Coherent PDF Library (libcpdf)

Developer's Manual

Version 1.8 (December 2013)

Coherent Graphics Ltd

For bug reports, feature requests and comments, email
contact@coherentgraphics.co.uk

Contents

Notes

The chapters are numbered to be the same as those in the manual for the Coherent PDF Command Line Tools (cpdfmanual.pdf).

Installation

The Coherent PDF Library is provided either in compiled form (the archive file `libcpdf.a` and the library header `cpdflibwrapper.h`), or as source code. Instructions for building from source are included in the distribution.

Place `libcpdf.a` somewhere suitable. Instruct your C linker to link with it. Place `cpdflibwrapper.h` somewhere suitable and instuct your C compiler to search for headers there. The library is now ready for use.

1 Preliminaries

```
/* The function cpdf_startup must be called with argv before using the
library. */
void cpdf_startup (char **);

/* Set demo mode. Upon library startup is false. If set, files written will
 * have the text DEMO stamped over each page. This stamping will also slow down
 * the library significantly. */
void cpdf_setDemo(int);

/* Errors. lastError and lastErrorString hold information about the last error
 * to have occurred. They should be consulted after each call. If
 * cpdf_lastError is non-zero, there was an error, and cpdf_lastErrorString
 * gives details. If cpdf_lastError is zero, there was no error on the most
 * recent cpdf call. */
int cpdf_lastError;
char* cpdf_lastErrorString;

/* Clear the current error state. */
void cpdf_clearError (void);

/* A debug function which prints some information about resource usage. This
 * can be used to detect if PDFs or ranges are being deallocated properly. */
void cpdf_onExit (void);

/* Remove a PDF from memory, given its number. */
void cpdf_deletePdf(int);

/* Calling replacePdf(a, b) places PDF b under number a. Original a and b are
 * no longer available. */
void cpdf_replacePdf(int, int);

/* To enumerate the list of currently allocated PDFs, call
 * cpdf_startEnumeratePDFs which gives the number, n, of PDFs allocated, then
 * cpdf_enumeratePDFsInfo and cpdf_enumeratePDFsKey with index numbers from
 * 0...(n - 1). Call cpdf_endEnumeratePDFs to clean up. */
int cpdf_startEnumeratePDFs(void);
int cpdf_enumeratePDFsKey(int);
char* cpdf_enumeratePDFsInfo(int);
void cpdf_endEnumeratePDFs(void);
```

2 Basics

3 Merging and Splitting

4 Pages

5 Encryption

6 Compression

7 Bookmarks

8 Presentations

9 Logos, Watermarks and Stamps

10 Multipage Facilities

11 Document Information and Metadata

12 File Attachments

13 Miscellaneous

14 Page Labels

15 Special functionality 1. – Encryption and Permission status

16 Special functionality 2. – Undo

A Dates

Dates in PDF are specified according to the following format:

D:YYYYMMDDHHmmSSOHH'mm'

where:

- YYYY is the year;
- MM is the month;
- DD is the day (01-31);
- HH is the hour (00-23);
- mm is the minute (00-59);
- SS is the second (00-59);
- O is the relationship of local time to Universal Time (UT), denoted by '+', '-' or 'Z';
- HH is the absolute value of the offset from UT in hours (00-23);
- mm is the absolute value of the offset from UT in minutes (00-59).

A contiguous prefix of the parts above can be used instead, for lower accuracy dates. For example:

D:2011 (2011)
D:20110103 (3rd March 2011)
D:201101031854-08'00' (3rd March 2011, 6:54PM, US Pacific Standard Time)

B Example Program in C

This program loads a file from disk, adds page numbers, and then writes the file encrypted to disk.