

A PDFlib Quick Reference

General Functions

Function prototype	page
<code>void PDF_boot(void)</code>	133
<code>void PDF_shutdown(void)</code>	133
<code>PDFlib_api * PDF_boot_dll(void)</code>	133
<code>void PDF_shutdown_dll(PDFlib_api *PDFlib)</code>	133
<code>PDF *PDF_new(void)</code>	133
<code>PDF *PDF_new2(void (*errorhandler)(PDF *p, int errortype, const char *msg), void* (*allocproc)(PDF *p, size_t size, const char *caller), void* (*reallocproc)(PDF *p, void *mem, size_t size, const char *caller), void (*freeproc)(PDF *p, void *mem), void *opaque)</code>	134
<code>void PDF_delete(PDF *p)</code>	135
<code>int PDF_open_file(PDF *p, const char *filename)</code>	135
<code>int PDF_open_fp(PDF *p, FILE *fp)</code>	136
<code>void PDF_open_mem(PDF *p, size_t (*writeproc)(PDF *p, void *data, size_t size))</code>	136
<code>const char * PDF_get_buffer(PDF *p, long *size)</code>	137
<code>void PDF_close(PDF *p)</code>	137
<code>void PDF_begin_page(PDF *p, float width, float height)</code>	138
<code>void PDF_end_page(PDF *p)</code>	138
<code>float PDF_get_value(PDF *p, const char *key, float modifier)</code>	138
<code>void PDF_set_value(PDF *p, const char *key, float value)</code>	139
<code>const char * PDF_get_parameter(PDF *p, const char *key, float modifier)</code>	139
<code>void PDF_set_parameter(PDF *p, const char *key, const char *value)</code>	139
<code>void PDF_create_pvf(PDF *p, const char *filename, int reserved, const void *data, size_t size, const char *optlist)</code>	139
<code>int PDF_delete_pvf(PDF *p, const char *filename, int reserved)</code>	140
<code>int PDF_get_errnum(PDF *p)</code>	141
<code>const char *PDF_get_errmsg(PDF *p)</code>	141
<code>const char *PDF_get_apiname(PDF *p)</code>	141
<code>void *PDF_get_opaque(PDF *p)</code>	141

Font Functions

Function prototype	page
<code>int PDF_load_font(PDF *p, const char *fontname, int reserved, const char *encoding, const char *optlist)</code>	143
<code>void PDF_setfont(PDF *p, int font, float fontsize)</code>	145
<code>void PDF_begin_font(PDF *p, char *fontname, int reserved, float a, float b, float c, float d, float e, float f, const char *optlist)</code>	145
<code>void PDF_end_font(PDF *p)</code>	146
<code>void PDF_begin_glyph(PDF *p, char *glyphname, float wx, float llx, float lly, float urx, float ury)</code>	146
<code>void PDF_end_glyph(PDF *p)</code>	146
<code>void PDF_encoding_set_char(PDF *p, const char *encoding, int slot, const char *glyphname, int uv)</code>	146

Text Output Functions

Function prototype	page
<code>void PDF_set_text_pos(PDF *p, float x, float y)</code>	147
<code>void PDF_show(PDF *p, const char *text)</code>	147
<code>void PDF_show_xy(PDF *p, const char *text, float x, float y)</code>	149
<code>void PDF_continue_text(PDF *p, const char *text)</code>	149
<code>void PDF_fit_textline(PDF *p, const char *text, int len, float x, float y, const char *optlist);</code>	150
<code>int PDF_show_boxed(PDF *p, const char *text, float x, float y, float width, float height, const char *mode, const char *feature)</code>	152
<code>float PDF_stringwidth(PDF *p, const char *text, int font, float fontsize)</code>	153

Graphics Functions

Function prototype	page
<code>void PDF_setdash(PDF *p, float b, float w)</code>	154
<code>void PDF_setdashpattern(PDF *p, const char *optlist)</code>	154
<code>void PDF_setflat(PDF *p, float flatness)</code>	154
<code>void PDF_setlinejoin(PDF *p, int linejoin)</code>	155
<code>void PDF_setlinecap(PDF *p, int linecap)</code>	155
<code>void PDF_setmiterlimit(PDF *p, float miter)</code>	156
<code>void PDF_setlinewidth(PDF *p, float width)</code>	156
<code>void PDF_initgraphics(PDF *p)</code>	156
<code>void PDF_save(PDF *p)</code>	156
<code>void PDF_restore(PDF *p)</code>	157
<code>void PDF_translate(PDF *p, float tx, float ty)</code>	157
<code>void PDF_scale(PDF *p, float sx, float sy)</code>	157
<code>void PDF_skew(PDF *p, float alpha, float beta)</code>	158
<code>void PDF_concat(PDF *p, float a, float b, float c, float d, float e, float f)</code>	158
<code>void PDF_setmatrix(PDF *p, float a, float b, float c, float d, float e, float f)</code>	159
<code>int PDF_create_gstate(PDF *p, const char *optlist)</code>	159
<code>void PDF_set_gstate(PDF *p, int gstate)</code>	160
<code>void PDF_moveto(PDF *p, float x, float y)</code>	160
<code>void PDF_lineto(PDF *p, float x, float y)</code>	161
<code>void PDF_curveto(PDF *p, float x1, float y1, float x2, float y2, float x3, float y3)</code>	161
<code>void PDF_circle(PDF *p, float x, float y, float r)</code>	161
<code>void PDF_arc(PDF *p, float x, float y, float r, float alpha, float beta)</code>	161
<code>void PDF_arcn(PDF *p, float x, float y, float r, float alpha, float beta)</code>	162
<code>void PDF_rect(PDF *p, float x, float y, float width, float height)</code>	162
<code>void PDF_closepath(PDF *p)</code>	162
<code>void PDF_stroke(PDF *p)</code>	163
<code>void PDF_closepath_stroke(PDF *p)</code>	163
<code>void PDF_fill(PDF *p)</code>	163
<code>void PDF_fill_stroke(PDF *p)</code>	164
<code>void PDF_closepath_fill_stroke(PDF *p)</code>	164

Function prototype	page
<code>void PDF_clip(PDF *p)</code>	164
<code>void PDF_endpath(PDF *p)</code>	164

Color Functions

Function prototype	page
<code>void PDF_setcolor(PDF *p, const char *fstype, const char *colorspace, float c1, float c2, float c3, float c4)</code>	165
<code>int PDF_makespotcolor(PDF *p, const char *spotname, int reserved)</code>	166
<code>int PDF_load_iccprofile(PDF *p, const char *filename, int reserved, const char *optlist)</code>	166
<code>int PDF_begin_pattern(PDF *p, float width, float height, float xstep, float ystep, int painttype)</code>	168
<code>void PDF_end_pattern(PDF *p)</code>	169
<code>int PDF_shading_pattern(PDF *p, int shading, const char *optlist)</code>	169
<code>void PDF_shfill(PDF *p, int shading)</code>	169
<code>int PDF_shading(PDF *p, const char *shtype, float xo, float yo, float x1, float y1, float c1, float c2, float c3, float c4, const char *optlist)</code>	169

Image Functions

Function prototype	page
<code>int PDF_load_image(PDF *p, const char *imagetype, const char *filename, int reserved, const char *optlist)</code>	171
<code>void PDF_close_image(PDF *p, int image)</code>	172
<code>void PDF_fit_image(PDF *p, int im, float x, float y, const char *optlist)</code>	174
<code>int PDF_begin_template(PDF *p, float width, float height)</code>	176
<code>void PDF_end_template(PDF *p)</code>	176

PDF Import (PDI) Functions

Function prototype	page
<code>int PDF_open_pdi_callback(PDF *p, void *opaque, size_t filesize, size_t (*readproc)(void *opaque, void *buffer, size_t size), int (*seekproc)(void *opaque, long offset), const char *optlist)</code>	177
<code>int PDF_open_pdi_callback(PDF *p, void *opaque, size_t filesize, size_t (*readproc)(void *opaque, void *buffer, size_t size), int (*seekproc)(void *opaque, long offset), const char *optlist)</code>	177
<code>void PDF_close_pdi(PDF *p, int doc)</code>	178
<code>int PDF_process_pdi(PDF *p, int doc, int page, const char *optlist)</code>	180
<code>void PDF_close_pdi_page(PDF *p, int page)</code>	179
<code>void PDF_fit_pdi_page(PDF *p, int page, float x, float y, const char *optlist)</code>	179
<code>int PDF_process_pdi(PDF *p, int doc, int page, const char *optlist)</code>	180
<code>float PDF_get_pdi_value(PDF *p, const char *key, int doc, int page, int reserved)</code>	181
<code>const char *PDF_get_pdi_parameter(PDF *p, const char *key, int doc, int page, int reserved, int *len)</code>	181

Block Filling Functions

<i>Function prototype</i>	<i>page</i>
<i>int PDF_fill_textblock(PDF *p, int page, const char *blockname, const char *text, int len, const char *optlist)</i>	183
<i>int PDF_fill_imageblock(PDF *p, int page, const char *blockname, int image, const char *optlist)</i>	183
<i>int PDF_fill_pdfblock(PDF *p, int page, const char *blockname, int contents, const char *optlist)</i>	184

Hypertext Functions

<i>Function prototype</i>	<i>page</i>
<i>int PDF_add_bookmark(PDF *p, const char *text, int parent, int open)</i>	187
<i>void PDF_set_info(PDF *p, const char *key, const char *value)</i>	188
<i>void PDF_attach_file(PDF *p, float llx, float lly, float urx, float ury, const char *filename, const char *description, const char *author, const char *mimetype, const char *icon)</i>	189
<i>void PDF_add_note(PDF *p, float llx, float lly, float urx, float ury, const char *contents, const char *title, const char *icon, int open)</i>	190
<i>void PDF_add_pdflink(PDF *p, float llx, float lly, float urx, float ury, const char *filename, int page, const char *optlist)</i>	192
<i>void PDF_add_locallink(PDF *p, float llx, float lly, float urx, float ury, int page, const char *optlist)</i>	193
<i>void PDF_add_launchlink(PDF *p, float llx, float lly, float urx, float ury, const char *filename)</i>	193
<i>void PDF_add_weblink(PDF *p, float llx, float lly, float urx, float ury, const char *url)</i>	194
<i>void PDF_set_border_style(PDF *p, const char *style, float width)</i>	194
<i>void PDF_set_border_color(PDF *p, float red, float green, float blue)</i>	194
<i>void PDF_set_border_dash(PDF *p, float b, float w)</i>	194
<i>void PDF_add_nameddest(PDF *p, const char *name, int reserved, const char *optlist)</i>	195
<i>void PDF_add_thumbnail(PDF *p, int image)</i>	195