

**Getting
started with
callas
pdfToolbox**

Table of Contents

Introduction	3
What exactly is pdfToolbox?	4
Common use cases with pdfToolbox.....	6
Working with PDF documents manually	12
What is pdfToolbox Desktop?	13
Working with PDF documents in pdfToolbox Desktop.....	15
The Switchboard: easy-to-use Actions	19
Introduction in Checks, Fixups, Profiles, Process Plans and Libraries	21
Profiles: the cornerstone of pdfToolbox.....	22
Process Plans: when Profiles are not enough.....	24
Libraries: for a broader overview.....	27
Analysing problem files.....	28
Processing PDF documents automatically	31
Working with hot folders in pdfToolbox Server	32
Advantages and limitations of using hot folders	34
pdfToolbox integration in automation systems	36
Integrating pdfToolbox technology using the command-line or the SDK.....	38
Where to go from here	41
callas Support.....	42

Introduction

What exactly is pdfToolbox?

pdfToolbox offers powerful, dynamic and easy ways to integrate PDF processing. It offers capabilities from quick visual inspection and fixing, to fully unattended processing of thousands of files. Using rock-solid Adobe technology, pdfToolbox provides checking and fixing of even very complex problems such as: color management, impositioning, conversion to and from PDF and more. pdfToolbox technology is used by small companies, large publishers and print providers as well as OEMs that integrate the technology in their own solution.

Let's look at this statement in a little more detail, shall we?

PDF processing

The pdfToolbox technology focuses on processing PDF files. In most cases, there will be an incoming PDF document that is checked or fixed in some way, and the result will be another PDF document. pdfToolbox can extend to non-PDF documents as well; however, it is capable of converting to and from multiple different file formats such as Postscript and EPS, image files, as well as office document file formats.

From manual to fully unattended

Callas software developed pdfToolbox as a product family; the same underlying core-technology is used in a number of different products.

- pdfToolbox Desktop allows operators to open PDF documents manually, examine them visually, and run preflight checks and fixes on them. Read the chapter on manual processing of PDF documents to know more ([Working with PDF documents manually](#)).
- pdfToolbox Server performs unattended processing using hot folders. All files placed into a watched folder are picked up and processed automatically. Read the article on processing PDF documents using hot folders for further information ([Working with hot folders in pdfToolbox Server](#)).
- Lastly, the pdfToolbox CLI and SDK provide the means to integrate the pdfToolbox engine in web portals or other

software products. Read the article on [Integrating pdfToolbox technology using the command-line or the SDK](#) for more information about those tools.

Using Adobe technology

The pdfToolbox engine is built upon the Adobe PDF Library technology; this ensures that the PDF documents it processes are handled by the same rock-solid technology you have in Adobe Acrobat, Adobe Reader and in many of the RIPs available and in-use. pdfToolbox uses the Adobe PDF Library for transparency flattening, rasterization to images, conversion to and from Postscript and EPS and more.

In the other direction, parts of the pdfToolbox technology have been integrated in Adobe Acrobat. More specifically, the "Preflight" plug-in that can be found in Adobe Acrobat Pro, was developed by callas software. If you are familiar with it, you will recognise much of the preflight functionality in pdfToolbox Desktop already.



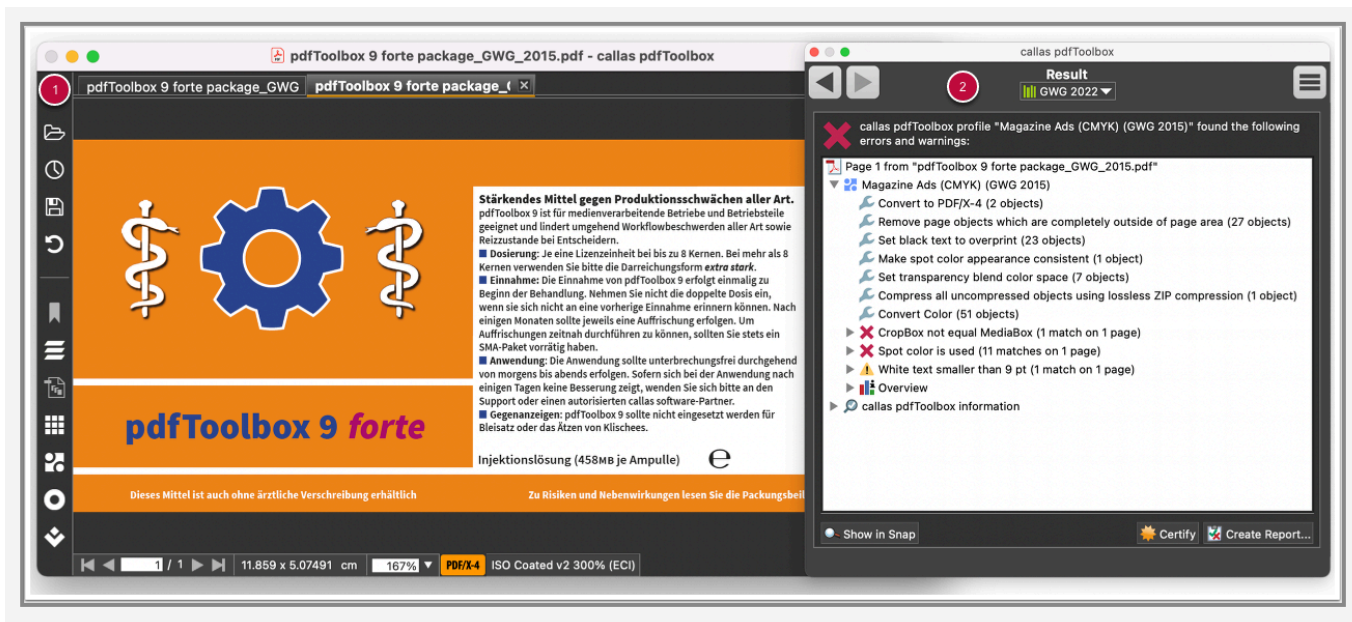
The second best way to learn something new after hands-on training is 'videos'. Enjoy this video about basics of pdfToolbox.

Common use cases with pdfToolbox

pdfToolbox is most commonly associated with preflight (quality control) of PDF documents, but it certainly is not the only way it is used by its customers. To give a brief overview of the performance spectrum of the pdfToolbox technology, various use cases are listed below.

Preflighting customer files

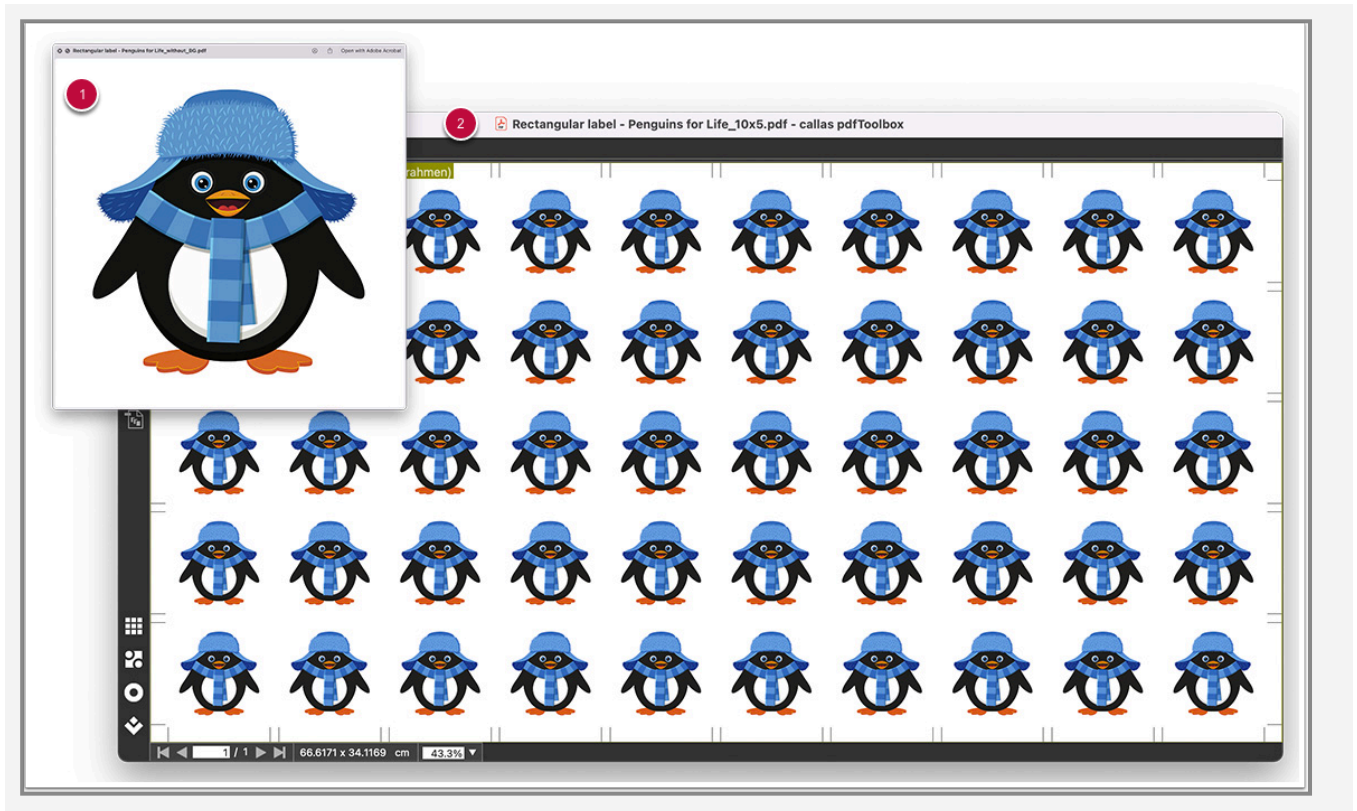
By far the most common reason to implement pdfToolbox technology still is doing quality control on incoming files and fixing common problems in those files. pdfToolbox supports virtually all preflight standards worldwide (from ISO to Ghent Workgroup, PDF/X-Ready, ...) and can quickly assess whether incoming files adhere to a given preflight standard.



1. A PDF file (magazine ad) is to be prepared for printing with a corresponding preflight profile. For this purpose, the file is checked and corrected if necessary by pdfToolbox.
2. The preflight result shows all corrections made by the software as well as any problems and warnings that occur.

Imposing PDF files

While pdfToolbox is far from a dedicated imposition tool, it does have an imposition engine on board that can handle common tasks such as creating a booklet or doing an n-Up, fill page, or Step & Repeat. It can also be configured to handle many more complex tasks.



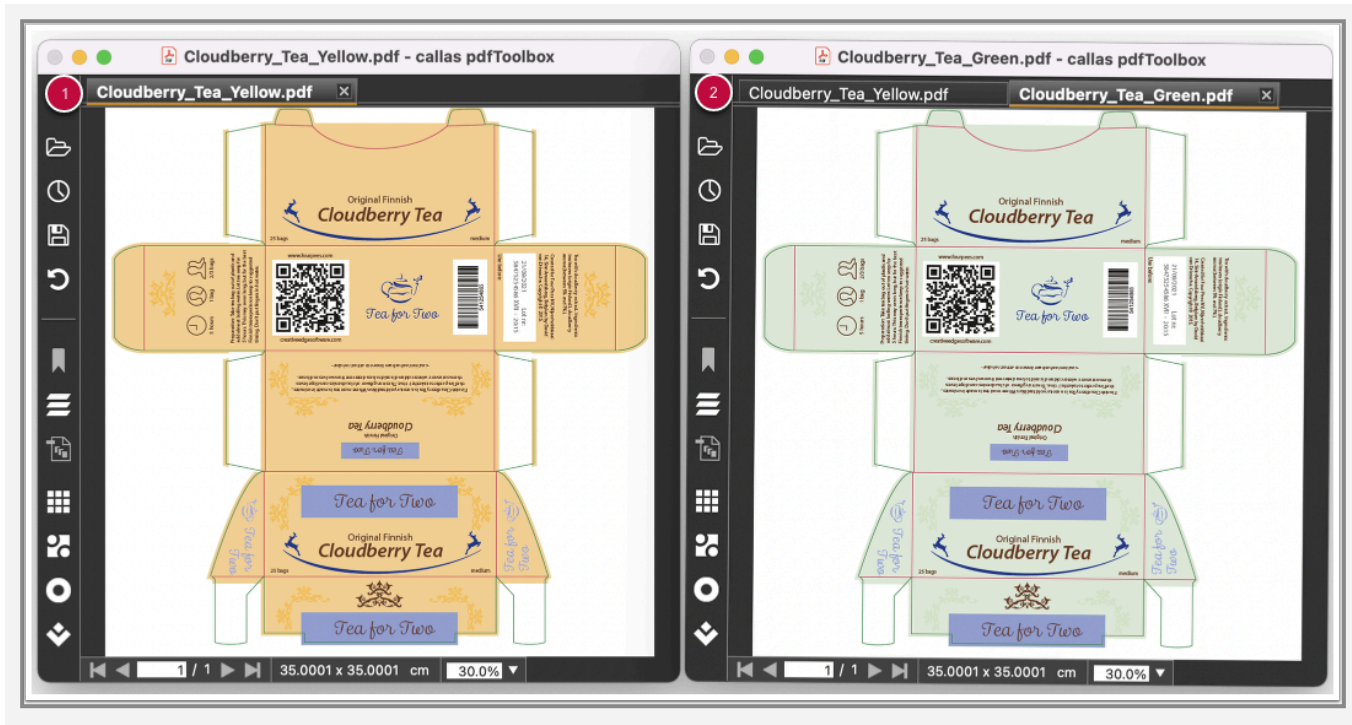
1. Shows the original PDF file.
2. With the help of the action "Step & Repeat", pdfToolbox places the PDF file several times on a new page. A distance between the individual files can be defined and cut marks can be added.

More detailed information can be found in the online documentation under the chapter: [Impose](#).

Performing color management

pdfToolbox can handle a lot of color related tasks; its capabilities can largely be divided in three big areas: with pdfTool-

box you can easily prepare your PDF files for the desired output condition. The color conversion can be done either with ICC profiles or via DeviceLink profiles. pdfToolbox also provides a wide range of functions for working with spot colors. For example, the naming of spot colors can be changed, assigned to another spot color or converted to CMYK.



1. Shows the original PDF file.
2. Using pdfToolbox, the spot color of the packaging was re-colored and renamed to another spot color.

Read more about color management in the online documentation under the chapter: [Color conversion](#).

Decorating PDF files

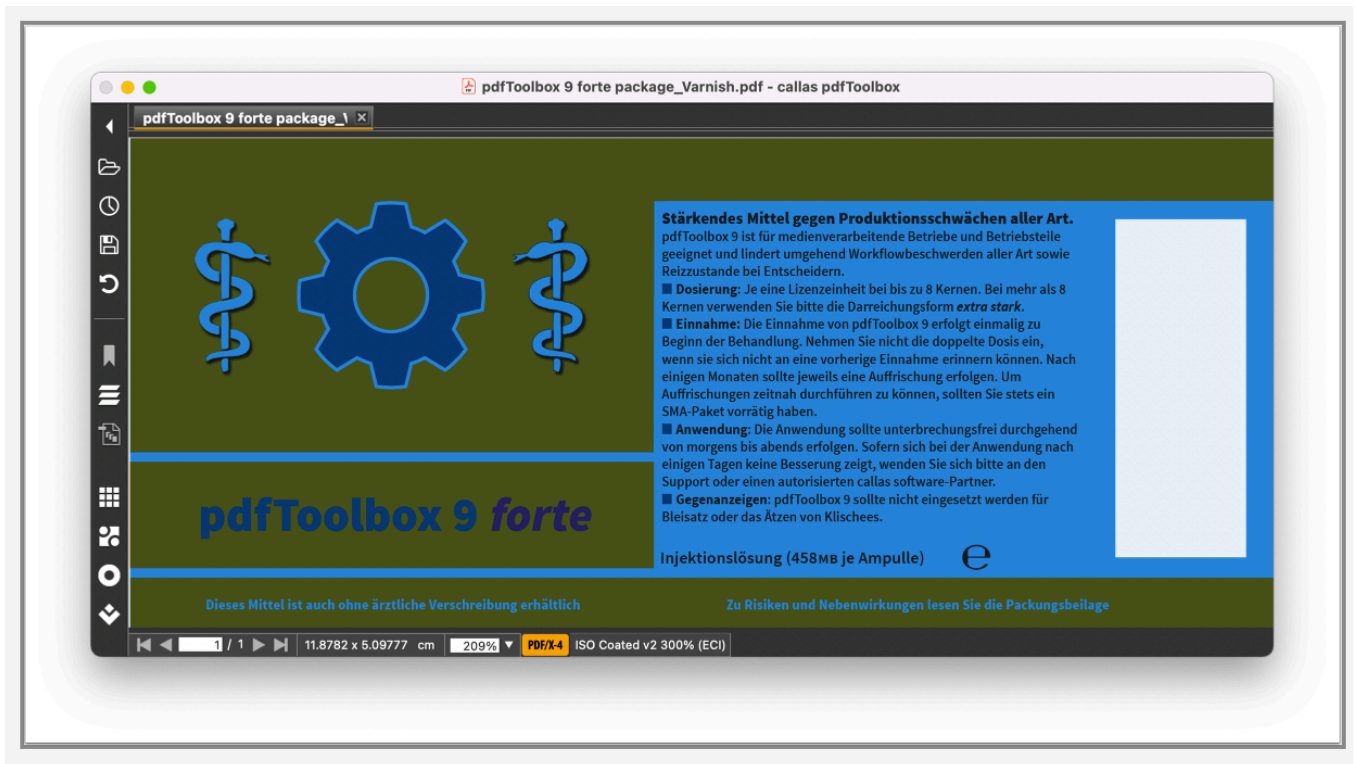
Very often, PDF files are 'decorated' (have additional content added) while they are being prepared for production. This can include trim marks, additional white space, color bars, textual information about the customer or file, bar codes... pdfToolbox has a range of different tools to add this necessary information to the PDF file.

In the following, three common use cases related to decorating are shown.

Decorating PDF files: creating varnish

With pdfToolbox it is possible to add a precisely fitting varnish to the PDF file. The pdfToolbox technology uses the already existing content or page information. This means that a varnish can be applied either to the entire page or only to specific objects.

In the following example, a partial varnish was created for all printable elements of the page except of the barcode area on the right-hand side because it is difficult to read barcodes with a glossy surface.



How to create varnish in pdfToolbox can be found in our on-line documentation under the chapter: [Shapes](#).

Decorating PDF files: generate bleed from page content

If a document does not contain a bleed, fine white lines can appear on the edges of the print product when the sheets are cut. To avoid this a bleed should be added. With pdfToolbox

it is possible to create a customized bleed from the page content. This can be done by mirroring, repeating pixels or stretching the content at the edges of the page.

In the following example, a bleed of 3 millimeters was created on all edges.



You can find a step-by-step instruction in the online documentation under the chapter: [Generate bleed from page content](#).

Decorating PDF files: creating a dieline

The pdfToolbox can be used to add a dieline to the PDF file based on the page content or page geometry. The dieline can be created as a spot color as well as on a new layer.

In the following example, a 1 mm thick dieline in the spot color "Dieline" was created for the PDF file.



How to create a dieline with pdfToolbox can be found in the online documentation under the chapter: [Shapes](#).

Working with PDF documents manually

What is pdfToolbox Desktop?

pdfToolbox Desktop is an interactive application to work with PDF documents. It gives access to all of the pdfToolbox functionality in a non-automatic mode. The application can be used in two distinct ways (both ways are available when you install and license the software – how you use it, is up to you).

pdfToolbox Desktop standalone

The standalone version of pdfToolbox Desktop is an application that can run on any computer without having any other software from Adobe (or anyone else) installed. It exists on Mac OS X and Windows and allows you to open PDF files and work with them in a separate application fine-tuned to be easy to use.



1. The main window where you open PDF documents and work with them.
2. The Switchboard window with easy-to-use tools.

pdfToolbox Desktop plug-in

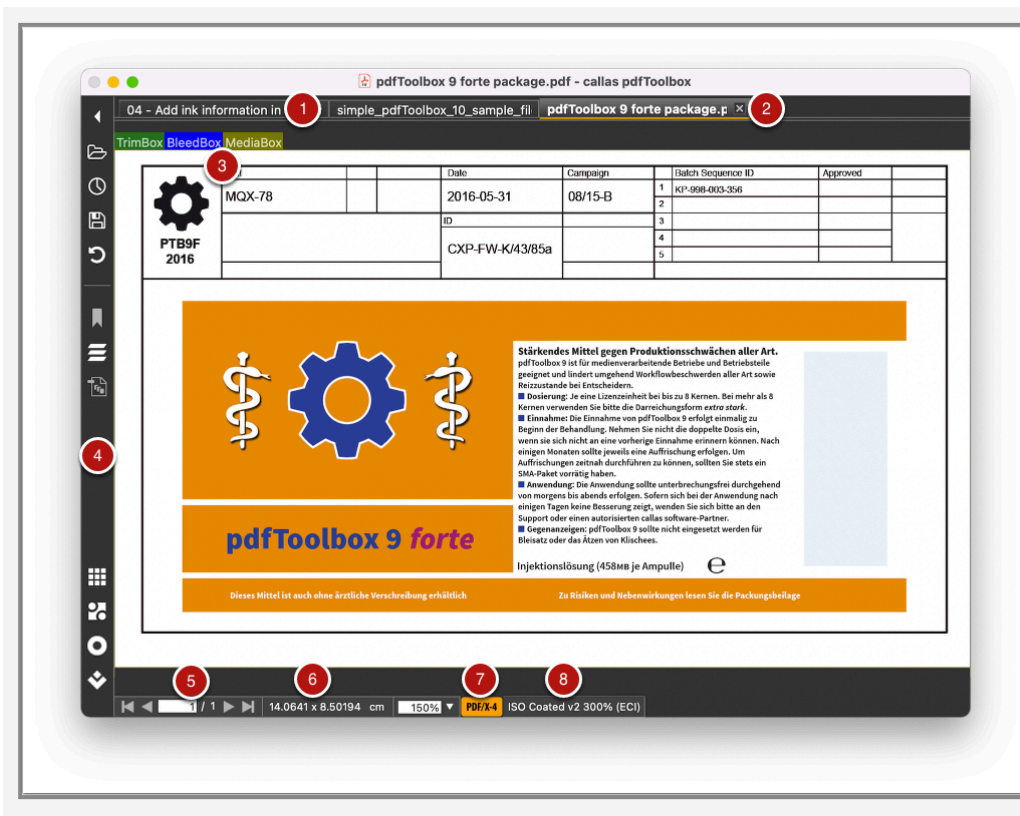
The **plug-in** version of pdfToolbox Desktop gives access to the same functionality of pdfToolbox Desktop standalone, but from within the familiar environment of Adobe Acrobat Pro. The pdfToolbox functionality is available from the "Plug-Ins" menu item.

Working with PDF documents in pdfToolbox Desktop

The main window

pdfToolbox Desktop standalone has a main window that you can use to open PDF documents. To open documents, you can use the "File" > "Open" menu or simply drag the PDF document on the main window. When you do so, the PDF file is previewed in the main window. This preview always:

- uses the fonts embedded in the PDF document (if they are embedded),
- shows transparencies and overprints correctly.



The main pdfToolbox Desktop window:

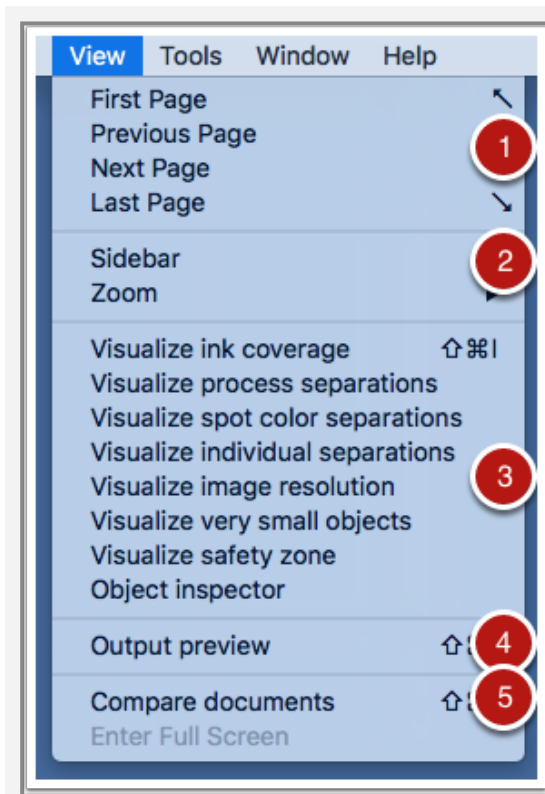
1. Each open document is shown using a tab containing the filename of the PDF document.
2. The currently shown PDF document has a highlighted tab.

3. In this example, the page geometry boxes which are defined in the PDF document, are shown as outlines over the preview. This is a preference setting.
4. The sidebar with icons allows quick access to specific areas of pdfToolbox.
5. The currently shown page number and the total number of pages in the PDF document.
6. The size (in cm) of the mediabox of the currently displayed PDF document.
7. If the currently displayed PDF document claims to be compliant with an ISO standard (through the use of its metadata), the name of the standard is displayed here.
8. If the currently displayed PDF document contains an output intent, this area shows the name of the ICC profile defined in this output intent.

Important menu items

The following are the most important menu items in pdfToolbox and where you can learn more about them.

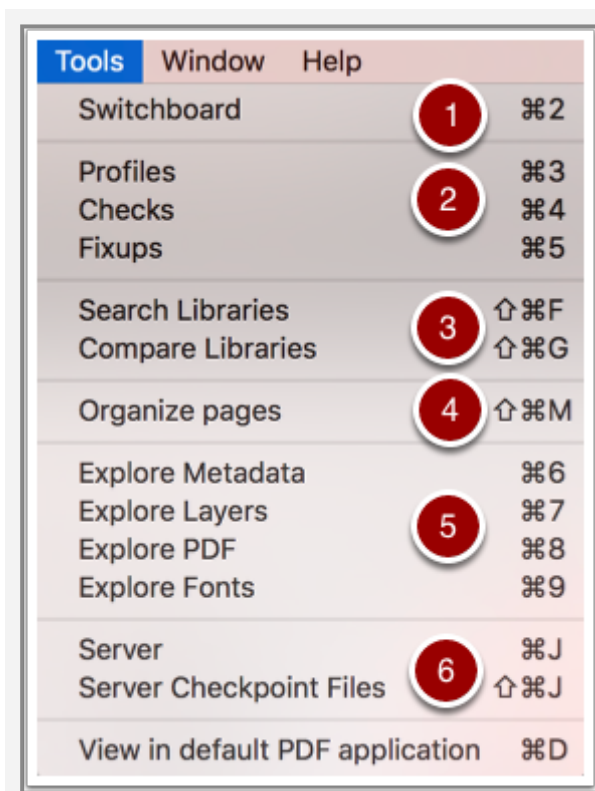
The "View" menu



Use the "View" menu to:

1. Navigate through the pages of the currently displayed PDF document.
2. Show or hide the sidebar. This is a narrow area on the left-hand side of the main window that can show additional information about a document such as bookmarks or embedded files.
3. Access the visualizer technology. Learn more about the visualizer in the article [Analysing problem files](#).
4. Show the output preview. Output preview provides a way to view the currently displayed PDF document using a specific preview ICC profile.
5. Compare two open PDF documents to see where they differ.

The "Tools" menu



Use the "Tools" menu to:

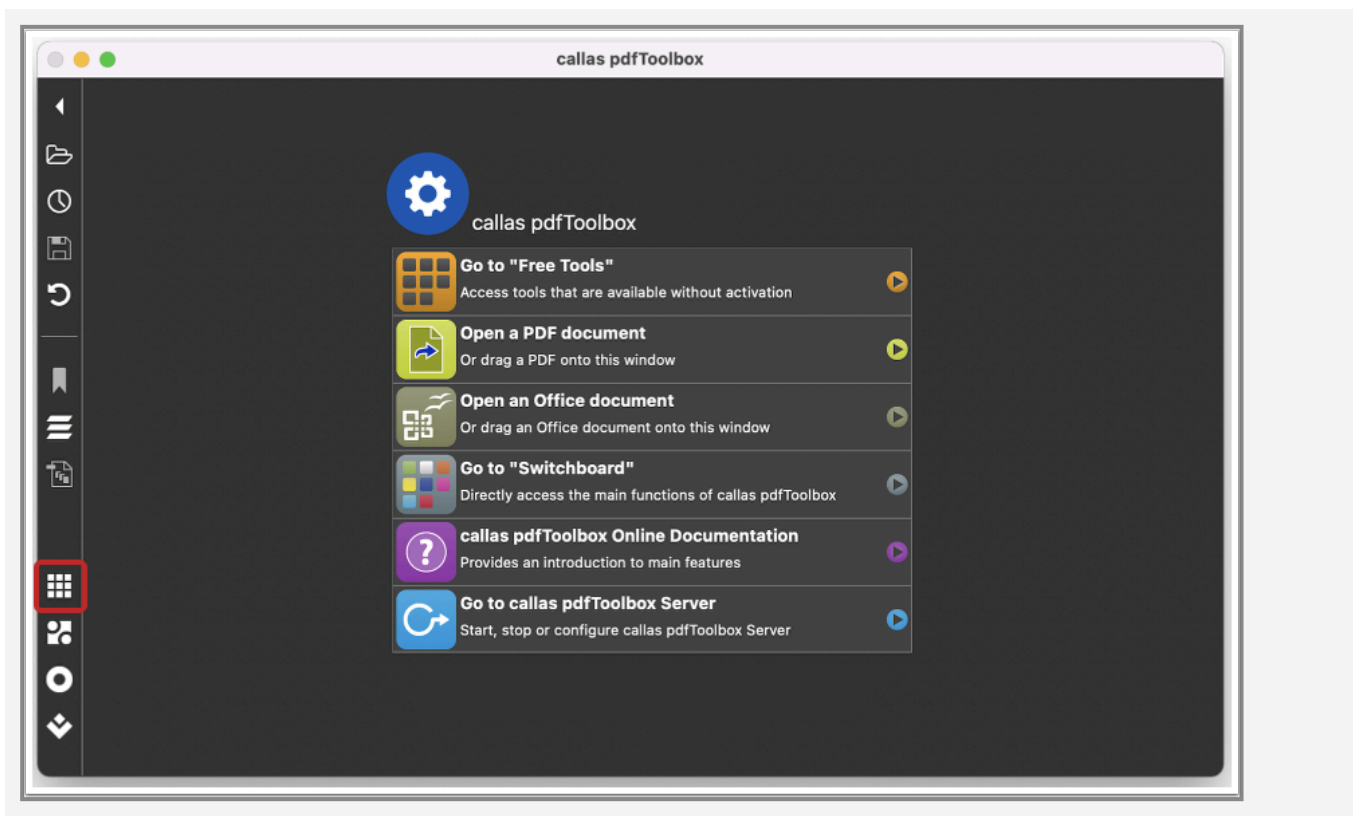
1. Open the Switchboard window; to know more about the Switchboard, read the article [The Switchboard: easy-to-use Actions](#).
2. Open the window with Profiles, Checks and Fixups. Read more about Profiles in the article [Profiles: the corner stone of pdfToolbox](#) and following articles.
3. An advanced way with different filters to Search and compare Libraries. Read [here](#) more about Libraries.
4. Organize the pages in the currently displayed document or move and copy pages between the open PDF documents.
5. Explore the details of the currently displayed PDF document. Read more about this in the article [Analysing problem files](#).
6. Access the pdfToolbox Server user interface. Read more about processing PDF documents automatically using pdfToolbox Server in the chapter [Working with hot folders in pdfToolbox Server](#).

The Switchboard: easy-to-use Actions

The Switchboard is the first window you should explore in pdfToolbox Desktop. pdfToolbox offers a set of predefined Actions. Many of these Actions are quickly accessible via the Switchboard. They ease the use of frequently needed processes like impositioning or color conversions.

The Switchboard window

To open the Switchboard if it is not visible, use the "Tools" > "Switchboard" menu item or click on the Switchboard icon which is located on the left side panel of the main window.





The following section briefly shows how to get to a particular Action.

1. The left screenshot shows the Switchboard main window with the available functionality groups.
2. With a click on a specific group symbol, all group-related Actions are displayed.
3. Depending on the Action you have selected, options may appear on the Switchboard pane; use those to configure exactly how you want the Action to be performed.
4. The "Execute" button can be used to run the Switchboard Action on the currently displayed PDF document.

An overview of all Actions contained in the switchboard can be found in the online documentation under the chapter: [Actions and their use in callas' products.](#)

Introduction in Checks, Fixups, Profiles, Process Plans and Libraries

callas pdfToolbox uses Profiles, Checks, Fixups, Process Plans and Libraries. These can be briefly summarized as follows:

- **Checks** can be used to search PDF files for specific factors, and **Fixups** allow you to alter files in line with specific criteria.
- **Profiles** are designed to link Checks and Fixups together.
- **Process plans** can be used to ensure that multiple Profiles, Checks or Fixups run one after another in a controlled fashion.
- Finally, **Libraries** enable users to bring all pre-specified categories together into personalized collections for a better overview.

Profiles: the cornerstone of pdfToolbox

Profiles (also referred to as preflight profiles) are the cornerstone of pdfToolbox. Profiles are intimately connected with Checks and Fixups. The following article gives a brief introduction to these three areas.

Profiles, Checks and Fixups

To access Profiles, Checks or Fixups you can use the sidebar on the left side of the main window. Otherwise, the "Tools" menu can be used to open one of the three menu items, which bring up the same window, but in a different tab: "Profiles", "Checks" and "Fixups". With a simple click on one icon, you can switch between these three areas:



1. The highlighted icon indicates we're looking at the "Checks" portion of the profiles window.

A Check is a single thing you want to check in a PDF document. A Check could detect the use of the Courier font, or low-resolution images, or a specific spot color, but in all cases it is a single thing you want to know about PDF documents. Checks never change your PDF document, they just look and report.

2. **The highlighted icon indicates we're looking at the "Fixups" portion of the profiles window.**

A Fixup is a single thing you want to fix (correct, change) in a PDF document. A Fixup could do a color conversion, flatten transparencies or add a QR code, but it's always one thing you want to do to a PDF document. Fixups *can* change your PDF documents, but they won't always *do*. That depends on the Fixup and the PDF document.

3. **The highlighted icon indicates we're looking at the "Profiles" portion of the profiles window.**

A Profile is a collection of Checks and/or Fixups. When you want to fix a number of problems in your PDF documents, and you want to check its quality, you build a Profile and run it on your PDF documents. That will cause the Fixups to do their work if necessary and it will run all Checks to check the quality of your PDF documents.

Further information on the subject of Profiles, Checks and Fixups, can be found in the online documentation under the chapter [callas pdfToolbox Basics](#).

Process Plans: when Profiles are not enough

When you look at the Profile window, you'll see that the list of Profiles contains two different types of items.



1. Regular Profiles, recognizable by their blue icons.
2. Process Plans, recognizable by their yellow icons.

Process Plans behave like Profiles in most ways. They can be run on PDF documents, you can import and export them, they can live anywhere in the Profiles windows (they can be in any group, not just the group named "Process Plans" as shown in the example above). But Process Plans are built differently and they can be used to solve different problems.

Why Process Plans?

Process Plans were invented to solve two very distinct problems:

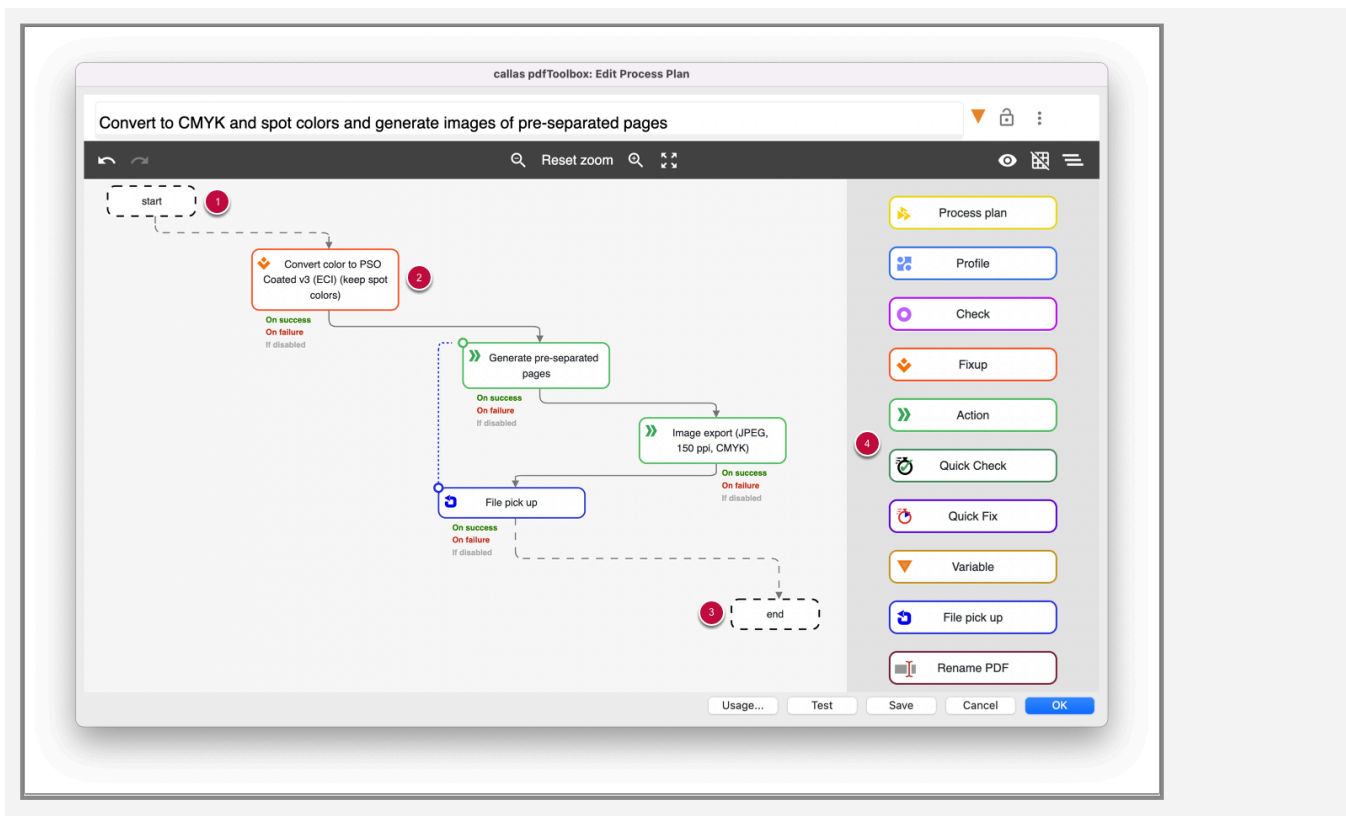
- In a Profile you can have Checks and Fixups, but you cannot control the order in which these are executed. For Checks that is not a problem, Checks do not change the PDF document they are run on so their order is irrelevant. But for Fixups this can be a real problem. Sometimes you *need* to do things in a certain order.
- Sometimes you need to be a little careful when you execute a Fixup; you really only want to perform a fix if a certain condition is met.

Process Plans make both of these things possible; you can control the order of execution and you can do conditional processing based on the result of a Check or Profile.

How is a Process Plan structured?

Since a Process Plan is a very comprehensive tool, only a brief insight into the anatomy of a Process Plan will be given below. More detailed information can be found in the online documentation under the chapter: [Process Plans in detail](#).

The structure of a Process Plan is quite simple. Users get a visual editor in which the steps can be clearly arranged and linked. The figure shows the editor "Edit Process Plan":



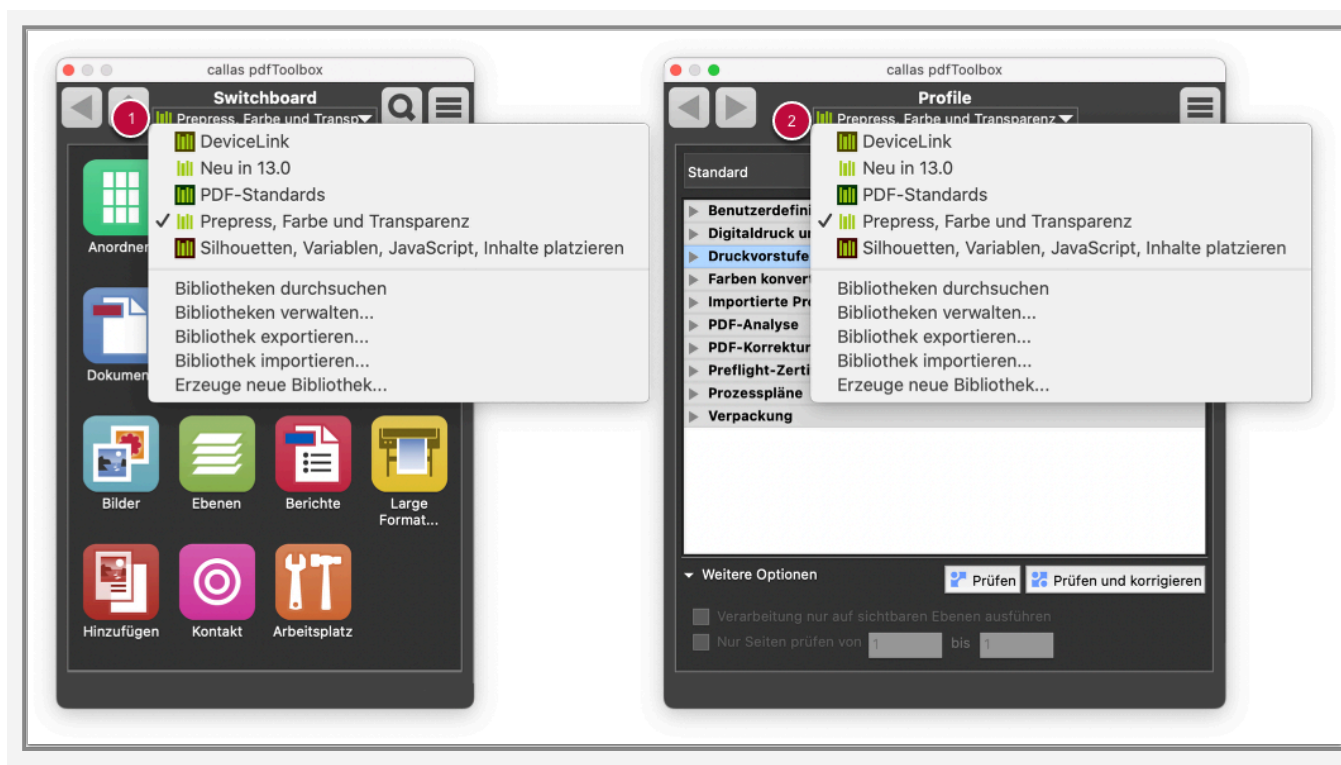
The main field in the middle shows all components of a Process Plan. Process Plans always have a start point (1), at least one sequence step (2) and one end point (3). Between start and end, sequence steps from the repertoire on the right side can be placed (4). Sequence steps can be connected to each other by connection lines. They define under which condition a subsequent and connected step is to be triggered.



Learn more about the Process Plans in pdfToolbox in this Video:

Libraries: for a broader overview

pdfToolbox offers the possibility to work with Libraries. You can use them to keep yourself organized on different projects, separate development from production profiles and much more. Libraries are not only repositories for all the items you can create but also for the predefined items that come with pdfToolbox such as Checks, Fixups, Profiles, Process Plans, Switchboard Actions, output conditions and imposition layouts.



pdfToolbox is supplied with a range of preconfigured Libraries. In the Switchboard (1), and in the overview of Profiles, Checks and Fixups (2), users can open the desired library from the pop-up menu.

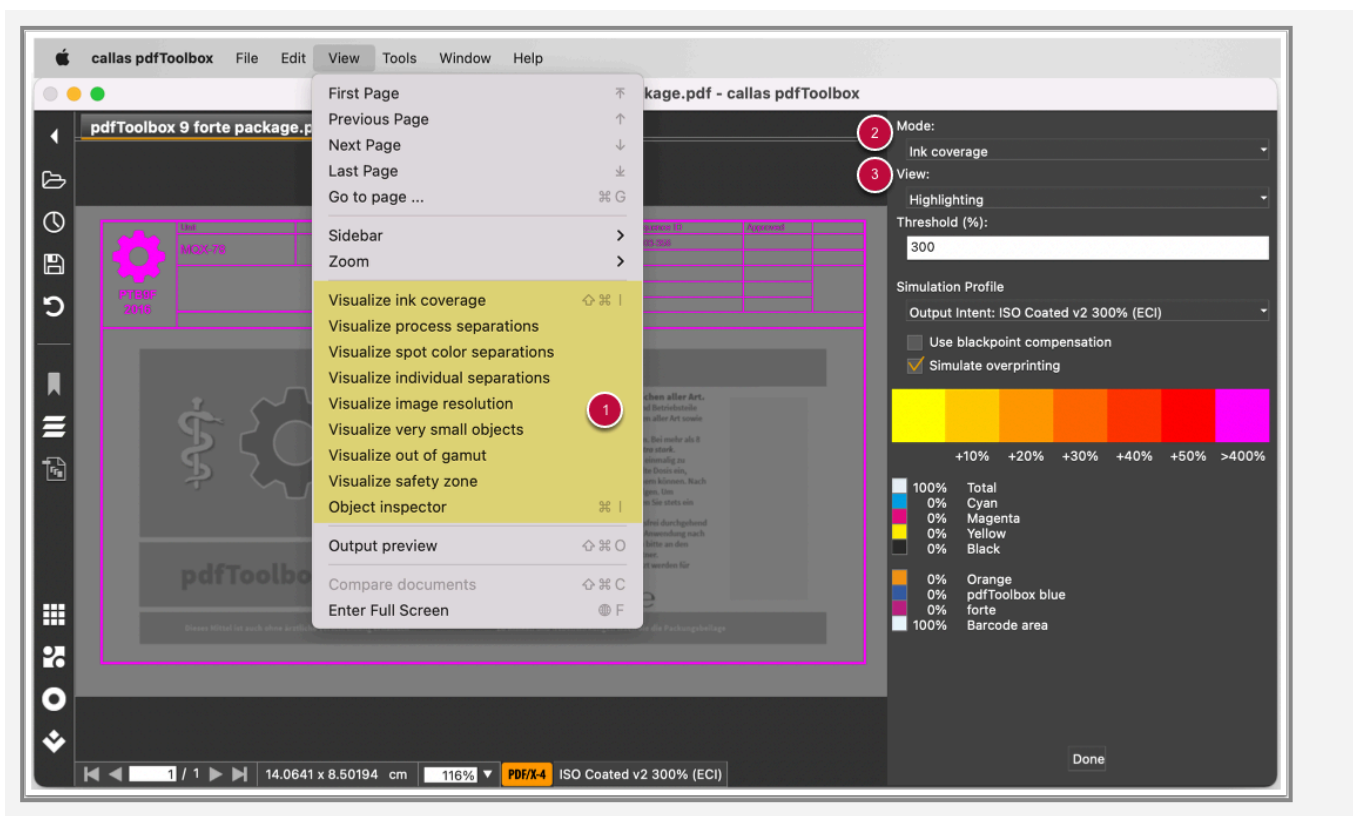
All other information relating to Libraries can be found in the online documentation under the chapter: [Libraries](#).

Analysing problem files

So what to do when you have preflighted a file and you have discovered there is or might be a problem with it? pdfToolbox Desktop provides a number of different tools to help you with the analysis of problematic PDF files; this article gives you an overview of the different possibilities.

Using the visualizer to inspect the PDF document

Sometimes knowing which objects are in the PDF document isn't sufficient, and in those cases the visualizer technology in pdfToolbox comes to the rescue. The Visualizer allows you to explore aspects of a page that may be relevant for printing purposes, such as color applications, color spaces, safety zone or page object types.



1. To access the Visualizer, use one of the "Visualize..." menu items under the "View" menu in pdfToolbox Desktop.
2. Depending on the menu item you use, a different visualizer "Mode" will be shown. You can use the pull-down

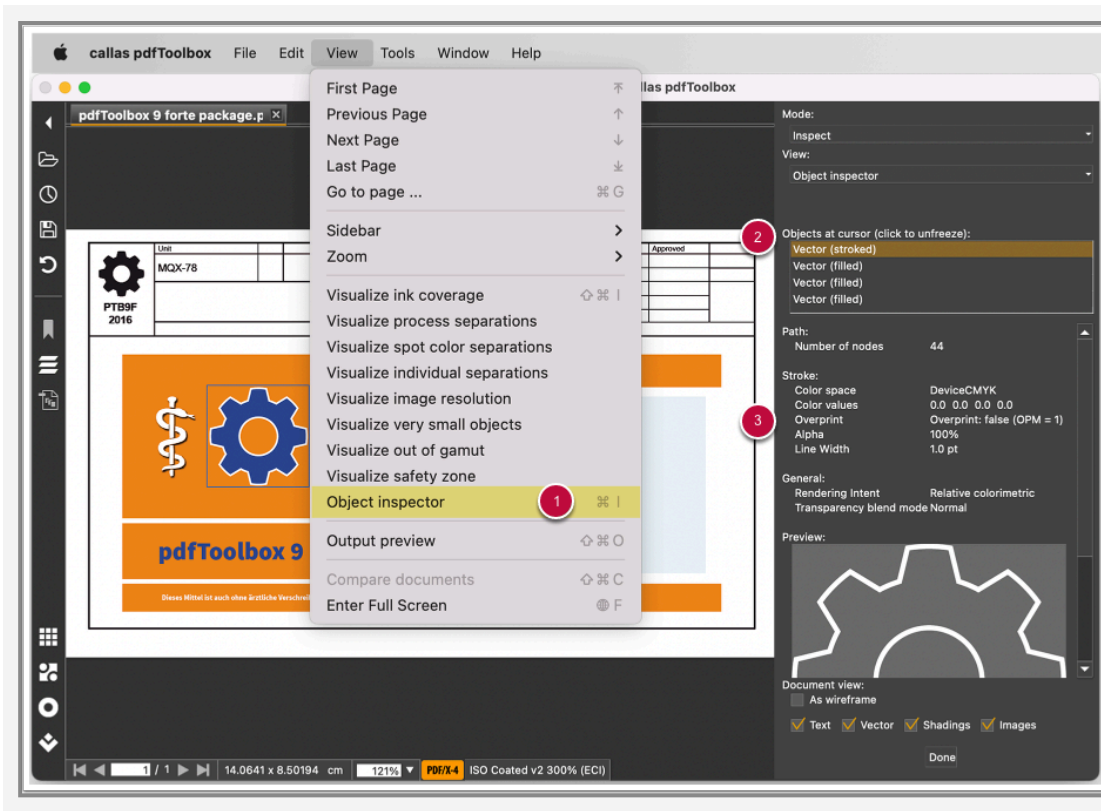
menu here to select different modes without having to go through the menus again.

- Each visualizer mode has different views. Use this menu item to select a different viewing mode.

Each visualizer mode has different additional information and options shown below. In the example you can see the ink coverage in the displayed PDF document using a thermometer color overlay. Moving the mouse over the document will list the exact color mix at each point in the document.

Using the object inspector

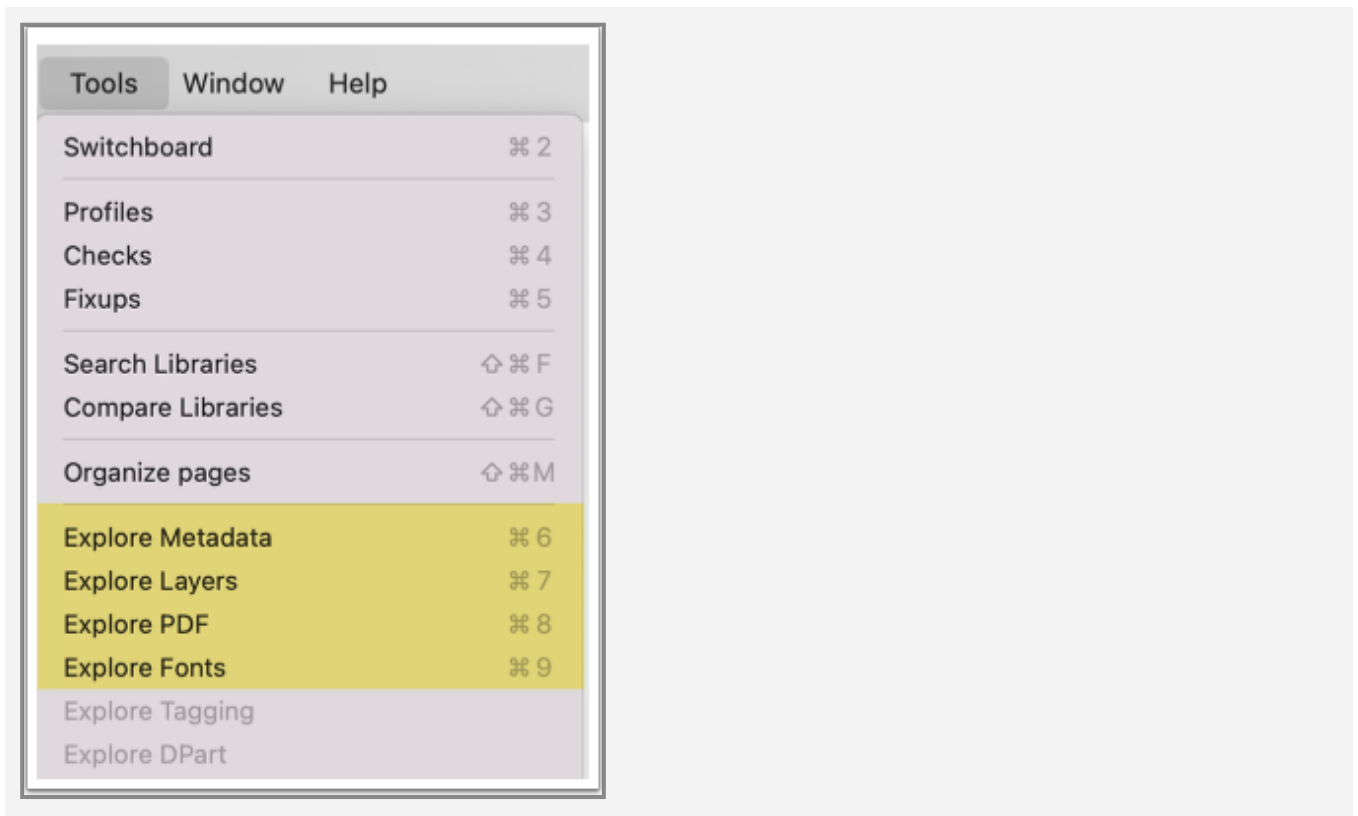
While technically part of the visualizer functionality, the "Object inspector" deserves its own mention here.



- Use the "Views" > "Object inspector" menu item to activate it.
- As soon as you move the mouse over the document, you're going to see a list of all objects on the page that are under the mouse. You can at any time click on the document to "freeze" that list so you can explore it.

3. For the object that is selected, the object inspector shows a summary of the properties of that object as well as a preview.

Exploring the technical details of the PDF document



In case you have still not found what is wrong with the document or an object, pdfToolbox Desktop has a number of much more technical tools you can use. You can find them under the "Tools" > "Explore ..." menu items but due to their more technical nature they are beyond the scope of this introductory manual.

Read more about Analysing problem files with pdfToolbox in the online documentation under the chapter: [Interactively analyse and explore PDF documents](#).

Processing PDF documents automatically

Working with hot folders in pdfToolbox Server

Previous articles of this manual talked about Profiles ([Profiles: the corner stone of pdfToolbox](#)) and Process Plans ([Process Plans: when Profiles are not enough](#)), but only on how to run those manually on PDF documents you open in pdfToolbox Desktop. Of course there are times where it is more efficient to process PDF documents automatically – and this article describes how to do that using hot folders in pdfToolbox Server.

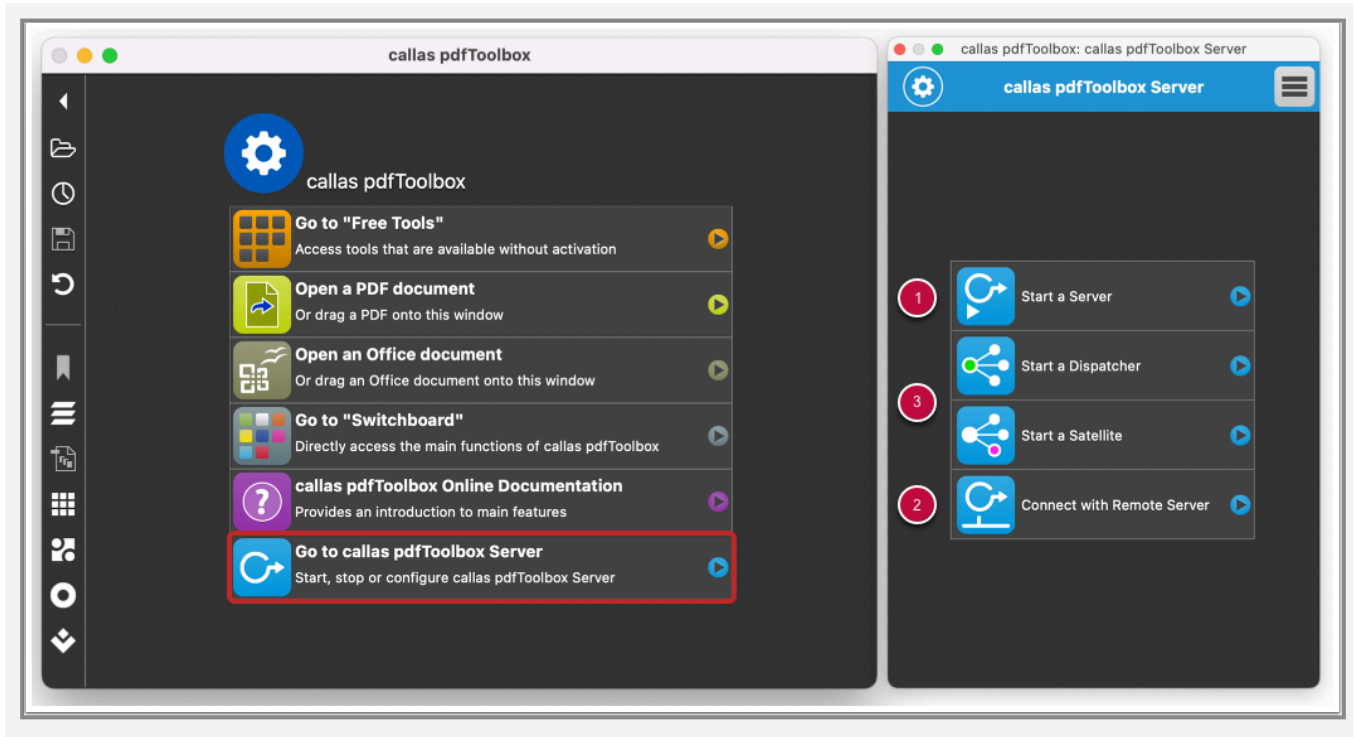
pdfToolbox Server is one of the automatic flavors of the pdfToolbox product family. It is capable of handling any number of *jobs*; each job is a hot folder setup with an input folder and one or more output folders. PDF documents dropped into an input folder are picked up and processed automatically.

Launching the pdfToolbox Server user interface

pdfToolbox Server doesn't have its own user interface; instead it can be controlled through pdfToolbox Desktop. To launch it:

1. Launch pdfToolbox Desktop.
2. Click the "Tools" > "Server" menu item.

The pdfToolbox Server control panel appears:



pdfToolbox Server can be started in different modes:

1. **Start a Server**

Use "Start a Server" to launch pdfToolbox Server *on the same machine* as pdfToolbox Desktop.

2. **Connect with Remote Server**

Use "Connect with Remote Server" to connect to a pdfToolbox Server instance that is already running either on the same machine or on a different machine on the network. This allows remote monitoring and configuration over the network. You will need to know the IP address of the machine running pdfToolbox Server.

3. **Start a Dispatcher and Start a Satellite**

With pdfToolbox Server, tasks can be distributed via a dispatcher in the network to one or more satellites, which receive the tasks, process them and send them back to the dispatcher. The dispatcher controls which tasks are to be processed by which satellite. Read more about this topic in the online documentation under the chapter: [Distributed Processing](#).

All other information relating to working with hot folders can be found in the online documentation under the chapter: [Server](#).

Advantages and limitations of using hot folders

The pdfToolbox product family provides a number of products that can automate a workflow. pdfToolbox Server does that using hot folders, but what are the advantages and disadvantages of that technique? When should pdfToolbox Server be used and when is it better to use pdfToolbox CLI or pdfToolbox SDK?

Advantages

As explained in the previous article ([Working with hot folders in pdfToolbox Server](#)), setting up pdfToolbox Server jobs is quick and easy. It's a matter of selecting which hot folder needs to be watched and what the Profile or Process Plan is that will be used to process files dropped in that hot folder. At the same time, these hot folder setups have a number of advantages:

- Error handling is done for you. If something is wrong with a PDF document, this will be intercepted for you and the document will automatically be moved into the error folder with an appropriate log.
- pdfToolbox Server automatically does parallel processing of PDF documents across all of its jobs; the number of parallel processes is limited by the license and how you configure its preferences.

Of course error-handling and parallel processing are also possible when using the CLI or SDK, but substantially more work is needed to handle this properly.

Limitations

Of course there are a few scenarios where using hot folders isn't the best choice:

- Using pdfToolbox CLI or pdfToolbox SDK closer integrations are possible. Using hot folders all you can do is drop a PDF document in a watched folder and wait until it has been processed. With pdfToolbox CLI or pdfToolbox SDK

you can get progress information for a file and you can specify where exactly the output document has to be generated.

- Not everything can be done using pdfToolbox Server and hot folders. Some commands are available only in environments where you use the CLI version or the SDK version of pdfToolbox.
- In scenarios where variables are used (read more about variables in this manual: [Variables and JavaScript: Overview](#)), pdfToolbox Server can set variables only at the job level (meaning that all PDF documents processed by a particular job will receive the same values for the variables in the used Profile or Process Plan). Using pdfToolbox CLI or pdfToolbox SDK, variables can be set differently on a file-by-file basis.

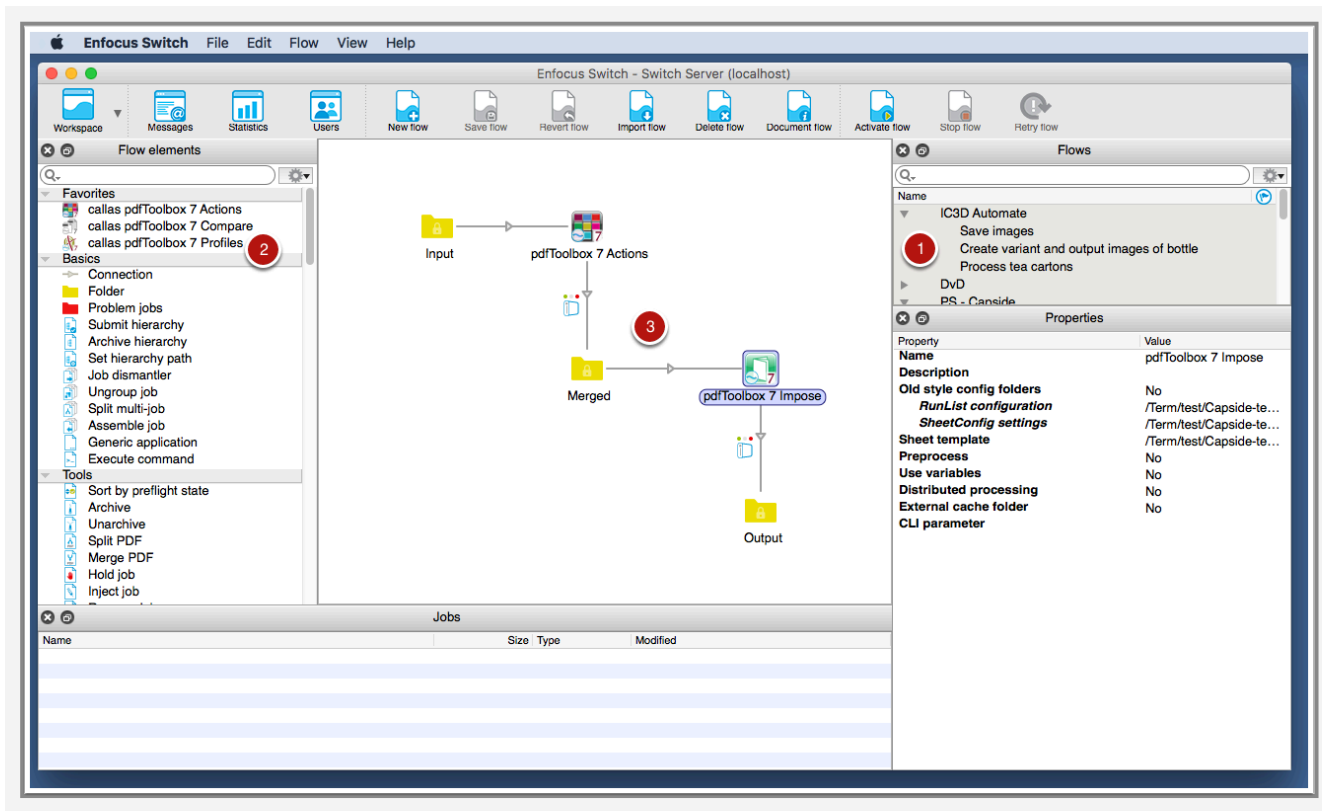
Remark that regardless of these limitations, it might still be possible to use pdfToolbox Server in a test phase before doing the actual implementation using pdfToolbox CLI or SDK.

pdfToolbox integration in automation systems

Instead of using pdfToolbox Server in hot folder mode ([Working with hot folders in pdfToolbox Server](#)), it can also be used integrated in automation systems such as Enfocus Switch. This article provides an introduction to how pdfToolbox can be used in such tools.

Enfocus Switch

Switch from Enfocus is a modular automation solution that can automate manual tasks; it includes both built-in tools (for FTP, renaming, grouping and ungrouping...) and connects to third-party tools. From the start, Switch has had a number of pdfToolbox configurators (the Switch name for the plug-ins that create a connection to a third-party tool).



1. The list of flows (the Switch terminology for a workflow that automates a series of tasks).

2. The list of built-in tools and configurators; at the top in the favorites section, three of the pdfToolbox configurators.
3. A workflow featuring two pdfToolbox configurators to automatically merge PDF files into a single file and subsequently impose them using the imposition engine built into pdfToolbox.

pdfToolbox provides the following configurators for Switch:

- Actions: runs one of the pdfToolbox actions (roughly equivalent to the functionality available in the Switchboard).
- Profiles: runs a pdfToolbox Profile or Process Plan.
- Compare: compares two incoming PDF files and determines whether they are visually the same or different.
- Impose: runs the pdfToolbox imposition engine.
- ConvertColors: runs a color conversion using pdfToolbox.

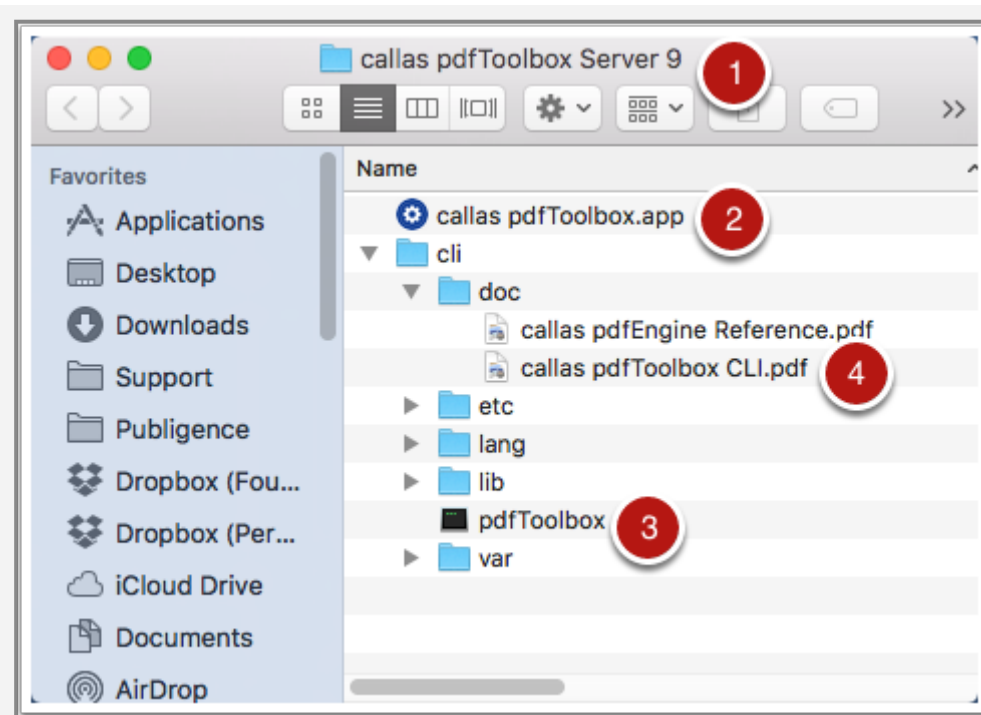
How to integrate callas pdfToolbox Server with Enfocus Switch can be found in the online documentation under the chapter: [pdfToolbox Server integration in automation systems](#).

Integrating pdfToolbox technology using the command-line or the SDK

Sometimes closer integration than simply hot folders is needed. In that case pdfToolbox provides integration through the command-line (with pdfToolbox CLI) and integration through the SDK (with pdfToolbox SDK).

pdfToolbox CLI

pdfToolbox CLI is available on MacOS, Windows and Linux. On all platforms it provides the same functionality; on top of that, the command-line is very complete, very flexible and extremely fast. To get going with pdfToolbox CLI, install pdfToolbox Server on Mac OS X or Windows, or install pdfToolbox CLI on the other platforms. The pdfToolbox Server installer will install the following components:




1. The pdfToolbox Server installation folder on Mac OS X.
2. The pdfToolbox Desktop application installed by the pdfToolbox Server installer.
3. The pdfToolbox CLI application. This is the application that needs to be called on the command-line.

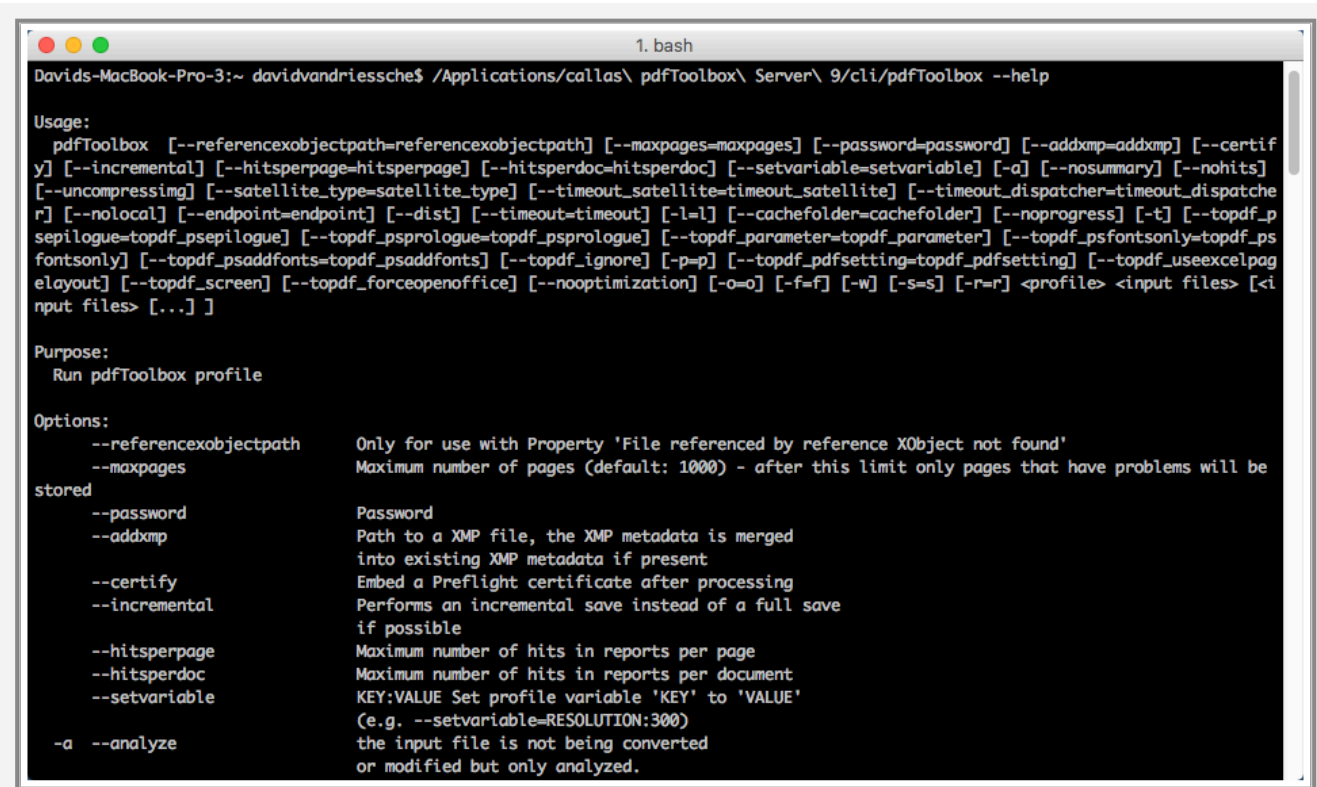
4. The documentation folder for the command-line version.

This manual can also be found online: [callas pdfToolbox CLI \(command line interface\)](#).

The easiest way to find out what you can do with the command-line, is to run the `--help` command.

```
./pdfToolbox --help
```

 Click to copy



```
1. bash
Davids-MacBook-Pro-3:~ davidvandriessche$ /Applications/callas\ pdfToolbox\ Server\ 9/cli/pdfToolbox --help
Usage:
 pdfToolbox [--referenceobjectpath=referenceobjectpath] [--maxpages=maxpages] [--password=password] [--addxmp=addxmp] [--certify]
 [--incremental] [--hitsperpage=hitsperpage] [--hitsperdoc=hitsperdoc] [--setvariable=setvariable] [-a] [--nosummary] [--nohits]
 [--uncompressing] [--satellite_type=satellite_type] [--timeout_satellite=timeout_satellite] [--timeout_dispatcher=timeout_dispatcher]
 [--nolocal] [--endpoint=endpoint] [--dist] [--timeout=timeout] [-l=l] [--cachefolder=cachefolder] [--noprogess] [-t] [--topdf_p
 sepiologue=topdf_psepiologue] [--topdf_psprologue=topdf_psprologue] [--topdf_parameter=topdf_parameter] [--topdf_psfontonly=topdf_ps
 fontonly] [--topdf_psaddfonts=topdf_psaddfonts] [--topdf_ignore] [-p=p] [--topdf_pdfsetting=topdf_pdfsetting] [--topdf_useexcelpag
 elayout] [--topdf_screen] [--topdf_forceopenoffice] [--nooptimization] [-o=o] [-f=f] [-w] [-s=s] [-r=r] <profile> <input files> [<i
 nput files> [...]]

Purpose:
 Run pdfToolbox profile

Options:
 --referenceobjectpath    Only for use with Property 'File referenced by reference XObject not found'
 --maxpages               Maximum number of pages (default: 1000) - after this limit only pages that have problems will be
 stored
 --password              Password
 --addxmp                Path to a XMP file, the XMP metadata is merged
                        into existing XMP metadata if present
 --certify               Embed a Preflight certificate after processing
 --incremental           Performs an incremental save instead of a full save
                        if possible
 --hitsperpage           Maximum number of hits in reports per page
 --hitsperdoc            Maximum number of hits in reports per document
 --setvariable            KEY:VALUE Set profile variable 'KEY' to 'VALUE'
                        (e.g. --setvariable=RESOLUTION:300)
 -a --analyze            the input file is not being converted
                        or modified but only analyzed.
```

This will provide a long help text with parameters available when working with Profiles or Process Plans (as shown in the screengrab above), and with the various "commands" usable with pdfToolbox CLI. Once you find the command you want to work with, drill-down in the help for it; for example, to know how to save images from pdfToolbox, issue the command:

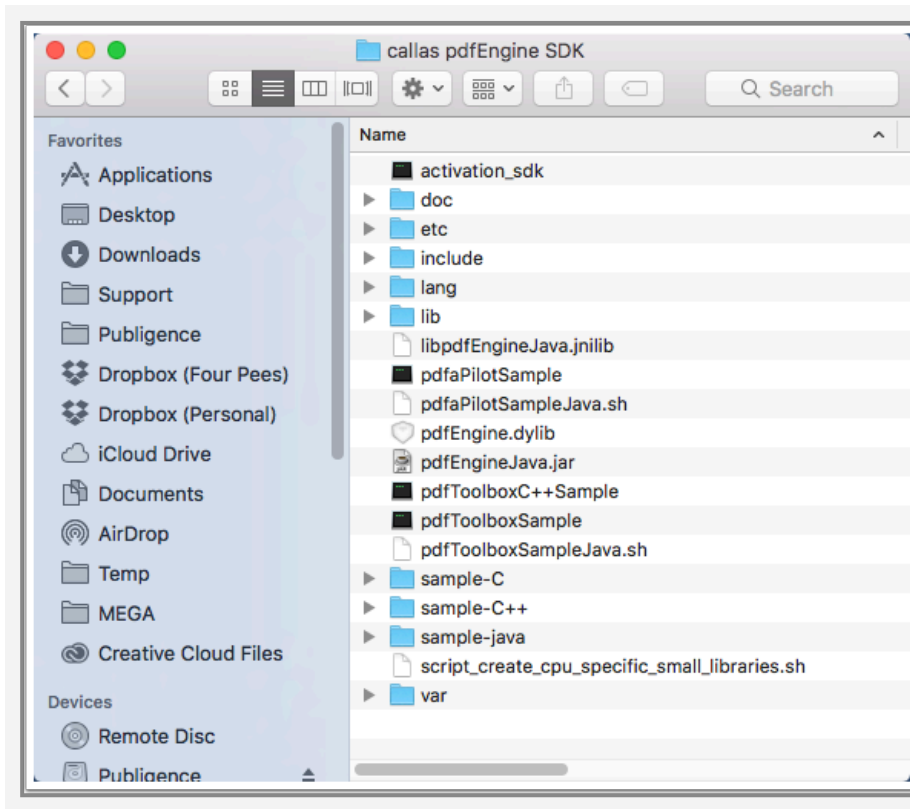
```
./pdfToolbox --help saveasimg
```

 Click to copy

Read more about pdfToolbox CLI in the online documentation under the chapter: [callas pdfToolbox CLI](#).

pdfToolbox SDK

callas pdfToolbox SDK offers a comprehensive programming interface and sample code for thorough integration of pdfToolbox technology into your own solution. It gives in-depth access for analysis and manipulation of the PDF and embedded formats such as fonts, metadata, color profiles, image compression and more.



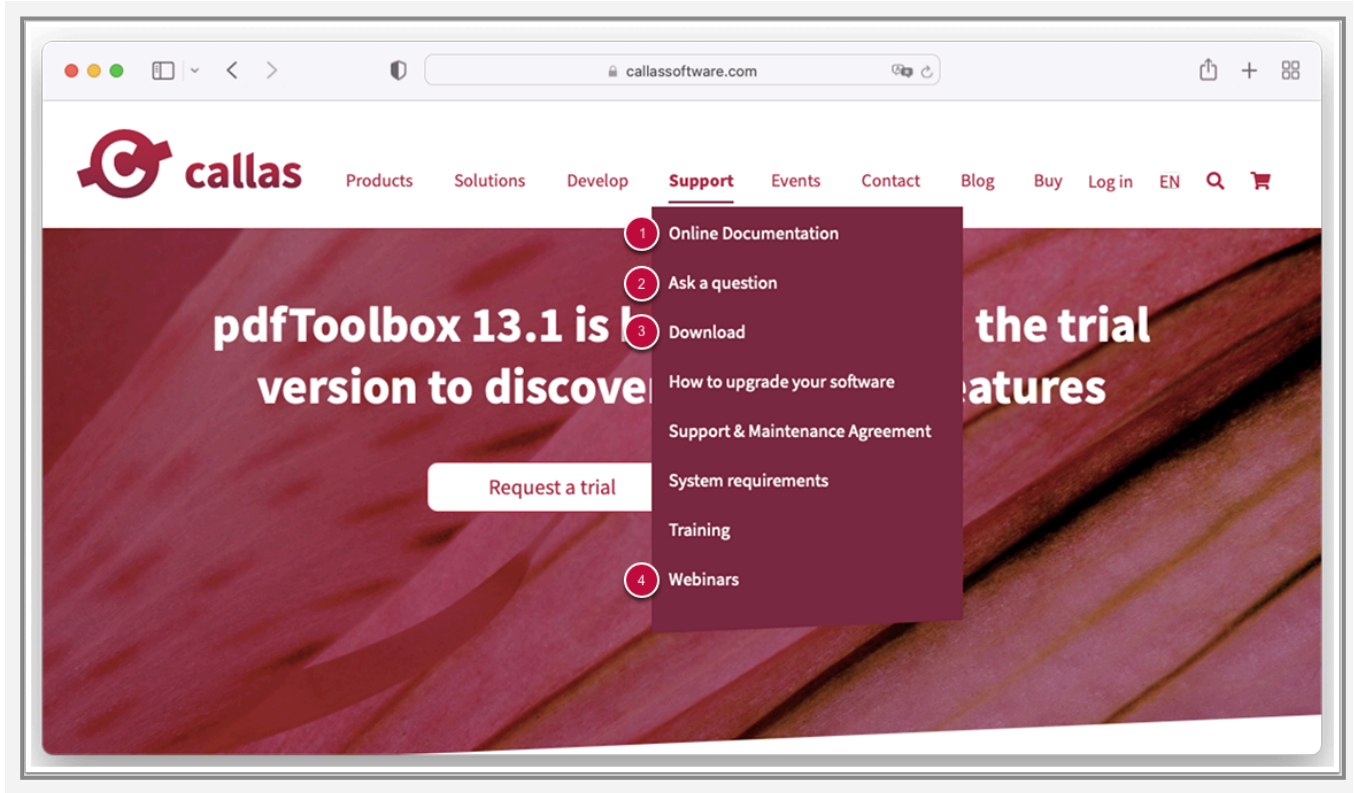
The [software development kit for pdfToolbox](#) has C/C++ as its primary development language, but it comes with APIs and example code for other development environments such as Java or .NET.

When evaluating whether you are going to use the SDK, you should keep in mind that it comes with *sample* code, but that the code is not production-ready. You will have to write actual production code to integrate the SDK in your solution.

Where to go from here

callas Support

At the end of this manual, you should have a good idea of what callas pdfToolbox can do for you. On the callas software website <https://www.callassoftware.com> you can get further guidance under the **Support** tab, as well as download a trial version.



- 1. Step by step – learn how to use callas products**
You can view our continuously growing online documentation under the "[Online Documentation](#)". Here, detailed information and step-by-step instructions for pdfToolbox can be found centrally in one place. You can also generate PDF versions of each article or export the entire documentation as a PDF manual.
- 2. Get in contact with us**
Sometimes a conversation is much easier than any documentation can be. We always appreciate receiving feedback and if you have questions about using pdfToolbox in your environment, we encourage you to get in contact. Under "[Ask a question](#)" you can fill in a form with your request and send it to us.

3. Working with the trial version

callas software provides trial versions. Under the "[Download](#)" tab you can test four pdfToolbox flavors: pdfToolbox Desktop, pdfToolbox Server, pdfToolbox CLI and pdfToolbox SDK. Use the guidelines explained [here](#) to determine which version you should download. After you have filled in the form with your details, a download link along with activation details will be sent to the e-mail address you provided. You will find further help on how to activate your trial version in the online documentation under the chapter: [Installation, activation, deactivation, updates](#).

4. Videos

On YouTube, check out the callas software channel to watch a long list of videos about the products. The videos range from webinar recordings to short feature videos. Visit [the callas YouTube channel](#).

Other questions or comments

For all other questions or comments about pdfToolbox or any of the other callas software products, please write info@callassoftware.com and we will help you as quickly as possible..