

NAME

xpdfrc – configuration file for Xpdf tools (version 3.04)

DESCRIPTION

All of the Xpdf tools read a single configuration file. If you have a *.xpdfrc* file in your home directory, it will be read. Otherwise, a system-wide configuration file will be read from */usr/local/etc/xpdfrc*, if it exists. (This is its default location; depending on build options, it may be placed elsewhere.) On Win32 systems, the *xpdfrc* file should be placed in the same directory as the executables.

The xpdfrc file consists of a series of configuration options, one per line. Blank lines and lines starting with a '#' (comments) are ignored.

Arguments may be quoted, using "double-quote" characters, e.g., for file names that contain spaces.

The following sections list all of the configuration options, sorted into functional groups. There is an examples section at the end.

INCLUDE FILES**include** *config-file*

Includes the specified config file. The effect of this is equivalent to inserting the contents of *config-file* directly into the parent config file in place of the *include* command. Config files can be nested arbitrarily deeply.

CHARACTER MAPPING**nameToUnicode** *map-file*

Specifies a file with the mapping from character names to Unicode. This is used to handle PDF fonts that have valid encodings but no ToUnicode entry. Each line of a nameToUnicode file looks like this:

hex-string name

The *hex-string* is the Unicode (UCS-2) character index, and *name* is the corresponding character name. Multiple nameToUnicode files can be used; if a character name is given more than once, the code in the last specified file is used. There is a built-in default nameToUnicode table with all of Adobe's standard character names.

cidToUnicode *registry-ordering map-file*

Specifies the file with the mapping from character collection to Unicode. Each line of a cidToUnicode file represents one character:

hex-string

The *hex-string* is the Unicode (UCS-2) index for that character. The first line maps CID 0, the second line CID 1, etc. File size is determined by size of the character collection. Only one file is allowed per character collection; the last specified file is used. There are no built-in cidToUnicode mappings.

unicodeToUnicode *font-name-substring map-file*

This is used to work around PDF fonts which have incorrect Unicode information. It specifies a file which maps from the given (incorrect) Unicode indexes to the correct ones. The mapping will be used for any font whose name contains *font-name-substring*. Each line of a unicodeToUnicode file represents one Unicode character:

in-hex out-hex1 out-hex2 ...

The *in-hex* field is an input (incorrect) Unicode index, and the rest of the fields are one or more output (correct) Unicode indexes. Each occurrence of *in-hex* will be converted to the specified output sequence.

unicodeMap *encoding-name map-file*

Specifies the file with mapping from Unicode to *encoding-name*. These encodings are used for text output (see below). Each line of a unicodeMap file represents a range of one or more Unicode characters which maps linearly to a range in the output encoding:

in-start-hex in-end-hex out-start-hex

Entries for single characters can be abbreviated to:

in-hex out-hex

The *in-start-hex* and *in-end-hex* fields (or the single *in-hex* field) specify the Unicode range. The *out-start-hex* field (or the *out-hex* field) specifies the start of the output encoding range. The length of the *out-start-hex* (or *out-hex*) string determines the length of the output characters (e.g., UTF-8 uses different numbers of bytes to represent characters in different ranges). Entries must be given in increasing Unicode order. Only one file is allowed per encoding; the last specified file is used. The *Latin1*, *ASCII7*, *Symbol*, *ZapfDingbats*, *UTF-8*, and *UCS-2* encodings are predefined.

cMapDir *registry-ordering dir*

Specifies a search directory, *dir*, for CMaps for the *registry-ordering* character collection. There can be multiple directories for a particular collection. There are no default CMap directories.

toUnicodeDir *dir*

Specifies a search directory, *dir*, for ToUnicode CMaps. There can be multiple ToUnicode directories. There are no default ToUnicode directories.

GENERAL FONT CONFIGURATION**fontFile** *PDF-font-name font-file*

Maps a PDF font, *PDF-font-name*, to a font for display or PostScript output. The font file, *font-file*, can be any type allowed in a PDF file. This command can be used for 8-bit or 16-bit (CID) fonts.

fontDir *dir*

Specifies a search directory for font files. There can be multiple fontDir commands; all of the specified directories will be searched in order. The font files can be Type 1 (.pfa or .pfb) or TrueType (.ttf or .ttc); other files in the directory will be ignored. The font file name (not including the extension) must exactly match the PDF font name. This search is performed if the font name doesn't match any of the fonts declared with the fontFile command. There are no default fontDir directories.

fontFileCC *registry-ordering font-file*

Maps the *registry-ordering* character collection to a font for display or PostScript output. This mapping is used if the font name doesn't match any of the fonts declared with the fontFile, fontDir, psResidentFont16, or psResidentFontCC commands.

POSTSCRIPT FONT CONFIGURATION**psFontPassthrough** yes | no

If set to "yes", pass 8-bit font names through to the PostScript output without substitution. Fonts which are not embedded in the PDF file are expected to be available on the printer. This defaults to "no".

psResidentFont *PDF-font-name PS-font-name*

When the 8-bit font *PDF-font-name* is used (without embedding) in a PDF file, it will be translated to the PostScript font *PS-font-name*, which is assumed to be resident in the printer. Typically, *PDF-font-name* and *PS-font-name* are the same. By default, only the Base-14 fonts are assumed to be resident.

psResidentFont16 *PDF-font-name wMode PS-font-name encoding*

When the 16-bit (CID) font *PDF-font-name* with writing mode *wMode* is used (without embedding) in a PDF file, it will be translated to the PostScript font *PS-font-name*, which is assumed to be resident in the printer. The writing mode must be either 'H' for horizontal or 'V' for vertical. The resident font is assumed to use the specified encoding (which must have been defined with the `unicodeMap` command).

psResidentFontCC *registry-ordering wMode PS-font-name encoding*

When a 16-bit (CID) font using the *registry-ordering* character collection and *wMode* writing mode is used (without embedding) in a PDF file, the PostScript font, *PS-font-name*, is substituted for it. The substituted font is assumed to be resident in the printer. The writing mode must be either 'H' for horizontal or 'V' for vertical. The resident font is assumed to use the specified encoding (which must have been defined with the `unicodeMap` command).

psEmbedType1Fonts yes | no

If set to "no", prevents embedding of Type 1 fonts in generated PostScript. This defaults to "yes".

psEmbedTrueTypeFonts yes | no

If set to "no", prevents embedding of TrueType fonts in generated PostScript. This defaults to "yes".

psEmbedCIDTrueTypeFonts yes | no

If set to "no", prevents embedding of CID TrueType fonts in generated PostScript. For Level 3 PostScript, this generates a CID font, for lower levels it generates a non-CID composite font. This defaults to "yes".

psEmbedCIDPostScriptFonts yes | no

If set to "no", prevents embedding of CID PostScript fonts in generated PostScript. For Level 3 PostScript, this generates a CID font, for lower levels it generates a non-CID composite font. This defaults to "yes".

POSTSCRIPT CONTROL**psPaperSize** *width(pts) height(pts)*

Sets the paper size for PostScript output. The *width* and *height* parameters give the paper size in PostScript points (1 point = 1/72 inch).

psPaperSize letter | legal | A4 | A3 | match

Sets the paper size for PostScript output to a standard size. The default paper size is set when `xpdf` and `pdftops` are built, typically to "letter" or "A4". This can also be set to "match", which will set the paper size to match the size specified in the PDF file.

psImageableArea llx lly urx ury

Sets the imageable area for PostScript output. The four integers are the coordinates of the lower-left and upper-right corners of the imageable region, specified in points (with the origin being the lower-left corner of the paper). This defaults to the full paper size; the `psPaperSize` option will reset the imageable area coordinates.

psCrop yes | no

If set to "yes", PostScript output is cropped to the CropBox specified in the PDF file; otherwise no cropping is done. This defaults to "yes".

psUseCropBoxAsPage yes | no

If set to "yes", PostScript output treats the CropBox as the page size. By default, this is "no", and the MediaBox is used as the page size.

psExpandSmaller yes | no

If set to "yes", PDF pages smaller than the PostScript imageable area are expanded to fill the imageable area. Otherwise, no scaling is done on smaller pages. This defaults to "no".

psShrinkLarger yes | no

If set to yes, PDF pages larger than the PostScript imageable area are shrunk to fit the imageable area. Otherwise, no scaling is done on larger pages. This defaults to "yes".

psCenter yes | no

If set to yes, PDF pages smaller than the PostScript imageable area (after any scaling) are centered in the imageable area. Otherwise, they are aligned at the lower-left corner of the imageable area. This defaults to "yes".

psDuplex yes | no

If set to "yes", the generated PostScript will set the "Duplex" pagedevice entry. This tells duplex-capable printers to enable duplexing. This defaults to "no".

psLevel level1 | level1sep | level2 | level2sep | level3 | level3Sep

Sets the PostScript level to generate. This defaults to "level2".

psPreload yes | no

If set to "yes", PDF forms are converted to PS procedures, and image data is preloaded. This uses more memory in the PostScript interpreter, but generates significantly smaller PS files in situations where, e.g., the same image is drawn on every page of a long document. This defaults to "no".

psOPI yes | no

If set to "yes", generates PostScript OPI comments for all images and forms which have OPI information. This option is only available if the Xpdf tools were compiled with OPI support. This defaults to "no".

psASCIHex yes | no

If set to "yes", the ASCIIHexEncode filter will be used instead of ASCII85Encode for binary data. This defaults to "no".

psLZW yes | no

If set to "yes", the LZWEncode filter will be used for lossless compression in PostScript output; if set to "no", the RunLengthEncode filter will be used instead. LZW generates better compression (smaller PS files), but may not be supported by some printers. This defaults to "yes".

psUncompressPreloadedImages yes | no

If set to "yes", all preloaded images in PS files will uncompressed. If set to "no", the original compressed images will be used when possible. The "yes" setting is useful to work around certain buggy PostScript interpreters. This defaults to "no".

psMinLineWidth float

Set the minimum line width, in points, for PostScript output. The default value is 0 (no minimum).

psRasterResolution float

Set the resolution (in dpi) for rasterized pages in PostScript output. (Pdftops will rasterize pages which use transparency.) This defaults to 300.

psRasterMono yes | no

If set to "yes", rasterized pages in PS files will be monochrome (8-bit gray) instead of color. This defaults to "no".

psRasterSliceSize pixels

When rasterizing pages, pdftops splits the page into horizontal "slices", to limit memory usage. This option sets the maximum slice size, in pixels. This defaults to 20000000 (20 million).

psAlwaysRasterize yes | no

If set to "yes", all PostScript output will be rasterized. This defaults to "no".

psFile *file-or-command*

Sets the default PostScript file or print command for xpdf. Commands start with a `|' character; anything else is a file. If the file name or command contains spaces it must be quoted. This defaults to unset, which tells xpdf to generate a name of the form <file>.ps for a PDF file <file>.pdf.

fontDir *dir*

See the description above, in the DISPLAY FONTS section.

TEXT CONTROL**textEncoding** *encoding-name*

Sets the encoding to use for text output. (This can be overridden with the "-enc" switch on the command line.) The *encoding-name* must be defined with the unicodeMap command (see above). This defaults to "Latin1".

textEOL *unix | dos | mac*

Sets the end-of-line convention to use for text output. The options are:

```
unix = LF
dos  = CR+LF
mac  = CR
```

(This can be overridden with the "-eol" switch on the command line.) The default value is based on the OS where xpdf and pdftotext were built.

textPageBreaks *yes | no*

If set to "yes", text extraction will insert page breaks (form feed characters) between pages. This defaults to "yes".

textKeepTinyChars *yes | no*

If set to "yes", text extraction will keep all characters. If set to "no", text extraction will discard tiny (smaller than 3 point) characters after the first 50000 per page, avoiding extremely slow run times for PDF files that use special fonts to do shading or cross-hatching. This defaults to "yes".

MISCELLANEOUS SETTINGS**initialZoom** *percentage | page | width*

Sets the initial zoom factor. A number specifies a zoom percentage, where 100 means 72 dpi. You may also specify 'page', to fit the page to the window size, or 'width', to fit the page width to the window width.

continuousView *yes | no*

If set to "yes", xpdf will start in continuous view mode, i.e., with one vertical scrollbar for the whole document. This defaults to "no".

enableFreeType *yes | no*

Enables or disables use of FreeType (a TrueType / Type 1 font rasterizer). This is only relevant if the Xpdf tools were built with FreeType support. ("enableFreeType" replaces the old "freetype-Control" option.) This option defaults to "yes".

enableFreeType *yes | no*

Enables or disables use of FreeType (a TrueType / Type 1 font rasterizer). This is only relevant if the Xpdf tools were built with FreeType support. ("enableFreeType" replaces the old "freetype-Control" option.) This option defaults to "yes".

disableFreeTypeHinting *yes | no*

If this is set to "yes", FreeType hinting will be forced off. This option defaults to "no".

antialias *yes | no*

Enables or disables font anti-aliasing in the PDF rasterizer. This option affects all font rasterizers. ("antialias" replaces the anti-aliasing control provided by the old "t1libControl" and "freetypeControl" options.) This default to "yes".

vectorAntialias *yes | no*

Enables or disables anti-aliasing of vector graphics in the PDF rasterizer. This defaults to "yes".

antialiasPrinting *yes | no*

If this is "yes", bitmaps sent to the printer will be antialiased (according to the "antialias" and "vectorAntialias" settings). If this is "no", printed bitmaps will not be antialiased. This defaults to

"no".

strokeAdjust *yes | no*

Enables or disables stroke adjustment. Stroke adjustment moves horizontal and vertical lines by up to half a pixel to make them look "cleaner" when vector anti-aliasing is enabled. This defaults to "yes".

screenType *dispersed | clustered | stochasticClustered*

Sets the halftone screen type, which will be used when generating a monochrome (1-bit) bitmap. The three options are dispersed-dot dithering, clustered-dot dithering (with a round dot and 45-degree screen angle), and stochastic clustered-dot dithering. By default, "stochasticClustered" is used for resolutions of 300 dpi and higher, and "dispersed" is used for resolutions lower than 300 dpi.

screenSize *integer*

Sets the size of the (square) halftone screen threshold matrix. By default, this is 4 for dispersed-dot dithering, 10 for clustered-dot dithering, and 100 for stochastic clustered-dot dithering.

screenDotRadius *integer*

Sets the halftone screen dot radius. This is only used when screenType is set to stochasticClustered, and it defaults to 2. In clustered-dot mode, the dot radius is half of the screen size. Dispersed-dot dithering doesn't have a dot radius.

screenGamma *float*

Sets the halftone screen gamma correction parameter. Gamma values greater than 1 make the output brighter; gamma values less than 1 make it darker. The default value is 1.

screenBlackThreshold *float*

When halftoning, all values below this threshold are forced to solid black. This parameter is a floating point value between 0 (black) and 1 (white). The default value is 0.

screenWhiteThreshold *float*

When halftoning, all values above this threshold are forced to solid white. This parameter is a floating point value between 0 (black) and 1 (white). The default value is 1.

minLineWidth *float*

Set the minimum line width, in device pixels. This affects the rasterizer only, not the PostScript converter (except when it uses rasterization to handle transparency). The default value is 0 (no minimum).

drawAnnotations *yes | no*

If set to "no", annotations will not be drawn or printed. The default value is "yes".

overprintPreview *yes | no*

If set to "yes", generate overprint preview output, honoring the OP/op/OPM settings in the PDF file. Ignored for non-CMYK output. The default value is "no".

launchCommand *command*

Sets the command executed when you click on a "launch"-type link. The intent is for the command to be a program/script which determines the file type and runs the appropriate viewer. The command line will consist of the file to be launched, followed by any parameters specified with the link. Do not use "%s" in "command". By default, this is unset, and Xpdf will simply try to execute the file (after prompting the user).

urlCommand *command*

Sets the command executed when you click on a URL link. The string "%s" will be replaced with the URL. (See the example below.) This has no default value.

movieCommand *command*

Sets the command executed when you click on a movie annotation. The string "%s" will be replaced with the movie file name. This has no default value.

mapNumericCharNames *yes / no*

If set to "yes", the Xpdf tools will attempt to map various numeric character names sometimes used in font subsets. In some cases this leads to usable text, and in other cases it leads to gibberish -- there is no way for Xpdf to tell. This defaults to "yes".

mapUnknownCharNames *yes / no*

If set to "yes", and mapNumericCharNames is set to "no", the Xpdf tools will apply a simple pass-through mapping (Unicode index = character code) for all unrecognized glyph names. (For CID fonts, setting mapNumericCharNames to "no" is unnecessary.) In some cases, this leads to usable text, and in other cases it leads to gibberish -- there is no way for Xpdf to tell. This defaults to "no".

mapExtTrueTypeFontsViaUnicode *yes / no*

When rasterizing text using an external TrueType font, there are two options for handling character codes. If mapExtTrueTypeFontsViaUnicode is set to "yes", Xpdf will use the font encoding/ToUnicode info to map character codes to Unicode, and then use the font's Unicode cmap to map Unicode to GIDs. If mapExtTrueTypeFontsViaUnicode is set to "no", Xpdf will assume the character codes are GIDs (i.e., use an identity mapping). This defaults to "yes".

enableXFA *yes / no*

If set to "yes", an XFA form (if present) will be rendered in place of an AcroForm. If "no", an XFA form will never be rendered. This defaults to "yes".

bind *modifiers-key context command ...*

Add a key or mouse button binding. *Modifiers* can be zero or more of:

shift-
ctrl-
alt-

Key can be a regular ASCII character, or any one of:

space
tab
return
enter
backspace
insert
delete
home
end
pgup
pgdn
left / right / up / down (arrow keys)
f1 .. f35 (function keys)
mousePress1 .. mousePress7 (mouse buttons)
mouseRelease1 .. mouseRelease7 (mouse buttons)

Context is either "any" or a comma-separated combination of:

fullScreen / window (full screen mode on/off)
continuous / singlePage (continuous mode on/off)
overLink / offLink (mouse over link or not)
scrLockOn / scrLockOff (scroll lock on/off)

The context string can include only one of each pair in the above list.

Command is an Xpdf command (see the **COMMANDS** section of the **xpdf(1)** man page for details). Multiple commands are separated by whitespace.

The **bind** command replaces any existing binding, but only if it was defined for the exact same modifiers, key, and context. All tokens (modifiers, key, context, commands) are case-sensitive.

Example key bindings:

```
# bind ctrl-a in any context to the nextPage
# command
bind ctrl-a any nextPage

# bind uppercase B, when in continuous mode
# with scroll lock on, to the reload command
# followed by the prevPage command
bind B continuous,scrLockOn reload prevPage
```

See the **xpdf(1)** man page for more examples.

unbind *modifiers-key context*

Removes a key binding established with the **bind** command. This is most useful to remove default key bindings before establishing new ones (e.g., if the default key binding is given for "any" context, and you want to create new key bindings for multiple contexts).

printCommands *yes / no*

If set to "yes", drawing commands are printed as they're executed (useful for debugging). This defaults to "no".

errQuiet *yes / no*

If set to "yes", this suppresses all error and warning messages from all of the Xpdf tools. This defaults to "no".

EXAMPLES

The following is a sample xpdfrc file.

```
# from the Thai support package
nameToUnicode /usr/local/share/xpdf/Thai.nameToUnicode

# from the Japanese support package
cidToUnicode Adobe-Japan1 /usr/local/share/xpdf/Adobe-Japan1.cidToUnicode
unicodeMap JISX0208 /usr/local/share/xpdf/JISX0208.unicodeMap
cMapDir Adobe-Japan1 /usr/local/share/xpdf/cmap/Adobe-Japan1

# use the Base-14 Type 1 fonts from ghostscript
fontFile Times-Roman /usr/local/share/ghostscript/fonts/n021003l.pfb
fontFile Times-Italic /usr/local/share/ghostscript/fonts/n021023l.pfb
fontFile Times-Bold /usr/local/share/ghostscript/fonts/n021004l.pfb
fontFile Times-BoldItalic /usr/local/share/ghostscript/fonts/n021024l.pfb
fontFile Helvetica /usr/local/share/ghostscript/fonts/n019003l.pfb
fontFile Helvetica-Oblique /usr/local/share/ghostscript/fonts/n019023l.pfb
fontFile Helvetica-Bold /usr/local/share/ghostscript/fonts/n019004l.pfb
fontFile Helvetica-BoldOblique /usr/local/share/ghostscript/fonts/n019024l.pfb
fontFile Courier /usr/local/share/ghostscript/fonts/n022003l.pfb
fontFile Courier-Oblique /usr/local/share/ghostscript/fonts/n022023l.pfb
fontFile Courier-Bold /usr/local/share/ghostscript/fonts/n022004l.pfb
fontFile Courier-BoldOblique /usr/local/share/ghostscript/fonts/n022024l.pfb
fontFile Symbol /usr/local/share/ghostscript/fonts/s050000l.pfb
```



```
fontFile ZapfDingbats      /usr/local/share/ghostscript/fonts/d050000l.pfb

# use the Bakoma Type 1 fonts
# (this assumes they happen to be installed in /usr/local/fonts/bakoma)
fontDir /usr/local/fonts/bakoma

# set some PostScript options
psPaperSize      letter
psDuplex         no
psLevel          level2
psEmbedType1Fonts yes
psEmbedTrueTypeFonts yes
psFile           "| lpr -Pprinter5"

# assume that the PostScript printer has the Univers and
# Univers-Bold fonts
psResidentFont Univers      Univers
psResidentFont Univers-Bold Univers-Bold

# set the text output options
textEncoding UTF-8
textEOL         unix

# misc options
enableFreeType yes
launchCommand   viewer-script
urlCommand      "netscape -remote 'openURL(%s)'"
```

FILES

/usr/local/etc/xpdfrc

This is the default location for the system-wide configuration file. Depending on build options, it may be placed elsewhere.

\$HOME/.xpdfrc

This is the user's configuration file. If it exists, it will be read in place of the system-wide file.

AUTHOR

The Xpdf software and documentation are copyright 1996-2014 Glyph & Cog, LLC.

SEE ALSO

xpdf(1), **pdftops(1)**, **pdftotext(1)**, **pdftohtml(1)**, **pdfinfo(1)**, **pdffonts(1)**, **pdfdetach(1)**, **pdftoppm(1)**, **pdftopng(1)**, **pdfimages(1)**
<http://www.foolabs.com/xpdf/>