

Extension of diary.rozsnyo.com with PDF thumbnails

Rendering images from a PDF file

To render image files from a PDF source I started to use *ghostscript*, initially the GPL version (7.07.1-r8), which was already installed on my server with Gentoo Linux. After several testing conversions it wrote something about unsupported format, so I had to find another tool.

I discovered that there is also one other *ghostscript* ebuild in the portage tree, and it has a higher version number, so I tried to remove the old one and install the AFPL variant, version 8.15. I also noticed that this version has an Aladdin license, not a GPL-2. A quick search on Google and the difference is that the AFPL prohibits all commercial distribution.

The command for conversion of PDF into images is:

```
gs -sOutputFile="$2/%04d.png" \  
-sPAPERSIZE=a4 -sDEVICE=pngalpha -dPrinted -r144 -dDOINTERPOLATE \  
-dNOPAUSE -dBATC -dNOPROMPT -dQUIET -dNOPAGEPROMPT \  
-dTextAlphaBits=4 "$1"
```

Explanation of the components:

<code>gs</code>	ghostscript executable
<code>-sOutputFile="\$2/%04d.png"</code>	output directory is the parameter 2 of the script and the filenames will be 4 digits, filled with zeros, and with .png extension
<code>-sPAPERSIZE=a4</code>	we are supporting only A4 printing, maybe by time it will change
<code>-sDEVICE=pngalpha</code>	the printer device, pngalpha is a true color png file, pnggray will save a grayscale png file (files)
<code>-dPrinted</code>	do not print screen-only information (remarks, etc)
<code>-r144</code>	resolution of 144 dpi (as double of screen defaults 72)
<code>-dDOINTERPOLATE</code>	turns on the interpolation of images, -dNOITERPOLATE will turn it off
<code>-dNOPAUSE</code> <code>-dBATC</code> <code>-dNOPROMPT</code> <code>-dQUIET</code> <code>-dNOPAGEPROMPT</code> <code>-dTextAlphaBits=4</code>	few settings to ensure non-interactive mode
<code>"\$1"</code>	the level of smoothing of the text, not quite sure if it means 4 bits (16 level) or 4x4 sub-sampling (but it is in that case the same, because 4x4 = 16 dots, 16 different levels)
<code>"\$1"</code>	the file name taken from the first parameter of the script

Making thumbnails

The thumbnails are made by the *convert* program from the *ImageMagick* package. The resizing script is the following:

```
cd "$2"  
for i in `ls *.png`  
do  
  f=`echo "$i" | cut -c1-4`  
  convert +matte -geometry '999x100' "$f.png" "$f.gif"  
done
```

Options:

<code>+matte</code>	discards the alpha channel
<code>-geometry '999x100'</code>	constant height of 100 pixels

Problems

Rendering

The main problem with *ghostscript* is that on one PDF file it ends with a segmentation fault when the image interpolation is turned on. I will have to wait for a new release, but until that time, I made a workaround for as much as possible correct rendering:

- render small files (10 dpi) without interpolation
- count the small files and then delete them
- render with interpolation at full resolution (144dpi)
- when the program ends, check the file count, if it differs, move the already rendered images to a temporary location, render the PDF again without interpolation and then overwrite the files with the ones with interpolation
- resize the images to thumbnails as in a normal case

File format

Because I tried to avoid unnecessary format conversions, the rendered files are directly in PNG format. The grayscale (pnggray device) output is without problem, but the true color variant (pngalpha device) contains an opacity mask, which is not really needed.

Color vs. grayscale

There is no option at *ghostscript* to save black and white pages as grayscale images and color pages as color images. I will have to write an output compressor, which will choose the best format (grayscale PNG or true color PNG) for the page, but for now, the generated images and thumbnails are in color.

The result

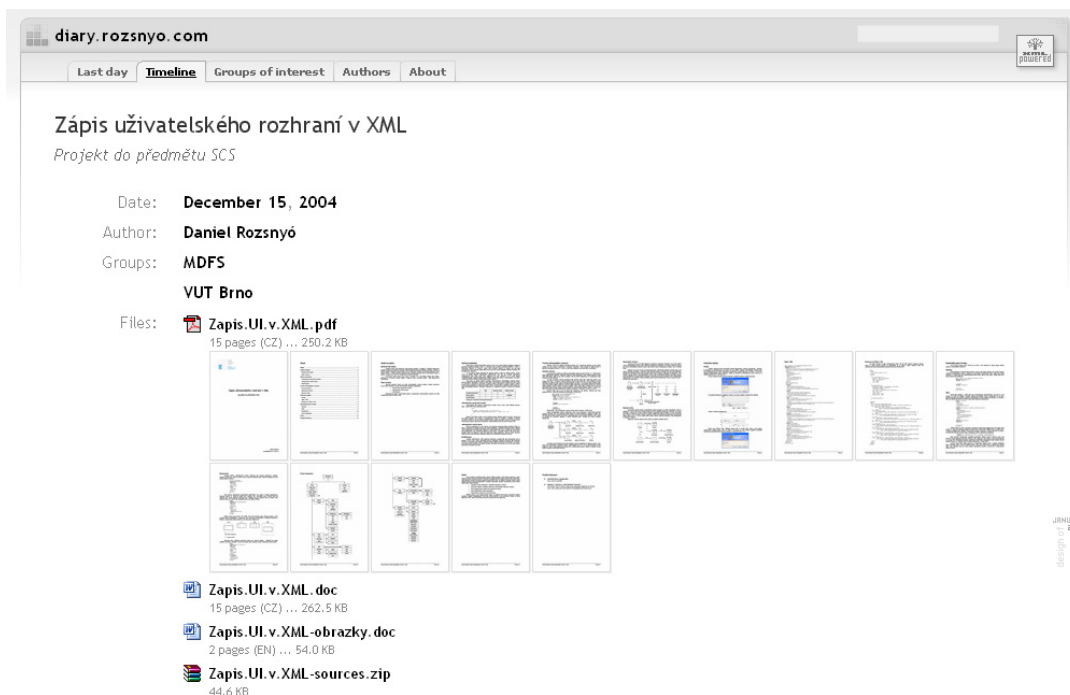


Fig. 1: Final result

Bibliography

How to use Ghostscript

- <http://www.cs.wisc.edu/~ghost/doc/AFPL/8.00/Use.htm>