

pdfaPilot CLI

Manual

pdfaPilot CLI – Manual – Last modified: 19 March 2012

© 1996-2012 by callas software gmbh, Berlin, Germany
All rights reserved

All trademarks are the property of their respective owners.

Content

Getting started	5
System requirements	5
Installing the software	5
Macintosh/Windows	5
Linux/Solaris/AIX	5
Activation	5
Request an activation code	5
Activating pdfaPilot CLI	6
Time-limited trial version	6
Displaying program information	6
Display program version	6
Display usage information	6
Display status	7
Performance enhancement	7
Get in touch	8
Processing	9
Input files from Office applications	9
OpenOffice	9
Create PDF for print	9
General options	9
Input file	9
Process folders recursively	9
Empty the font cache	9
Analyze only	9
Analyze only certain pagerange	10
PDF/A specific options	10
PDF/A Compliancy level	10
Deactivate removal of non-compliant metadata	10
Force Conversion to PDF/A	10
File content options	11
Deactivate transparency flattening	11
Add XMP metadata	11
Add bookmarks	11
Set the OutputIntent	11
Define ICC profiles	11
Font Embedding	12
Define an additional font folder	12
Substitute fonts	12
Creating file packages	13

Settings for file embedding	13
Report creation	14
Report types and their options	14
Report triggers	15
Report path	15
Set the report language	15
Command line output	16
Display progress	16
Display hits	16
Display summary	16
Display timestamp	16
Further options	16
Quick processing	16
Define the overwrite mode	16
Create output files only for successful conversion	16
Set the result path	17
Use an additional profile	17
Using response files	17
Structure of command line output	18
Checks	18
Corrections	18
Failed corrections	18
Missing glyphs	19
Progress	19
Results	20
Return codes and their usage	20
When executing kfp profiles	20
When executing in pdfa mode	20
When executing the compare action	20
All other commands or actions	20
Reason codes	21

Getting started

pdfaPilot CLI may be used to analyse files or to analyse and convert to PDF/A (if possible).

System requirements

The command line version of pdfaPilot is available for the following operating systems:

- Windows 2000/2003 Server/XP/Vista/7
 - Mac OS X 10.5 or newer, Intel or PPC
 - Linux Debian 5.0 (Lenny)
 - Sun Solaris SPARC version 8 or newer
 - Sun Solaris Intel version 10 or newer
 - AIX 5.3 (oslevel 5300-07) or newer (call `oslevel -q` to check)
- ☛ Spoken generally, pdfaPilot CLI should work with other Linux distributions as well, as long as there are system libraries installed that are compatible to gcc-v3.4 or newer. The dependent libstdc++ is delivered with the pdfaPilot CLI.

You can easily test if pdfaPilot CLI is working on your system: Just type `pdfaPilot --help` in the terminal.

- ☛ pdfaPilot CLI does also run on 64 bit systems if the required 32 bit compatibility packages are available.

Installing the software

Macintosh/Windows

To install the software start the pdfaPilot Server installer. The installation program will then take you through the necessary steps.

Linux/Solaris/AIX

Extract all files from the archive to a destination folder of your choice.

For automation purposes you should set the PATH variable to the path of the pdfaPilot CLI executable.

Additional information is provided in `<pdfaPilot CLI directory>/ReadMe.txt`

Activation

Before callas pdfaPilot CLI can be used, the software has to be activated.

Request an activation code

Open a terminal window and change to your pdfaPilot CLI installation directory. Type:

```
pdfaPilot --keycode <name> <company> <licenceCode>
```

Parameters

name	Name of licensee (e.g. "Registered User")
company	Name of company (e.g. "User's company")
licencecode	Licence key obtained from the registration card To make a request for a trial version, please use the keyword "trial" (for a pdfaPilot trial version) for this parameter The textual output of --keycode has to be send via email to the email address named in the text in order to receive an activation code from the registration server.

Activating pdfaPilot CLI

After having received the automatical reply email to the activation request, save the attached licence file to the file system. Then use the following command:

```
pdfaPilot --activate <licence file>
```

Parameters

licence file Full path to licence file

- 🔑 In order to activate pdfaPilot CLI for all user accounts of one machine, save the license file next to the pdfToolbox binary instead of installing it by using the `--activate` command.
- 🔑 pdfaPilot CLI is searching for the license file at various folders:
 - user-preferences-folder of actual user
 - next to the pdfaPilot CLI binary
 - cachefolder (if set)
 - user-preferences-folder for all users (shared)
- 🔑 When using UNIX-based-systems the environment variable `CALLAS_SYSTEM_PREFERENCES` the path of the standard `/usr/share/callas software/callas pdfaPilot CLI` can be changed:


```
CALLAS_SYSTEM_PREFERENCES=tmp
```

 would result in the searchpath: `/tmp/callas software/callas pdfaPilot CLI`
 It is highly recommended to use the option `--cachefolder` instead.

Time-limited trial version

After requesting and entering a trial activation code, pdfaPilot CLI can be tested without any restrictions. When the evaluation period has expired, processing PDF files will no longer be possible until you request and enter a new activation code.

Displaying program information

Display program version

```
pdfaPilot --version
```

will display the currently used version of pdfaPilot CLI.

Display usage information

```
pdfaPilot --help
```

will give you a complete overview about all available commands for processing.

```
pdfaPilot --help <command>
```

will give you an overview about all available options for the command.

Display status

```
pdfaPilot --status
```

will inform you about the current license state as well as the possible return and reason codes (see "Results").

Performance enhancement

If you want to enhance the performance of your pdfaPilot CLI processes, please keep in mind the following rules:

- For embedding missing fonts your system font folder will be scanned unless defined otherwise (see "Font Embedding"). pdfaPilot will create a font cache to improve the performance time, but still it might be useful to remove fonts that are not needed from this directory.

Get in touch

If some necessary information is not provided by this manual or if there are any questions or feedback please contact the product management by using the "Contact Support" form on www.callassoftware.com.

You can also send an e-mail to support@callassoftware.com.

If you file a bug report please make sure your inquiry contains the following information:

- operating system
- pdfaPilot version (call `pdfaPilot --version`)
- command line call
- original PDF (please delete unnecessary pages to avoid long file transfers), used profiles or configuration files
- converted PDF (if available)

You can also visit the support section on www.callassoftware.com to get answers to common questions or find a reseller near you. The latter might be useful if you want to send a support request that is neither in English nor German.

Processing

Input files from Office applications

pdfaPilot CLI is able to convert common file formats from Office applications directly to PDF/A. For more information and a list of supported applications and files have a look at:

<http://www.callassoftware.com/callas/doku.php/en:support:faqs:topdf>

🔊 Office file conversion is currently not supported on Solaris and AIX systems.

OpenOffice

```
--topdf_forceopenOffice
```

When defined, Microsoft Office files are processed with OpenOffice.

Create PDF for print

```
--topdf_print
```

The PDF will be created with image resolution sufficient for printing, thus leading to larger files.

General options

Usually pdfaPilot CLI is started with:

```
pdfaPilot <PDF file>
```

Input file

May be one or a number of input files (PDF or Office files) to be analysed and converted.

If an input file spec is pointing to an existing folder, all files inside this folder are processed:

Process folders recursively

```
--recursive
```

If the file spec for the input file is pointing to an existing folder all PDF files inside the folder on all levels are processed

Empty the font cache

```
--emptyfontcache
```

Removes all font files from the font cache folder of pdfaPilot CLI.

Analyze only

```
-a --analyze
```

The input file is not being converted but is analyzed whether it is PDF/A compliant.

Analyze only certain pagerange

```
--p --pagerange=<firstpage>[-<lastpage>]
```

Only applied when analyzing not when converting PDF files.

- When converting non-PDF documents the page range of the original document can be specified.

Parameters

firstpage	Page where analysis should start
lastpage	optional, page where analysis should end

Example:

```
--pagerange=5-33
```

Setting the cache folder

```
--cachefolder=<path>
```

Sets the cache folder path. This is set by default to:

Windows: C:\Documents and Settings\<user>\Application data\
callas software\callas pdfaPilot CLI

Macintosh: /Users/<user>/Library/Preferences/callas software/
callas pdfaPilot CLI

Unix: <home directory as defined in /etc/passwd>/.callas software/
callas pdfaPilot CLI

- This option is mandatory when running the CLI as a user without a home directory.

Parameters

path	absolute path to custom cache folder
------	--------------------------------------

PDF/A specific options

PDF/A Compliancy level

```
--level=<level>
```

You can define which PDF/A level you need (default is 1b).

Parameters

level	3b, 3u, 3a, 2b, 2u, 2a, 1b or 1a
-------	----------------------------------

Deactivate removal of non-compliant metadata

```
--noxmpremoval
```

Normally, XMP Metadata which is not compliant with PDF/A is removed during conversion. This switch prevents the removal.

Force Conversion to PDF/A

Due to several reasons a regular conversion may not result in a valid PDF/A document. To ensure conversion 3 additional steps can be performed after a normal conversion if a file can not be converted to PDF/A within this first step. The order they will be performed is as listed below.

After each step the resulting file is checked for compatibility with the chosen standard. Each step is optional.

```
--forceconversion_reconvert
```

Performs a re-conversion of the PDF via PostScript.

```
--forceconversion_pagestoimages
```

Convert pages with problems into images, while converting the text into invisible text, but correctly positioned to keep the text available for marking and copying.

```
--forceconversion_doctoimages
```

Convert all pages into images, while converting the text into invisible text, correctly positioned to keep the text available for marking and copying.--

```
forceconversion_resolution
```

Image resolution in ppi used for the rendered content. (Default = 100 ppi).

File content options

Deactivate transparency flattening

```
--notransparencyflattening
```

This switches off both flattening of any contained transparency and setting the blend color space to sRGB.

Add XMP metadata

```
--addxmp=<path>
```

The XMP metadata is merged into any existing XMP metadata.

Parameters

path	Path to an XMP file
------	---------------------

Add bookmarks

```
--addbookmarks=<path>
```

The bookmarks are embedded into the PDF file.

Parameters

path	Path to an XML file containing bookmarks
------	--

Set the OutputIntent

```
-i --OutputIntent=<path>
```

Path to a PDF file with an OutputIntent – forces use of this OutputIntent.

Parameters

path	Path to an OutputIntent
------	-------------------------

Define ICC profiles

CMYK

```
--defaultprofile_cmyk=<path>
```

The given profile is embedded as the default profile making device dependent CMYK page objects device independent.

Parameters

path Path to a CMYK ICC profile

RGB

```
--defaultprofile_rgb=<path>
```

The given profile is embedded as the default profile making device dependent RGB page objects device independent.

Parameters

path Path to an RGB ICC profile

Gray

```
--defaultprofile_gray=<path>
```

The given profile is embedded as the default profile making device dependent Gray page objects device independent.

Parameters

path Path to a Gray ICC profile

Font Embedding

If a font is not embedded pdfaPilot CLI will search the system's font directories in order to find the needed font file, which are:

Windows	• C:\Windows\Fonts
Macintosh	• /Users/<user>/Library/Fonts
	• /Library/Fonts
	• /System/Library/Fonts
Linux, Solaris Sparc, Solaris x86, AIX	• /usr/lib/X11/fonts
	• /<user home>/fonts

Additionally the font folder installed together with pdfaPilot CLI will be searched. This folder lies next to the executable in "*<callas pdfaPilot CLI directory>|etc|APDFL|Resource|Font*".

Define an additional font folder

```
--fontfolder=<path>
```

Additional folder to look up fonts for embedding.

Parameters

path Path to config file

Example:

```
--fontfolder="C:\AdditionalFonts"
```

- You can force pdfaPilot CLI to only scan the folder defined by `--fontfolder` (and not search the system's font folders) by using the option `--fontonly`.

Substitute fonts

```
--substitute[=<path>]
```

Font substitution can be used when the original fonts used in a PDF file are not available. By default, the font substitution file `pdfa.cfg` stored in `"etc/FontSubstitution/"` provides the basis for substitution.

Alternatively, you can also enter a custom path to a fontsubstitution file, e.g.:

```
--substitute=C:\fontsubstitution.cfg
```

If you want to switch off font substitution completely, just hand over an invalid value, e.g.:

```
--substitute=no
```

Parameters

path	Optional, path to config file
------	-------------------------------

The following notations are allowed in the `fontsubstitution.cfg`:

SubstituteAll

Each font can be substituted by every other font of the entry.

```
SubstituteAll<tab>fontname<tab>fontname<tab>fontname...
```

SubstituteFirst

Only the first font of the entry can be substituted by the following fonts.

```
SubstituteFirst<tab>font to be substituted<tag>fontname<tab>...
```

Creating file packages

Some PDF standards allows the embedding of PDF- and also non-PDF-files into another PDF file. Sometime these filepackages are also called collections. Using pdfaPilot CLI it is possible to define different ways how a file which shall be embedded is handled. In general a file package is created with `--collection` This will create an index document, which lists all embedded files.

```
--collection <file> [<file>]
```

Settings for file embedding

```
--collection [--embedinto=[target],<file>] [--embedfile=[target,[relationship],<file>]
```

--embedinto

It is possible to use own templates or normal PDF for embedding files. The standard for the file where other files will be embedded can be defined

using the conversion target (see below). If no file is defined, an index file is created.

--embedfile

Also for files to embed a conversion target can be defined using the conversion target. For PDF/A-3 standards also a relationship entry for each embedded file can be set.

Parameters

target A3b, A3u, A3a, A2b, A2u, A2a, A1b, A1a or No (Default)

Using the target "No", no conversion to PDF is done. (Only available for embedded files.)

relationship Source, Data, Alternative, Supplement, Unspecified (Default)

Example

```
--collection --embedinto=A3b,<PDF file> --embedfile=A3b,
Alternative,<file> --embedfile=A2b,Source,<Office file>
--embedfile=No,Data,<file>
```

Report creation

For each pdfaPilot CLI run several reports may be generated by inserting `--report` or `-r` switches into the pdfaPilot call. Following all options and reporttypes are listed. The options of `--report` are treated case insensitive and have to be separated by commas.

```
--report=[<type>,<trigger>,<options>,<PATH=path>
```

Parameters

type Optional, see "Report types and their options"

trigger Optional, see "Report triggers"

options Optional, see "Report types and their options"

path See "Report path"

Report types and their options

HTML (default)

A html file is created. The format may be modified by combining this option with other options. It can be opened by any webbrowser.

Additional options for HTML reports

All following switches may be simultaneously used in order to add different types of additional content.

NOICONS	Create a report without any images
NOCORRECTIONS	Do not log corrections
NODETAILS	Suppress details for the occurrences
OPENRESULT	All entries in the report are opened in Initial view (closed by default if JavaScript is enabled)

Set the path for referenced objects

```
--linkpath
```

HTML reports are generated without exporting referenced objects. This option requires a path (URL) to a folder where the referenced objects reside (Default: folder "`etc\reporttemplate`" in pdfaPilot CLI directory)

Example:

```
--linkpath="file:\\Programme\pdfaPilot\etc\reporttemplate"
```

Customize your report

If you create a HTML report, you can completely customize it by adapting the CSS-File to your needs. You can even exchange the pictures.

You can find all material used for the HTML reports within the folder "`etc\reporttemplate`" in your pdfaPilot CLI directory.

XML

XML file which is intended to be processed with software (parsers). The scheme of the XML report structure can be found within the xsd-file stored in "`var|XMLV2 report schema`".

MHT

A mht file is created. MHT is a data format created by Microsoft which offers the possibility to include all resources (like style sheets, images and JavaScripts) to be included in one file. This file type can only be read by Internet Explorer.

Report triggers

ALWAYS	Always create a report (default)
IFNOPDFA	Create a report only if the file cannot be converted to PDF/A
IFPDFA	Create a report only if the file can be converted to PDF/A

Report path

```
--report=<Report type>,<Report options>,PATH=<Path>
```

Full path of report file – the report path must not contain any commas.

🔊 PATH= must always be the last element.

Example:

```
--report=HTML,NOICONS,OPENRESULT,PATH=C:\Sample.html
```

Set the report language

```
-l --language=<language>
```

Defines the language in which the report is generated. Default language is English (en). The language is specified by using a two digit abbreviation. The following values can be used in the standard configuration:

Parameters

EN	English
DE	German
FR	French
IT	Italian
ES	Spanish
JA	Japanese

Command line output

Display progress

```
--noprogess
```

Switch off progress information.

Display hits

```
--nohits
```

Switch off output of hits (errors, warnings, information).

Display summary

```
--nosummary
```

Switch off summary information.

Display timestamp

```
--timestamp
```

Show a timestamp for each line of command line output.

Further options

Quick processing

```
-q --quick
```

Processing is stopped after the first detection of an error in the corrected PDF (only applied if no report is generated).

Define the overwrite mode

```
--overwrite
```

New files override existing files with the same name (applies to report files and to created PDF files).

Create output files only for successful conversion

```
--onlypdfa
```

Create an output file only if the file could be converted to PDF/A.

Set the result path

- 🔗 If neither an output path nor an output folder is defined, any result will be created next to the input file (default: input file name with suffix `_PDFA` or `_NOPDFA`, will be indexed if necessary).
- 🔗 The use of `--outputfile` together with `--outputfolder` is not supported within one CLI call.

Path to output file

```
-o --outputfile=<path>
```

Defines the absolute path of the destination file. The parent folder must exist.

- 🔗 Consult section "Results" to see if a new file was created.

Parameters

path	absolute path to output file
------	------------------------------

Path to output folder

```
-f --outputfolder=<path>
```

Defines an absolute path to a folder where the files resulting of an execution are stored.

Parameters

path	absolute path to output folder Output file
------	---

Use an additional profile

```
--profile
```

Run additional checks and fixups by defining a full path to a kfp or kfpX file (exported from pdfToolbox / pdfaPilot Desktop or Preflight in Adobe Acrobat). The input file is converted if no warnings or errors occur.

Using response files

To keep the command line call structured and straightforward, pdfaPilot CLI supports the usage of response files. These offer the possibility to define each command line switch line by line and also add some comments.

Example

Response file analyze.rsp:

```
#####
# PDF/A analysis
#
--analyze
#
#####
# EOF
```

Command line call:

```
pdfaPilot @analyze.rsp <PDF file>
```

Structure of command line output

Checks

Hit	Type of hit	Name of check	[Name of font]	[Name of glyph]
-----	-------------	---------------	----------------	-----------------

1. Hit

- always visible
- keyword

2. Type of hit

- always visible
- possible values: "PDF/A", "Error", "Info", "Warning" (only for additional profile)

3. Name of check

- always visible
- never empty

4. Name of font

- only visible for hits that belong to a font check
- can be empty

5. Name of glyph

- only visible for hits that belong to a font check
- can be empty

Corrections

Fix	Name of fixup	[Name of font]
-----	---------------	----------------

1. Fix

- always visible

- keyword
- 2. Name of fixup**
- always visible
- never empty
- 3. Name of font**
- only visible for fixups that belong to a font correction
- can be empty

Failed corrections

FixFailure	Name of fixup	[Name of font]
------------	---------------	----------------

1. FixFailure

- always visible
- keyword

2. Name of fixup

- always visible
- never empty

3. Name of font

- only visible for fixups that belong to a font correction
- can be empty

Missing glyphs

Hit	Missing glyphs in font <Name of font>	CID=<ID>
-----	---------------------------------------	----------

1. Hit

- always visible
- keyword

2. Missing glyphs in font <Name of font>

- always visible
- never empty

3. CID

- always visible
- Character ID of missing glyph

Progress

Progress	Value	%
----------	-------	---

Results

Return codes and their usage

All return codes below 100 indicate a successful operation.

When executing kfpX profiles

0	No hit, no fixups executed
1	At least one hit with severity 'info', no fixups executed
2	At least one hit with severity 'warning', no fixups executed
3	At least one hit with severity 'error', no fixups executed
5	No hit, fixups have been executed
6	At least one hit with severity 'info', fixups have been executed
7	At least one hit with severity 'warning', fixups have been executed
8	At least one hit with severity 'error', fixups have been executed; fixup failed

When executing in pdfa mode

0	PDF is valid PDF/A-file additional checks without problems
1	PDF is valid PDF/A-file but additional checks with problems – severity info
2	PDF is valid PDF/A-file but additional checks with problems – severity warning
3	PDF is valid PDF/A-file but additional checks with problems – severity error
4	PDF is not a valid PDF/A-file

When executing the compare action

0	Compared PDFs are equal
1	Compared PDFs have differences

All other commands or actions

0	Successful operation
---	----------------------

Errors:

100	Not serialized (no valid serialization found or keycode expired)
101	Command line parameter error
102	Command line syntax error (illegal command)
103	Unknown error (internal error)
104	A file could not be opened
105	An encrypted PDF file could not be opened for writing
106	A file could not be saved

- On a Windows computer the return code may be accessed by using the system variable `errorlevel`, on a Linux/Sun Solaris/Mac OS X computer `$?` can be used.

Reason codes

1000	Unknown reason
1001	A parameter is wrong
1002	A requested file could not be found
1003	A requested folder could not be found
1004	A requested folder is a file
1005	A requested file is a folder
1006	30 days trial period expired
1007	Time limited keycode expired
1008	Invalid activation
1009	PDF does not contain ICC profiles
1010	A file could not be opened
1011	An encrypted PDF file could not be opened for writing
1012	A file could not be saved