

# Ghostscript and MuPDF Status OpenPrinting Summit May 2022

Michael Vrhel, Ph.D.  
Artifex Software Inc.  
Novato CA



# Outline



Ghostscript

MuPDF

Ghostscript – Changes since last meeting

MuPDF – Changes since last meeting

PyMuPDF, MuJS, Web Assembly

LittleCMS2MT, GS API language bindings

Code security

Current and future work

# Ghostscript



Ghostscript is a document conversion and rendering engine.

Ghostscript, GhostPCL, GhostXPS, GhostPDF, GhostPDL

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

Dual license GNU AGPLv3 / Commercial

Source and documentation available at [www.ghostscript.com](http://www.ghostscript.com)

# MuPDF



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

Dual license GNU AGPLv3 / Commercial

Written in C, but has JNI bindings that work on both Oracle's Java and Android

Source and documentation available at [www.mupdf.com](http://www.mupdf.com)

# 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.

## 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 9.55 October 2021

- Includes new PDF interpreter. Invoked with `-dNEWPDF=true`
- Includes stand-alone PDF only binary
- pdfwrite device supports passthrough for JPX/JPG2000
- Improvements in demo apps for C, C#, Java, and Python
- Object specific halftoning

# Changes to GS since last meeting



## Release 9.56 March 2022

- New PDF interpreter enabled by default (`-dNEWPDF=false` to disable)
- Calling Ghostscript via GS API now threadsafe. Limitation is the X11 devices which cannot be made threadsafe due to interaction with X11 server.
- PSD (Photoshop) output format includes ICC profile
- API demo added for MATLAB

# Changes to MuPDF since last meeting

Release 1.19 October 5<sup>th</sup> 2021



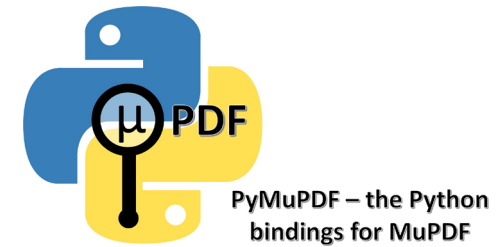
Microsoft Word DOCX output format  
ODT output format (OpenOffice)

Undo and redo functions  
Autosave and restore functions  
Reflow document wrapper

Enhancements to OCR functions  
Additional 'mutool run' and Java library functions  
Added new Unicode scripts and fonts for EPUB  
Support opacity for all markup annotation types

C++ and Python bindings now run on Windows  
Python bindings are now available as pre-built binaries  
Added demo PDF viewer using the Python MuPDF bindings and PyQt

# PyMuPDF



Python binding for MuPDF

PyMuPDF can be installed from Python wheels

Windows (32bit and 64bit), Linux (64bit, Intel and ARM) and Mac OSX (64bit, Intel), for Python versions 3.7 and later.

See <https://pymupdf.readthedocs.io/en/latest/intro.html> for detailed documentation

Heavily used in the Python community (downloaded more than 19 million times)

Dual license GNU AGPLv3 / Commercial

# 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 .

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

# MuJS



MuJS is a library, written in C.  
MuJS has no notion of a main program:  
it only works embedded in a host client program.  
The host program can invoke functions to execute  
Javascript code, read and write Javascript variables,  
and register C functions to be called by Javascript.

Implements ECMAScript ECMA-262

Open source under ISC license

<https://opensource.org/licenses/ISC>

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

The screenshot shows a web browser window with the URL `git.ghostscript.com/?p=mujs.git;a=summary`. The page title is "projects / mujs.git / summary". Below the title, there are links for "summary", "shortlog", "log", "commit", "commitdiff", and "tree". The main content area displays the following information:

- description: MuJS: a Javascript interpreter
- owner: Artifex Software, Inc.
- last change: Wed, 16 Mar 2022 10:34:18 +0000 (11:34 +0100)
- URL: `git://git.ghostscript.com/mujs.git` and `http://git.ghostscript.com/mujs.git`

Below this information is a "shortlog" section showing a list of recent commits:

Date	Author	Commit Message	Branch
2022-03-16	Tor Andersson	Bug 705052: Don't use private STACK/TOP macros in jsstate.c	master
2022-03-10	Tor Andersson	Add "console" object to mujs shell.	
2022-03-10	Tor Andersson	Issue #156: Fix check for duplicate formal parameters...	
2022-02-23	Tor Andersson	Some minor optimizations to Ap_join.	
2022-02-16	Avi Halachmi...	array join: avoid strcat, speedup from O(N^2) to O(N)	
2021-12-08	Tor Andersson	Add JS_VERSION_MAJOR/MINOR/PATCH defines to mujs.h	1.2.0
2021-12-08	Tor Andersson	Bug 704756: Don't trust function.length property!	
2021-12-06	Tor Andersson	Minor cleanups.	
2021-12-06	Tor Andersson	Bug 704748: Save original object in stack slot for...	

# MuPDF WebAssembly

Binary instruction format for a stack-based virtual machine.



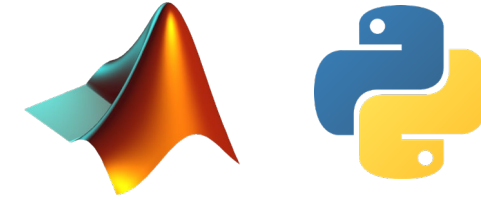
Wasm is designed as a portable target for compilation enabling deployment on the web for client and server applications.

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

# Ghostscript API Bindings



In ghostpd/demos folder:



C – Contains a VS project that exercises API

C# – Contains simple demo viewer for Windows (WPF UI) and for Linux (MONO with GTK UI). Mimics C API but has helper methods to extend API

Java – Contains simple Java demo viewer.  
Mimics C API but has helper methods to extend API

Python – Mimics C API. Has demo example usage (ghostpd\demos\python\examples.py)

MATLAB – Uses MEX file to interface to GS API.  
Allows MATLAB function calls like:

```
page_image = gs_displaydevice(input_file, PlanarRGB, page_number, resolution);
```



# Code Security/Analysis Methods



Fuzzing of test files used to detect simple faults

Coverity : <https://scan.coverity.com/projects/ghostpd> (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

# Current/Future Work



`-dBlackVector -dBlackText -dBlackThresholdL -dBlackThresholdC`

These options allow the forcing of vector and text content to black (or white) even to pdfwrite device.

Improved page control for PDF files like what is available for XPS files. E.g.

`-sPageList=1,2,1,2`

`-sPageList=10-5`

`-sPageList=1-,-1`

Preservation of PDF accessibility content for pdfwrite device

# More Information



Repositories located at  
[git://git.ghostscript.com](https://git.ghostscript.com)

Ghostscript discussions on Discord channel #ghostscript  
MuPDF discussions on Discord channel #mupdf

Bug reports  
[bugs.ghostscript.com](https://bugs.ghostscript.com)

Additional information at [www.mupdf.com](http://www.mupdf.com) [www.ghostscript.com](http://www.ghostscript.com)