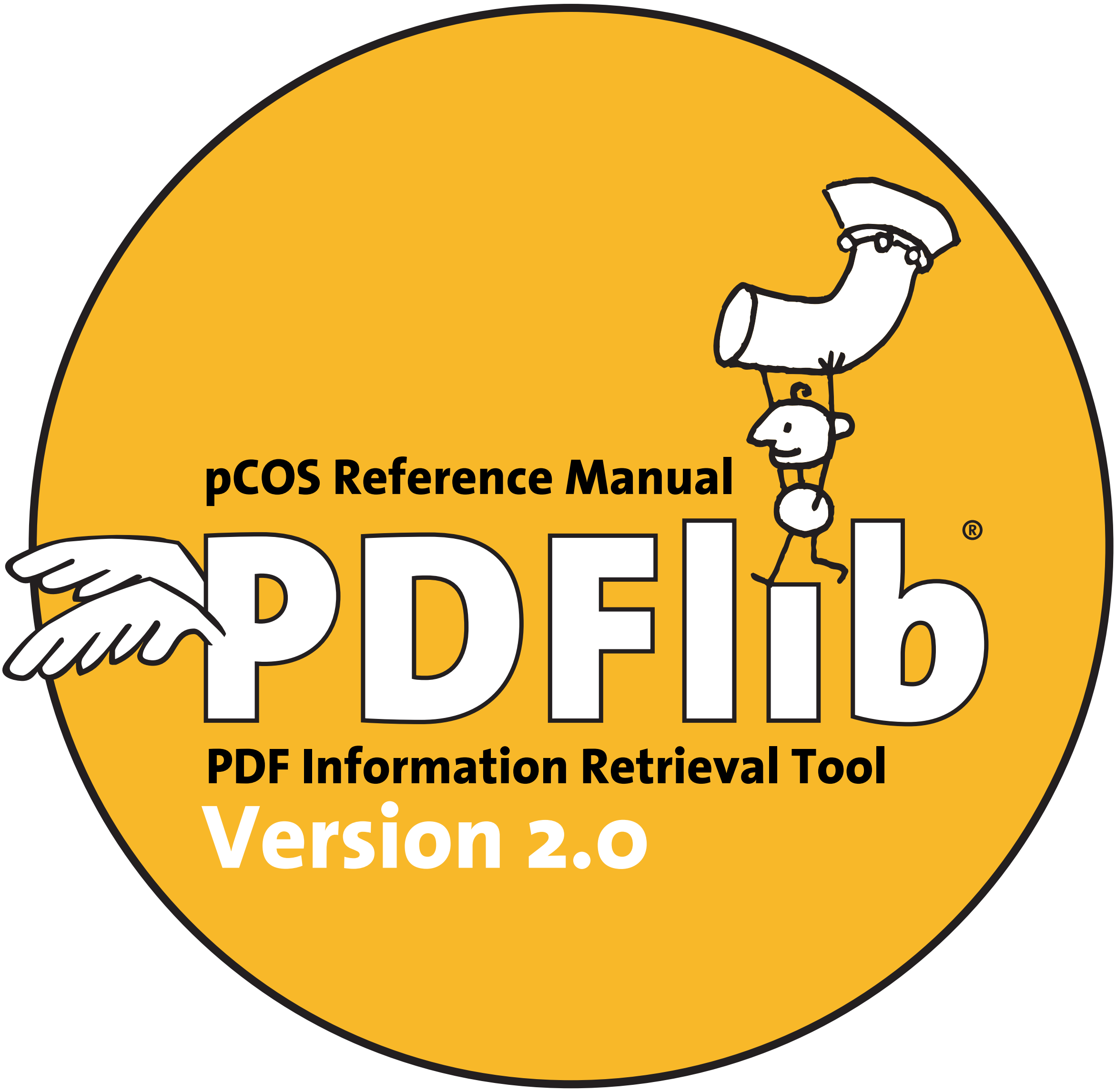
PDFlib GmbH München, Germany

www.pdflib.com



pCOS Reference Manual

PDFlib®

PDF Information Retrieval Tool

Version 2.0

Test Document only!

PDFlib GmbH
Franziska-Bilek-Weg 9, 80339 München, Germany
www.pdflib.com

phone +49 • 89 • 452 33 84-0
fax +49 • 89 • 452 33 84-99

If you have questions check the PDFlib mailing list and archive at tech.groups.yahoo.com/group/pdflib

Licensing contact: sales@pdflib.com
Technical support: support@pdflib.com (please include your license number)

This publication and the information herein is furnished as is, is subject to change without notice, and should not be construed as a commitment by PDFlib GmbH. PDFlib GmbH assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes and noninfringement of third party rights.

Contents

0 First Steps with pCOS

0.1 Installing the Software

pCOS is delivered as an MSI installer package for Windows systems, and as a compressed archive for all other supported operating systems. All pCOS packages contain the pCOS command-line tool and the pCOS library/component, plus support files, documentation, and examples. After installing or unpacking pCOS the following steps are recommended:

- An introduction to the pCOS features by means of various examples can be found in Chapter 1, »pCOS Examples«, page 9.
- Users of the pCOS command-line tool can use the executable right away. It can be found in the *bin* subdirectory of the installation directory. The available options are discussed in Chapter 2, »pCOS Command-Line Reference«, page 19, and are also displayed when you execute the pCOS command-line tool without any options.
- Users of the pCOS library/component should read one of the sections in Chapter 4, »pCOS Library Language Bindings«, page 43, corresponding to their environment of choice, and review the installed examples. On Windows, the pCOS programming examples are accessible via the Start menu.

If you obtained a commercial pCOS license you must enter your pCOS license key according to the next page.

Restrictions of the evaluation version. The pCOS command-line tool and library can be used as fully functional evaluation versions even without a commercial license. Unless a valid license key is applied, pCOS will support all features, but will only process PDF documents with up to 10 pages and 1 MB size. Unlicensed versions of pCOS must not be used for production purposes, but only for evaluating the product. Using pCOS for production purposes requires a valid license.

0.2 Applying the pCOS License Key

Using pCOS for production purposes requires a valid license key. Once you purchased a pCOS license you must apply your license key in order to allow processing of arbitrarily large documents. There are several methods for applying the license key; choose one of the methods detailed below.

Note pCOS license keys are platform-dependent, and can only be used on the platform for which they have been purchased.

Entering the license key in the Windows installer. Windows users can enter the license key when they install pCOS using the supplied installer. This is the recommended method on Windows. If you do not have write access to the registry or cannot use the installer, refer to one of the alternate methods below instead.

Entering the license key in a license file. Set an environment (shell) variable which points to a license file before pCOS functions are called. If you are using the pCOS library you can alternatively set the path to the license file by setting the *licensefile* parameter with the *pCOS_set_option( )* function. The license file must be a text file with the following structure (you can use the license file template *licensekeys.txt* which is contained in all distributions). Lines beginning with a '#' characters contain comments, and will be ignored:

```
# Licensing information for PDFlib GmbH products
PDFlib license file 1.0
pCOS 2.0 ...your license key...
```

The details of setting environment variables vary across systems, but a typical statement for a Unix shell looks as follows:

```
export PDFLIBLICENSEFILE="/path/to/your/license/file"
```

Set the license key in an option for the pCOS command-line tool. If you use the pCOS command-line tool you can supply an option which contains the name of a license file or the license key itself:

```
pcos --pcosopt "license ...your license key..." ...more options...

pcos --pcosopt "licensefile /path/to/your/license/file" ...more options...
```

If the path name contains space characters you must enclose the path with braces:

```
pcos --pcosopt "licensefile {/path/to/license file}" ...more options...
```

Setting the license key with a pCOS library call. If you use the pCOS library, add a line to your script or program which sets the license key at runtime. The *license* option must be set immediately after instantiating the pCOS object, i.e., after calling *pCOS_new( )* (in C) or creating a pCOS object (in C++, COM, .NET, and Java):

- In COM/VBScript:

```
p.set_option "license=...your license key..."
```

- In C:

```
pCOS_set_option(p, "license=...your license key...");
```

- In C++, .NET/C#, and Java:

```
p.set_option("license=...your license key...");
```

- In Perl:

```
pCOS_set_option($p, "license=...your license key...");
```

0.3 What's new in pCOS 2.0?

pCOS 2.0 extends the set of pseudo objects which were available in pCOS 1.0. The following PDF objects and properties can be conveniently queried without getting bogged down in the details of the PDF syntax:

- images, including bit depth, color properties, and compression filter
- color spaces, including simplified access to details of complex color spaces, e.g. Separation
- other page resources, such as graphics states, patterns, etc.
- page labels (e.g. roman numerals instead of decimal numbers)
- PDF/X and PDF/A status
- article threads (beads) including the number of the destination page

The PDF parser in pCOS 2.0 includes the following improvements:

- the new repair mode corrects various problems in PDF input documents.
- supports the AES encryption algorithm so that AES-encrypted PDF documents are accepted as input
- implements the CCITTFax, RunLengthDecode, and DCTDecode compression filters for image streams
- Predictors are supported for Flate and LZW compression (relevant for compressed image data)

Other changes:

- The pCOS cookbook on *www.pdflib.com/pcos-cookbook* presents dozens of pCOS programming examples.
- performance improvements for several special cases
- *pCOS_get_stream( )* now also supports string objects in addition to stream objects. This is useful for rare cases where PDF string objects are used to carry binary information, e.g. the color palette in an Indexed color space.
- *pCOS_get_stream( )* now also supports the option *convert* which treats stream contents as textual data, and applies appropriate Unicode conversion. This is relevant, for example, for JavaScript in stream objects.