Ghostscript and MuPDF Status OpenPrinting Summit April 2019

Michael Vrhel, Ph.D.

Artifex Software Inc.

Novato CA

Artifex

Outline

Ghostscript

Ghostscript is a document conversion and rendering engine.

Converts between and renders PDF, PS, PCL, PCL-XL, XPS

Dual license GNU AGPLv3 / Commercial

Source and documentation available at www.ghostscript.com

MuPDF

Open source software framework for viewing and converting PDF, XPS, and e-book documents.

Designed toward mobile environment use.

Dual license GNU AGPLv3 / Commercial

Source and documentation available at www.mupdf.com

Artifex

MuPDF

Command Line Tools

mutool draw

Primarily used for rendering a document to image files.

mutool convert

For converting documents into other formats.

mutool trace

Debugging tool used for printing a trace of the graphics device calls on a page.

MuPDF

Command Line Tools

mutool show

A tool for displaying the internal objects in a PDF file.

mutool extract

Extract images and embedded font resources.

mutool clean

Rewrite PDF file. Used to fix broken files, or to make a PDF file human editable.

MuPDF

Command Line Tools

mutool merge

Merge pages from multiple input files into a new PDF.

mutool create

Create a new PDF file from a text file with graphics commands.

mutool run

A tool for running Javascript programs with access to the MuPDF library functions.

MuPDF

JavaScript

Examples in docs/examples:

pdf-merge.js

pdf-portfolio.js

pdf-create.js

and more…

Example:

mutool run pdf-merge.js output.pdf input1.pdf input2.pdf …

Changes to GS since last meeting

Release 9.24 September 3rd 2018

- Primarily a security release for several real and potential exploits.

Release 9.25 September 13th 2018

- Additional security fixes

Release 9.26 November 11th 2018

- Additional security fixes

Fixes malicious PostScript exploits found by:
Man Yue Mo of Semmle Security Research Team
Jens Müller of Ruhr-Universität Bochum
Tavis Ormandy of Google's Project Zero

Changes to GS since last meeting

Release 9.27

This release involves making a large number of non-standard operators only intended for internal use inaccessible to "user" Postscript jobs.

Tools that use heavily non-standard Postscript may be affected by this.

If, in your testing, you find such tools, we'd appreciate it if you could encourage the tools author to contact us to the discuss the issue (the best places are #ghostscript on freenode or the gs-devel list).

Changes to GS since last meeting

Release 9.27

- Addition of psdcmyk16 device. A 16-bit full spot color supported output device (Photoshop format *.psd).

- Support for 3rd party plug-ins in LittleCMS fork.

- Support for j2k enumerated color spaces of e-sRGB and ROMMRGB.

- Fix for TRC in default gray color space (to gamma of 2.2)

- Fix for cups use of RGBW color space. More about this later.

Changes to MuPDF since last meeting

Release 1.13: (April 20th 2018)

- New "mutool sign" tool for showing and verifying digital signatures.

- Chinese, Japanese, Korean, Cyrillic, and Greek font support in mutool create.

- Improvements to annotation editing API.

Artifex

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

Annotation editing:

- New annotation editing mode in mupdf-gl.
- Can create, edit, and delete most annotation types.
- Can create appearance streams for most annotation types.
- Can create appearance streams for Tx form fields.
- Can create appearance streams for Ch form fields.

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

Form filling in mupdf-gl:
- Can click buttons, checkboxes, and radioboxes.
- Can fill out text fields using dialog box.
- Can select choice options using dialog box.
- Can verify and sign digital signatures.

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

Improved UI for mupdf-gl:
- Password dialog.
- Error dialog.
- Open/save file dialog.
- Snap selection to words or lines by holding control or control+shift.
- Save and restore current location, bookmarks, and navigation history.

Artifex

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

Build system simplifications:
- Auto-generated CMap, ICC, and JS source files are checked in to git.
- Embedded CMap resources are now generated by a python script.
- Embedded font resources are linked directly if using GNU ld or windows.
- Namegen tool replaced by use of C macros.
- Simplified Makefile.

Artifex

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

API changes:
- Pass rectangle and matrix structs by value.
- Replaced PDF_NAME_Xxx macros with PDF_NAME(Xxx).
- Added PDF_TRUE, PDF_FALSE, and PDF_NULL constant pdf_obj* macros.
- Added helper functions: pdf_dict_get_int, etc.
- Removed 'doc' argument in pdf_new_int, etc.
- Quads instead of rects when highlighting and searching text.
- mutool run: Pass arguments to script in scriptArgs global.

Changes to MuPDF since last meeting

Release 1.14: (October 4th 2018)

New features:

- Added "Source Han Serif" CJK fallback font.
- Added more scripts to the Noto fallback fonts.
- Multi-page PNM support.
- "mutool show" now supports a path syntax for selecting objects to show.

Artifex

Little CMS2MT

We continue to use a fork of Little CMS2 that is thread safe.

Fork is currently available with git checkout of Ghostscript.

We bring in any bug fixes applied to Little CMS2 .

Possible to make use of 3rd party plug-ins.

Developed SSE4.2 and AVX2 plug-in for tetrahedral interpolation with cmyk, rgb, or gray output. (Commercial license only)
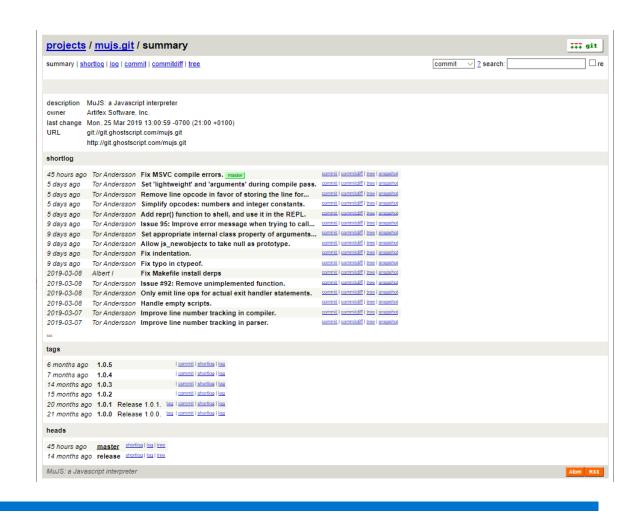
Artifex

MuJS

Lightweight Javascript interpreter.

Implements EMCAScript ECMA-262

Open source under ISC license
https://opensource.org/licenses/ISC

http://git.ghostscript.com/?p=mujs.git;a=summary

MuPDF WebAssembly

Binary instruction format for a stack-based virtual machine.

Wasm is designed as a portable target for compilation enabling
deployment on the web for client and server applications.

```
mupdf.openDocument(filename)    Open a document and return a handle.
mupdf.freeDocument(doc)         Free a document and its associated resources.
mupdf.documentTitle(doc)        Return the document title as a string.
mupdf.documentOutline(doc)      Return element containing the table of contents formatted as an unordered HTML list with links to pages
mupdf.countPages(doc)           Return the number of pages in the document.
mupdf.pageWidth(doc, page, dpi)         Return the width of a page.
mupdf.pageHeight(doc, page, dpi)        Return the height of a page.
mupdf.drawPageAsPNG(doc, page, dpi)  Render the page and return a PNG image formatted as a data URI.
mupdf.pageLinks(doc, page, dpi)         Retrieve an HTML string describing the links on a page.
mupdf.drawPageAsSVG(doc, page)          Return a string with the contents of the page in SVG format.
mupdf.drawPageAsHTML(doc, page)         Return a string with the contents of the page in HTML format, using absolute positioned elements.
```

Artifex

Current/Future Work

New PDF interpreter

More security updates

Language switching build ghostpdl (PCL, XPS, PDF, PS)

16-bit support in Transparency compositor

SIMD accelerations (commercial customers)

Improved Java bindings for MuPDF

Signature and Forms improvements in MuPDF

Javascript manipulations for MuPDF (using mujs)

CUPS Device

Ended up being related to use of RGBW color space.   Currently the cups device treats this as an additive RGB color space with an additional channel. The fix required

+   cups->color_info.depth = 4 * cups->header.cupsBitsPerColor;

+   cups->color_info.num_components = 3;

+ if (cups->header.cupsColorSpace == CUPS_CSPACE_RGBW) {

+   ci = (ci << shift) | cups->EncodeLUT[gx_max_color_value];

CUPS Device

Color space mapping for cups is odd with these color spaces:

For RGBW color space, a source CMYK value is mapped to RGB, since GS sees an RGB device

CUPS device then maps the RGB to CMYK and then to RGBW and finally it does an encode

Unfortunately, the number of color values for RGBW is 3 since it is using an RGB ICC profile
this means that the W mapping value from cups is lost in cmap_rgb_direct

Questions:
Should RGBW be a 4 color inverted CMYK device with a CMYK ICC profile?

Any interest in moving some of the 4+ color cups device to planar memory format in GS?

Much easier to deal with (supports 64 colors) and for which we have examples in
tiffsep, psdcmyk and psdcmyk16

Code Security/Analysis Methods

Fuzzing of test files used to detect simple faults

Coverity : https://scan.coverity.com/projects/ghostpdl (dereferences of NULL pointers, use of uninitialized data,

memory corruptions, buffer overruns, control flow issues, incorrect expressions, unsafe signed values)

Coverage tests run periodically: https://ghostscript.com/coverage/

Various compilers used and warning report provided with every commit (gcc, clang)

Address Sanitizer: Testing for buffer overflows, dangling pointer overflows

Valgrind: Testing for buffer overflows, use of uninitialized memory/variables

Memento: Memory leak/corruption analyzer and "Memory Squeezing".   Part of Ghostscript build memento.h/c

More Information

Repositories located at
> git://git.ghostscript.com

Ghostscript discussions on IRC freenode #ghostscript channel
MuPDF discussions on IRC freenode #mupdf channel

Bug reports
> bugs.ghostscript.com

Additional information at www.mupdf.com www.ghostscript.com