:Magyarorszag/Gyor/7\_05720\_1\_2-HDR.jpg]

Magyarorszag/Budapest/Matyas templom/A magyar korona

[Title-en:The Hungarian crown]\*299965539

[Title-hu:A magyar korona]

[Icon: Magyarorszag/Budapest/Matyas templom/!!Cr\_4402full.jpg]

Magyarorszag/Budapest/Matyas templom/!!CrownCopy\_4402full.jpg   
 Magyarorszag/Budapest/Matyas templom/!!CrownSceptreApple\_4406.jpg

Example #3

Processed album from Exaple #2

# Album structure:

(A:1)

[Title-en:Andreas Falco Photography]\*2014183073

[Title-hu:Andreas Falco Photography]

[Icon:3114355225]

Magyarorszag(A:4131623263)

[Title-en:Hungary]\*3634818861

[Title-hu:Magyarország]

[Descr-en:Hungary is a relatively small landlocked country in the Pannonian Basin, (or Carpathian Basin) in Central Europe]]\* 1267573459

[Descr-hu:Hazánk Európa 18. legnagyobb országa]

[Icon:3127873579]

A magyar korona(A:3175693179)Magyarorszag/Budapest/Matyas templom/

[Title-en:The Hungarian crown]\*299965539

[Title-hu:A magyar korona]

[Icon:1220531141]

!!Cr\_4402full.jpg(1220531141,3504x2332,3504x2332,2019-03-10,754239)Magyarorszag/Budapest/Matyas templom/

!!CrownSceptreApple\_4406.jpg(3575104234,3128x2346,3128x2346,2019-03-10,569127)Magyarorszag/Budapest/Matyas templom/

### Case #2

Source images (possibly together with text files for file ordering, image/album title and description) are already in a hierarchical directory system. If you have downloaded the program JAlbum when it was still free for personal use you can use it to create all sub-directories and all text files. If you don’t have it just put the images into a hierarchy and do not care about the texts. When you run falconG to process a directory without the *gallery.struct* file it will create one and you can add the texts later to it[[9]](#footnote-9).

## File Structures

1. AFAIK there are no multi-language JAlbum gallery, but if you delimit your text for the different languages with two consecutive ‘@’ characters (example: @@English Text@@Dautcher texte@@Texto en alemán@@), then falconG will generate separate HTML files for all languages. (You must also have the corresponding .lang, files, or the text must be present inside the gallery structure file.) [↑](#footnote-ref-1)
2. If you plan to use different language sets to different galleries then it makes sense to put the corresponding *.lang* files into the source gallery directory. [↑](#footnote-ref-2)
3. I respectfully ask you to *not replace* the copyright text with one of your own. [↑](#footnote-ref-3)
4. And not .docx, pdf, etc or other binary files! [↑](#footnote-ref-4)
5. If you change the name of the source gallery, you must rename the structure file too, unless it is named *gallery.struct*! [↑](#footnote-ref-5)
6. On windows these starts with the drive (e.g. C:\), on linux these start with ‘/’ [↑](#footnote-ref-6)
7. Only space characters ‘ ‘ can be used here, no other whitespace characters are allowed. [↑](#footnote-ref-7)
8. If you delete the title and description lines from album *spring* they will be generated again, when the structure is next generated and saved. [↑](#footnote-ref-8)
9. Checking to checkboxes you make falcon to generate empty text holders for all languages. You only need to fill them in. [↑](#footnote-ref-9)