

New in pdfToolbox 11



Table of Contents

Barcode or Matrix code
Barcode symbologies supported and limitations for reading and analysis in pdfToolbox 11.0.
Read Barcode or Matrix code and determine properties
Extended bleed creation14
Generate bleed for irregular shapes1
Check and fix bleed19
Create a dieline and bleed for irregular shapes2
Create a dieline and bleed for irregular shapes with gaps in outer border
Spotify28
Why Spotify?
How does Spotify work
Spotify in callas pdfToolbox
Spotify parameters49
Process plan improvements54
When profiles are not enough: Process plans55
Actions in Process Plans64
Using Quick Check as a step in a Process Plan73
Using Process Plans in Process Plans89
Generating additional PDF output from a Process Plan step
Using the PDF input file from a previous Process Plan step
Switch on/off a Process Plan step via a Variable (11.0)10
New Process Plan step "Rename" (11.0)104
DPart metadata108
Display DPart metadata109
Use DPart metadata in a Process Plan via QuickCheck112
JavaScript enhancements114
Using object coordinates from a hit in a Process Plan11
Map (spot and process) colors using script variables11



Convert colors	124
Convert RGB to CMYK using custom tolerance for gray	125
Convert colors: Advanced settings (previously "Policies")	129
Processing black objects with Advanced settings	143
New Server functionalities	149
Using Job Tickets	150
Sidecar files and their use when processing files with a Server-Job	154
Ask-at-runtime dialog	158
Introduction	159
Working with ask-at-runtime templates	161



Barcode or Matrix code



Barcode symbologies supported and limitations for reading and analysis in pdfToolbox 11.0

Focussing on quality over quantity, this page defines those symbologies that will be supported for best results in pdfToolbox 11.0. More symbologies may follow at a later stage, also driven by your feedback/requests. If you think there is something missing from the below list and it might be of interest to you, please let us know in the comment section below or contact our support.

Code type	Notes
EAN_13	EAN-13 is an extension of the UPC-A barcode symbology that usu- ally carries a GTIN-13. Also known as: EAN, UCC-13, European Arti- cle Number, International Article Number, JAN, JAN-13, IAN, WPC, SAAN, UCCET, ABAC, BCCI, ICA, MANA, KANC, ANA, ANC.
ISBN	An ISBN barcode is a variant of EAN-13 that is used to identify books. Also known as: ISBN-13, International Standard Book Num- ber, Bookland EAN-13. Standards: ISO 2108, ISO/IEC 15420, BS EN 797, GS1 General Speci- fications.
ISMN	Variant of EAN-13 with a prefix 979 that is used to identify printed music. Also known as: International Standard Music Number, ISMN-13. Standards: ISO 10957, ISO/IEC 15420, BS EN 797, GS1 General Specifications
ISSN	An EAN-13 with prefix 977 used to identify periodicals. Also known as: International Standard Serial Number. Standards: ISO 3297, ISO/IEC 15420, BS EN 797, GS1 General Speci- fications
EAN_8	EAN-8 is derived from the EAN-13 barcode symbology and is de- signed for small packaging. It usually carries a GTIN-8. Also known as: UCC-8, JAN-8.

Supported Barcodes



Code type	Notes
UPC_A	The UPC-A barcode sym- bology is used for identifi- cation of retail goods at point of sale inside of the US. It usually carries a GTIN-12. Also known as: UPC, UCC-12, Universal Product Code.
UPC_E	UPC-E is a compacted form of the UPC-A barcode symbology that usually carries a GTIN-12. tandards: ISO/IEC 15420, BS EN 797, GS1 General Specifications.
CODE_39	Also known as: Code 3 of 9, LOGMARS, Alpha39, USD-3, USS-39. Standards: ISO/IEC 16388, ANSI/AIM BC1 - USS Code 39, BS EN 800, MIL STD 1189
PZN7/PZN8	Used for pharmaceutical products in Germany
CODE_93	Code 93 is a barcode symbology designed in 1982 by Intermec to provide a higher density and data security enhancement to Code 39
Code 128	Alpha-numeric codes supported. Used extensively worldwide in shipping and packaging industries as a product identification code for the container and pallet levels in the supply chain
ITF	Also known as: UPC Shipping Container Symbol, SCS, UPC Case Code. Standards: ISO/IEC 16390, ANSI/AIM BC2-1995 USS, BS EN 801, GS1 General Specifications.
CODABAR	Widely used for applications that require serial numbers, such as management of blood banks, slips for door-todoor delivery services and member cards
RSS_14	GS1 DataBar
RSS_EXPANDED	
QR_CODE	Quick Response Code is the trademark for a type of matrix barcode or two-dimensional barcode
DATA_MATRIX	A two-dimensional code consisting of black and white "cells" or dots arranged in either a square or rectangular pattern, also



Code type	Notes
	known as a matrix
AZTEC	2D barcode. Standards: ISO/IEC 24778:2008
MAXICODE	Not fully implemented

Limitations for barcode reading

As of now, a custom rectangle on a page with multiple barcode/matrix codes is unable to find all the codes in the user defined area. The engine finds the first code from the left side of the page and shows the first result. We are continuously trying to improve the barcode reading for future versions.



Read Barcode or Matrix code and determine properties

You can read barcodes or matrix codes using pdfToolbox Action 'Read Barcode' under the Group 'Report' in Switchboard. You can either define the position and size of the custom rectangle where you want to find the Barcode or work it automatically via "Mouse selection" like in "Text" from the "Decorate" group.

This will render the selected region to a grayscale buffer at a high resolution.

The only requirement here is that the barcode or the matrix code should be orthogonal (portrait or landscape orientation).

You can exit the barcode reading mode by leaving the Switchboard Action.



The results (value, type of barcode, bar width reduction, exact position and dimension of the barcode etc.), if any barcode or matrix code is found, will be shown in a new window



from where the barcode information can be selected and copied, like the one below:

```
Symbology: QR Code
Value: Created from SVG file.
Module width: 2.225mm/6.306pt
Bar width reduction: 0.225mm/0.637pt/20.198%
Left: 170.328mm/482.82pt
Bottom: 83.488mm/236.66pt
Width: 55.181mm/156.42pt
Height: 55.181mm/156.42pt
```

We recommend using the Switchboard action instead of Profiles or Checks to read barcodes for superior results.

Determine barcode properties using pdfToolbox Checks

You can determine barcode properties like coordinates, height, module width, symbology (type), bar width reduction or width of barcode, using Checks in pdfToolbox 11 (screenshot below).

It is recommended to define an area in which the barcode (or matrix code) shall be searched by using the Property "Barcode is in area" in the Check as a second Property.



Find barcode/matrix code in area using variables

In order to find barcode/matrix code in a user defined area using variables, click on the orange triangles to define new variables (screenshot below).





	callas pdfToo	oolbox: Edit Check
When check fires, report as:	Group:	Property: Find
When check fires, report as: Find Barcode When it does not fire, report as: Explanation for this check: Custom ID (for JavaScript and Report Fire if any condition is met Invert check result Apply this check to: Page contents Appearance of annotations Appearance of form fields Soft masks Current file	Callas pol foo	oolbox: Edit Check Property: Find All glybs in the text can be mapped to Unicode. It Character has ActualText entry It Character references noted glyph It Reference note noted note note noted glyph It reference with town It reference with town It is called horizontally It is scaled horizontally It is scaled horizontally It is called horizontally It is scaled property entry to insert the property.
Embedded files	0	
This check is not locked.	× 1	
Usage		Test Cancel OK

After clicking OK and Analyse or Test button will prompt you to input X and Y offset (from the lower left corner of the document) and the width and height of the area where you want to find the barcode/matrix code.



	callas pdfToolbox: Specify variable value
nter new v	alues or leave unchanged to use the default values.
Height:	35
Width:	45
X-offset:	0
Y-offset:	0
Inspect v	variables 📔 💽 🖸
	Cancel OK

Once you input the values and click OK, the Check will find barcodes/matrix codes in the area defined by you, if any.

calla	as pdfToolbox: Test profile
EN, 3, reduction pdf	Find Barcode Image: a find TealN3_reduction.pdf Image: a find Barcode Image: a find Barcode
1/1 EAN_13_reduction.pdf 2 Log Profile Execution	28%
	Test OK

C callas

Barcode Reference Manual

If you want more information about barcodes and matrix codes in general, please download the "Barcode Reference Manual":



Barcode_Reference_EN_2015-10-30.pdf



Extended bleed creation



Generate bleed for irregular shapes

In pdfToolbox 10, you were able to <u>create bleed</u> for 'regular' pages by various methods like:

- · Adding mirrored images
- Repeating the last color pixel at the edges inside Trimbox
- Mirroring the page content

pdfToolbox 11 introduced a Fixup for generating bleed for irregular pages. Using the Fixup 'Generate bleed for irregular shapes', you can create a shape from the settings specified in the "Shape definition" setting.

A predefined Fixup is available in the "Prepress, Color and Transparency" library: "Generate 5mm bleed at dieline (Processing Steps)".

Here is how you can achieve this:



How to generate bleed for irregular shapes

1. Type 'bleed' in the Search field



- 2. Click on 'Generate bleed for irregular shapes' in the 'Type of Fixup' area and specify the settings to generate bleed.
- Repeat only: Here you can define what content do you want to repeat in order to generate bleed. In the screenshot above, the last pixel of the 'Visible page content' would be repeated.
- Usually you would e.g. not generate bleed for a die line.
- 4. Except spot colors: Specified spot colors are excluded
- Colorspace: Various colorspace parameter are available in the drop down list like 'Use OutputIntent (CMYK will be used if no OI available)', 'CMYK and spot colors', 'Grayscale' etc
- 6. If you already have some bleed and only want to add to it you can create new bleed behind existing content
- Shape definition: Define the configuration of the shape around which you want to create the bleed. There are 3 "Create shape" settings in the drop down list:
 - "From tracing page content (including white areas)": If the white areas (as opposed to areas that just look white because they are actually transparent and let the white background shine through) are considered part of the rendered page content
 - "From tracing page content (excluding white areas)": If the white areas (as opposed to areas that just look white because they are actually transparent and let the white background shine through) are NOT considered part of the rendered page content
- 8. Set bleedbox: You can either override the bleedbox on the existing one, set bleedbox if it is completely missing or ignore this setting



Testing the Fixup in Test mode



For our original test PDF as shown above and using the Test mode with the Fixup 'Generate bleed for irregular shapes', the resultant PDF looks like below:





You can watch all this and more about 'bleed' in the video below:



Check and fix bleed

pdfToolbox 11 introduces a new Process Plan for identifying and fixing bleed issues. Limitation: It will only work with PDF files where all pages have the same size.

This Process Plan is based on many years prepress experience and has been developed by calibrate (office@calibrate.at). All JavaScripts in the Process Plan are protected and cannot be displayed.



When you start the Process Plan you are asked for:

- Safety zone distance from TrimBox
- Required bleed
- Bleed tolerance
- Unit
- Create required bleed (mirroring)
- Page type (single or facing)

The first three input fields are illustrated by this diagram.





During processing pages are analyzed on all four edges. A page edge is classified as requiring bleed if

- there are objects in the "safety zone distance from Trim-Box"
 - AND
- the required bleed zone is empty or the required bleed zone is not empty but at the edge of the tolerance there are no objects

The respective page edge is then either reported or - if "Create required bleed" is selected - bleed is added via mirroring page content.

The method to add bleed can be adjusted in the Process Plan.

You can watch all this and more about 'bleed' in the video below:



C callas

Create a dieline and bleed for irregular shapes

When you create bleed for irregular shapes you will usually have a dieline or a similar object that defines the border of the shape that requires bleed.

However, if that is missing you may want to create such a dieline on the fly and you can use pdfToolbox' shapes technology for that. The attached Process Plan adds a dieline to the outer border of the shape generated from the combined objects on a page and adds bleed to it. The dieline is created as a spot color (Dieline) on a layer (Dieline) and the layer is associated with Processing Steps metadata (Structural:Cutting).

Make_bleed_and_dieline_for_irregular_shapes.kfpx

You can use the Process Plan with any PDF and any shape, but you may as well use this one.

📴 Logo.pdf

Original





Result (Dieline has been colord pink)





In <u>this article</u> you can read what you can do when the shape has gaps.

You can watch how to 'create a dieline and bleed for irregular shapes in case its missing' in the video below:



Create a dieline and bleed for irregular shapes with gaps in outer border

If a dieline is missing and the outer border of the shape that needs bleed has gaps creating bleed does not easily work.

When the original looks like this:



Normal bleed creation would generate something like





Nice, but not was is needed.

This Process Plan creates a shape of all objects on a page and adds a line to each single object. The width of the line can be adjusted.

Make_bleed_if_dieline_is_missing_and_there_ar.kfpx

In this example the size of the gaps is roughly 13 mm.





The Process Plan creates lines with a thickness of 4 x size of the gaps as a "HelpShape" on a layer and then creates a dieline 4 x size of the gaps inside of HelpShape. It then switches the HelpShape layer off and creates bleed at the dieline.







You can watch how to 'create a dieline and bleed for irregular shapes with gaps in outer borders' in the video below:



Spotify



Why Spotify?

Some printing processes do not use CMYK process inks at all, instead only spot inks are available, and sometimes even a relatively small number. Furthermore, in some cases halftoning does not work at all, and thus iit may not be feasible to overprint spots with each other to achieve mixed ink colors.

This may be due to the way the printing process works, but could also go back to cost considerations, for example when imprints on small give-away items must be as inexpensive as possible and thus use only two spot colors.

For such scenarios, print ready PDFs should only use a specific number of spot colors, and always at a 100% tint value.

How to go from arbitrary images or PDFs to print-ready PDFs using only a small number of spot colors?

As any print service provider will try not to decline a print job just because the print file has not been prepared perfectly well, the question in this context is: how to turn an arbitrary PNG or JPEG image, or an arbitrary PDF, into a print ready PDF file that only uses a specified number of spot colors, none of them overprinting each other, and each of the spot colors with a tint value of 100% – while maintaining the overall appearance of the orginal image or PDF as well as possible?

This is where the new Spotify feature in pdfToolbox comes into play. Based on a complex analysis process, and controlled by a number of user configurable parameters, it derives the spot colors that offer the best appearance match, and assigns these to the respective areas of the original PDF, in order to provide a clean looking and pleasing result.

Everything you need to know about Spotify right here in this video:



How does Spotify work

Spotify uses a combination of techniques to allow a user to produce pleasing results:

- identify areas in an image that are to be ignored
- handle artifacts, whether created by anti-aliasing, JPEG artefacts or 'impurities' in the image
- find the specified number of individual colors that best represent the whole image
- assign the best matching color to each pixel (or leave it transparent by masking it out)

Areas to ignore

Areas in an image may have to be ignored for a number of reasons:

- in most cases, white areas are to be ignored; this may include areas that are 'almost white'
- almost always, fully transparent areas are to be ignored; sometimes this also applies 'almost fully transparent' areas
- in rare cases, gray areas are also to be ignored (this is only available in Spotify mode on the command line)
- in rare cases, black areas are also to be ignored (this is only available in Spotify mode on the command line)

Before going through the actual processing, and depending on how parameters are set, pixels in certain areas will not be taken into account (and will later on be masked out).

Artefacts

Even for digitally created – and seemingly very 'clean' – images there will be artefacts, most notably going back to antialiasing and – when JPEG compression is applied – to JPEG compression block artefacts.

The biggest issue with both is that in these areas colors will be present that are not representartive of the image appearance. Just envision an image with blue square next to a yellow square. If antialising kicks in, the pixels at the border between the two will use some shade(s) of green. Nonetheless



30



the representative colors for both areas still are blue and yellow, and not green.

Now envision a yellow background with some very thin text (e.g. using Helvetice Neue Light) on it using blue: there will be many yellow pixels (from the background), some but not many blue pixels (for the text) – and a sometimes surprisingly large number of green (or greenish) pixels – for the border between the blue text and the yellow background. In some real world files, the number of green pixels my be larger than the number of plain blue pixels.

Spotify has options – based on edge detection algorithms – that make it possible to ignore such border pixels when looking for the best matching colors, but to still to take these pixels into acount when assigning the best matching colors (and in this example assigning either blue or yellow to the green or hreenish pixels).

Determine best matching colors

Spotify uses k-means clustering algorithms to identify those (usually few) colors that best represent an image. Research has shown that first looking for more colors than needed, and then condensing the list of found colors into a smaller list of just those few desired colors almost always gives better results than looking directly for the desired number of colors.

As k-means clustering by its very nature has random aspects to it, running the algorithm twice (or more often) on the same data set will yield slightly (or even not so slightly) different results each time. As a consequence, Spotify will go through several iterations and the pick the iteration with lowest aggregated difference between original image and image represented by the result colors.

Assigning spot colors to pixels

Regardless whether Spotify is used on images or PDFs containing not just images but also text and vector, the result will always be an image on a PDF page, where that image extends over the whole page area. That image

• has one channel for each spot color



- each channel is bitonal, that is it is either 100% of the given spot color, or 0%
- for each pixel, only one of the channels will be set to 100%
- for every pixel, where all channels are set to 0%, a mask is applied to make the image transparent at th given pixel.

As the only way to encode such images in PDF is to use a DeviceN color space, the created image will be a DeviceN colored multi-channel bitonal image.

Names for the spot colors are either taken from a spot color library provided by the user, or generated from the RGB values that represent the appearance of the spot color.

The alternate space for the spot colors is by default encoded using sRGB. For the Spotify mode on the command line it is also possible to use DeviceRGB or Lab as the alternate space. Using Lab has proven to cause rendering problems on at least some viewers.

This video below goes directly to 'how Spotify works':



Spotify in callas pdfToolbox

Spotify features are available in the following ways:

- as an action, which works in the same way as the fixup, but offers access only to the two most relevant parameters and thus is easier to use
- as a fixup, with access to most parameters that can be configured
- as an inspection tool, with access to most parameters, offering a way to play around with the parameters in an ad hoc manner and immediately see the effect of each parameter
- for command line versions of pdfToolbox in the form of a specific Spotify mode (providing access to the complete set of Spotify parameters)

Meaning of Spotify parameters

The description of the various parameter available in the different ways Spotify may be used are available in a separate article <u>Spotify parameters.</u>



Spotify action in the Switchboard

Finding the Spotify action





	callas pdfToolbox
	Spotify Prepress, Color and Transpr
	Processes page content such that it will use only a configurable number of spot colors while aiming to maintain the overall visual appearance of the page content as much as possible.
Number of sp	bot colors
2	
Use spot	color library
Library	
CSS Cold	r Module Level 4 🔹
= -	Batch Execute
	Licensed to olaf (callas)

35



Managing spot color libraries

Choose spot color library to use for Spotify action



Setting up your own spot color libraries




🔴 🔴 🛑 🚞 Spot	ColorLibraries	
Name	 Date Modified 	Size
CSS Color Module Level 4.json	21. Oct 2019 at 14:50	24 KB
Photoshop PANTONE Swatches.ase	Today at 17:57	63 KB

Export spot color libraries ("Swatches") from Adobe Photoshop







Spotify Fixup

Create or edit a Spotify Fixup

		callas pdfToolbox: Edit Fixup
Name:		Fixup category: Type of fixup: Find
Spotify example Comment:		Image: Color spaces, spot colors, inks Pages Document info and Metadata Interactive elements and properties Document Page contents
Custom ID (for Reports): Apply this fixup to: Current file Embedded files Hide in list of Fixups	-	 Spotify: Number of spot colors: 2 Spot color library: Spot color library: CSS Color Module Level 4.json Parameter: Alternate color space for DeviceN image: ICC based sRGB; Number of iterations for color means detection: 3; Resolution in ppi: 300; Number of color means used as basis for color selection: 10; Maximum numbers of data points used in KMeans algorithm: 100000; Alpha threshold are ignored: 0; White threshold are ignored: 0; Colors with distance to white below threshold are ignored: 0; Colors with distance to the color below threshold are ignored: 20; Enable edge detection: No; Edge width: 1; Edge threshold: 0
Usage		Test Cancel OK





Spotify parameters

Spotify parameters		
Alternate color space for DeviceN image:	ICC based sRGB 👻	
Number of iterations for color means detection:	3	▼
Resolution in ppi:	300	▼
Number of color means used as basis for color selection:	10	▼
Maximum numbers of data points used in KMeans algorithm:	100000	▼
Alpha threshold - pixels with alpha below threshold are ignored:	20	▼
White threshold - pixels with distance to white below threshold are ignored:	0	▼
Colors with distance to other color below threshold are ignored:	20	▼
Enable edge detection		
Edge width:	1	▼
Edge threshold:	0	▼
	Cancel	K



Managing spot color libraries

Choose spot color library to use for Spotify action

Name:	Fixup category:		Type of fixup: Find	
Spotify example Comment:	All Color spaces, spot or Pages Document info and M Interactive elements a Document Page contents	olors, inks etadata and properties	Remove UCR Set halftone dictionary Set ICC profile for Output In Set Overprint and Knockout Set Rendering Intent Set transparency blend colo	tent(s)
Custom ID (for Reports): Apply this fixup to: Current file Embedded files Hide in list of Fixups D This firms is not looked	Number of spot colors V V Use spot color Spot color library: Parameter:	 Z CSS Color Module Level 4 CSS Color Module Level 4 CSS Color Module Level 4 Photoshop PANTONE Open folder with conf points used in KMea threshold - pixels wi ignored: 20; White ti white below thresho distance to other co Enable edge detecti threshold: 0 	Lison el 4.json el 4.json swatches.ase iguration files numbers of data ns algorithm: 100000; Alpha th alpha below threshold are nreshold - pixels with distance to ld are ignored: 0; Colors with lor below threshold re: ignored: 20; on: No; Edge width: 1; Edge	
Usage			Test Cancel	ОК





Setting up your own spot color libraries

Spotify:		
Number of spot colors:	2	-
Use spot color	library	
Spot color library:	CSS Color Module Level 4.json 👻 🔻	
Parameter:	 CSS Color Module Level 4.json Photoshop PANTONE Swatches.ase ge: ICC based eans detection: 3; 	
	Open folder with configuration files numbers of data	
	points used in KMeans algorithm: 100000; Alpha threshold - pixels with alpha below threshold are ignored: 20; White threshold - pixels with distance to white below threshold are ignored: 0; Colors with distance to other color below threshold are ignored: 20; Enable edge detection: No; Edge width: 1; Edge threshold: 0	

🔴 🔴 🌑 📄 Spo	tColorLib	raries	
Name	^	Date Modified	Size
CSS Color Module Level 4.json		21. Oct 2019 at 14:50	24 KB
The Photoshop PANTONE Swatches.ase		Today at 17:57	63 KB



Export spot color libraries ("Swatches") from Adobe Photoshop





Spotify Inspector

Open Spotify Inspector via the menu at bottom of Spotify action in the Switchboard





Spotify Inspector window



Spotify mode in the CLI version

For a full list of options and parameters available for the Spotify mode of the command line, use

./pdfToolbox --help spotify

Please not that some of the parameters listed by <u>--help</u> spotify are specific to the Spotify mode, whereas others are general parameters applicable also to other modes.

Spotify usage in pdfToolbox Switchboard, Fixups and command line here:





Spotify parameters

Quite a number of parameters control how Spotify operates. The degree to which these parameters are accessible and can be configured varies, depending on the context where the Spotify functionality is used. The Spotify Switchboard action only offers access to two parameters: number of spot colors to create, and whether and which spot color library to work against. The Spotify fixup, the Spotify Inspector and the Spotify mode available on the command line on the other hand offer access to all or most parameters. There are a few parameters to which access is only provided on the command line, as they only can be put to good use in command line mode. Below a list of all parameter descriptions can be found.

Iterations

Number of iterations for color means detection (default: 3)

Spotify Switchboard action: n.a.

Spotify inspector: "KMeans:" → "Iterations"

Spotify fixup: "Number of iterations for color means detection"

Spotify mode in CLI version: --iterations

Max. runs

Maximum number of runs inside one iteration for color means detection (default: 10)

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --maxruns



Resolution

Resolution in ppi (pixels per inch) (default: 72) or "Width x Height" in pixel

Spotify Switchboard action: n.a.

Spotify inspector: Resolution

Spotify fixup: Resolution in ppi

Spotify mode in CLI version: --resolution

Max. samples

Maximum number of data points to be used by KMeans algorithm. Only has an effect if the number of pixels to be processed (possibly already constrained by --resolution) exceeds this value. Data points are selected randomly.

Spotify Switchboard action: n.a.

Spotify inspector: Max samples

Spotify fixup: Maximum number of data points to be used by KMeans algorithm.

Spotify mode in CLI version: --maxsamples

Alpha threshold

Alpha threshold. Pixels with alpha < threshold are ignored. An alpha value of 0 means *fully transparent*, and a value of 255 means *fully opaque* (default: 0)

Spotify Switchboard action: n.a.

Spotify inspector: Ignore pixels with alpha below threshold

Spotify fixup: Alpha threshold - pixels with alpha below threshold are ignored

Spotify mode in CLI version: --alphathreshold



White threshold

White threshold. Pixels with distance to white below threshold are ignored. If value is negative the threshold is applied to gray instead of RGB (default: 10).

For threshold values prepended with a minus sign, computation is done via a gray value derived by the formula YCbCr Luma = 0.299*r + 0.587*g + 0.114*b, values can range from 0 to 255 (for the comparison, the minus sign is actually discarded).

For threshold values not prepended by a minus sign, the comparison is done based on the L component of the Lab value of the color against an L value of 100 for white. Values can thus range from 0 to 100.

Spotify Switchboard action: n.a.

Spotify inspector: Ignore pixels with distance to white below threshold

Spotify fixup: White threshold - pixels with distance to white below threshold are ignored

Spotify mode in CLI version: --whitethreshold

Black threshold

Pixels with distance to black below threshold are mapped to black (default: 0)

Distance is computed as dE value in Lab between color value and a Lab value of 0,0,0 for black.

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --blackthreshold



Gray threshold

Pixels with distance to gray below threshold are mapped to gray (default: 0)

Computation is done based on the a and b components of the Lab value: a < threshold && b < threshold. Values can thus range from 0 to 128.

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --graythreshold

Color threshold

Colors with distance to other color below threshold are ignored (default: 20). Distance is computed as dE value in Lab.

Spotify Switchboard action:

Spotify inspector: dE Threshold

Spotify fixup: Colors with distance to other color below threshold are ignored

Spotify mode in CLI version: --colorthreshold

Number of spot colors as result

Number of spot colors to create (default: 2) Spotify Switchboard action: Number of spot colors Spotify inspector: Colors Spotify fixup: Number of spot colors Spotify mode in CLI version: --colors 48



Number of spot colors for analysis

Number of color means used as basis for color selection (de-fault: 10)

Spotify Switchboard action: n.a.

Spotify inspector: Colors \rightarrow out of

Spotify fixup: Number of color means used as basis for color selection

Spotify mode in CLI version: --colormeans

Spot color library

Use colors from an Adobe color book file (*.acb or *.aco) or JSON format (*.json) and match colors to nearest colors from color book

Spotify Switchboard action: Use spot color library → Library popup menu

Spotify inspector: Match colors → checkbox and popup menu

Spotify fixup: Use spot color library → Spot color library popup menu

Spotify mode in CLI version: --spotcolorlibrary

Number of gray colors

Number of gray means used as basis for color selection (de-fault: 0)

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --graymeans



Edge detection

Enable edge detection.

For more information about the edge detection kernels supported and how they work, find some useful details at <u>https://en.wikipedia.org/wiki/Edge_detection</u> and <u>http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.files/MachineVision_Chapter5.pdf</u>

Spotify Switchboard action: n.a. Spotify inspector: Enable edge detection Spotify fixup: Enable edge detection Spotify mode in CLI version: --edgedetection

Edge kernel

Edge kernel to use (default: sobel):

- laplace; 3x3 Laplace operator with weight 4
- *laplace2*; 3x3 Laplace operator with weight 8
- log: 5x5 Laplacian of Guassian operator
- *sobel*; 3x3 Sobel operator (coefficients: 1,2,1)
- sobelopt; 3x3 Sobel operator (coefficients: 5,10,5)
- scharr; 3x3 Scharr operator
- canny; 3x3 Canny operator

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --edgekernel

Edge width

Edge width in pixel (default: 1)

Spotify Switchboard action: n.a.

Spotify inspector: Enable edge detection → Width



Spotify fixup: Edge width

Spotify mode in CLI version: --edgewidth

Edge mode

Edge fill mode (default: grow):

- lab; fill edge pixels with color of mean with smallest distance in Lab (dE)
- hue; fill edge pixels with color of mean with smallest distance in Hue
- hsv; fill edge pixels with color of mean with smallest distance in HSV
- grow; fill edge pixels with color of nearest non-edge pixel with smallest dE (default)

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --edgemode

Edge threshold

Edge threshold (default: 0)

Suitable value ranges depend on the chose edge kernel.

Spotify Switchboard action: n.a.

Spotify inspector: Enable edge detection → Threshold

Spotify fixup: Edge threshold

Spotify mode in CLI version: --edgethreshold

Edge threshold 2

Edge threshold (default: 0)

This second edge threshold is used only for the Canny edge detection algorithm, for details see <u>https://de.wikipedia.org/</u><u>wiki/Canny-Algorithmus</u>



Value ranges cannot easily be given since the thresholds are applied to the first or second derivation of the image brightness, thus the possible range is $[0..\infty]$

Spotify Switchboard action: n.a. Spotify inspector: n.a. Spotify fixup: n.a. Spotify mode in CLI version: --edgethreshold2

Alternate color space (--alternatecs)

Alternate color space to use for the DeviceN image in the created PDF (default: *srgb*):

- *rgb*: Use DeviceRGB as the alternate color space for the DeviceN image in the created PDF
- *srgb*: Use sRGB as the alternate color space for the DeviceN image in the created PDF
- *lab*: Use Lab as the alternate color space for the DeviceN image in the created PDF

Spotify Switchboard action: n.a.

Spotify inspector: n.a.

Spotify fixup: n.a.

Spotify mode in CLI version: --alternatecs

Output (--output)

- pdf: Create a PDF file with one page containing a single DeviceN image using spot colors
- png: Create an RGB PNG image representing the appearance of the PDF that has been or would be created using the 'pdf' option.
- json: Create a JSON formatted log file, representing the same output as seen in the log section at the bottom of the Spotify inspector window
- blob: A binary representation of the Spotify analysis result which may be used at a later stage for creation of a PDF file as would have been created using the 'pdf' option.



Spotify Switchboard action: n.a. Spotify inspector: n.a. Spotify fixup: n.a. Spotify mode in CLI version: --output

Spotify via command line in this 5 minute video:



Process plan improvements



When profiles are not enough: Process plans

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

Sh	ow all 🔹 💦 💽 🚱 👘 Find
►	Acrobat/PDF version compatibility
►	Convert colors
•	Create PDF layers
•	Custom profiles
V	Digital printing and online publishing
-	2 Digital printing (B/W)
1	Digital printing (color)
-	Conline publishing (optimize for quality)
	Conline publishing (optimize for size)
1	Imported profiles
-	PDF analysis
-	PDF lixups
	PDF/A compliance
	PDF/IA compliance
	PDF/VT compliance
	PDF/X compliance
	Place content
	Preflight Certificate
	Prepress
v	Process plans
	k Check for non-CMYK color, create a report, convert to CMYK (ISO Coated v2) and flatten
	A Check for PDF/A-1b compliance and convert if not compliant
	Scheck for spot colors, create a report and convert spot colors to CMYK
	🔥 Convert for Newspaper ads, if fails re-distill
٤,	Convert for Sheetfed offset and create preview thumbnails
1	Convert spot to CMYK (for transparency flattened PDFs)
	Separate PDF for specified page size
	💫 Remove all text information
	Split PDF into half and impose as a brochure
	Further Options

- 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: 55

- 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.

Anatomy of a Process plan

Process plans are quite simple, below is the process plan editor with its major parts:



- 1. Process plans of course have a name and description, just as profiles.
- 2. This area at the top details what you can use as the steps in a process plans. Profiles, checks and fixups are de-



scribed in this manual, actions and variables go beyond the scope of this manual but are described elsewhere in the online help.

- 3. The first step in this process plan is always executed; in this example that is a profile. Notice the up and down arrows to allow changing the order of the steps in the process plan. This allows determining what exactly the order is going to be of all of the steps in the process plan. The "On Error", "On Warning", ... options allow you to specify what needs to happen based on the result of this first step. This could simply be executing the next step, but the options can also be used to jump to a specific step in the process plan or exit the sequence altogether.
- 4. This second step is an action, which was created in the Switchboard and which will re-distill the file.
- 5. The last step in this example is the same as the first step. This is to re-execute the same profile if the first step wasn't successful and thus the re-distill action was used.

Process plans in pdfToolbox 11

With pdfToolbox 11, users get a visual editor in which the steps can be clearly arranged and linked. The figure shows the editor "Edit process plan":







Area at the top: Basic information on the process plan

1. Field for naming the process plan (appears later in the "Profiles" window)

2. Button for adding a variable to the name of the process plan

3. Button for locking or unlocking the process plan

4. Pop-up menu: Here you can store metadata on the author of the Process plan including e-mail, password and description. This description is then displayed in the "Profiles" window and helps to communicate the purpose of the process plan.

The horizontal bar below: Editing and View Modes

5. Undo / Restore / (Delete connection, not visible in this view)

6. Reduce view / Reset zoom / Enlarge view / Adjust view to window

7. Show or hide connection info

8. Enable or disable automatic alignment to the grid



9. Generate layout automatically (Note: This cannot be undone (only by "Cancel", but then all other changes will be lost))

Vertical area on the right: List of elements that can be included in Process plans

10. Process plan, Profile, Variable, Check, Action, Fixup, create PDF copy, Quick Check, pick up file, Rename PDF. "Rename PDF" has been added to pdfToolbox 11.

Main field in the middle: Area for arranging sequences

- 11. Example of a sequence step
- 12. Example for connection type

Horizontal area below: Completing the set-up Process plan

13. Before saving, you can check the process plan in practice by clicking on the "Test" button. More about the test mode in pdfToolbox can be found in the chapter "How to use the test mode".

- 14. "Cancel" editing
- 15. Press "OK" to save the process plan.



How is a Process plan structured?

Convert to CMYK and spot colors and generate images of pre-separated pages
Q Reset zoom Q II Q Reset zoom Q II Process plan Profile Variable On Natives On failure If disabled

Process plans always have a start point, at least one sequence step and one end point.

Between start and end, sequence steps from the repertoire can be placed from the right-hand side. Sequence steps are possible in the following categories:

- Process plan
- Profile
- Variable
- Check
- Action
- Fixup
- Create a PDF copy
- Quick Check
- File pick-up
- Rename PDF

Set up Sequence steps

Convert to CMYK and spot colors and generate images of pre-separated pages	callas pdfToolbox: Edit Process Plan	
Q Reset zoon Q :: Q Rest zoon Q :: Image: Convert color to PSO Control (Convert color to PSO Control (Conv	Convert to CMYK and spot colors and generate images of pre-separated pages	● ÷
<complex-block></complex-block>	ାଳ ନା 🗟 ଦ୍ୟି Reset zoom ଦ୍ୟୁ 🕻	◎ 照 =
	Cristian () Convert coor to PSO Co On success On failure If disabled If disabled If disabled The failed of the failure File pick up The failure If disabled The failure If disabled	Name + 🖻 🖍

If you click on a placed sequence step, a pop-up menu is displayed on the right side, in which all profiles / checks / Fixups / etc. available in pdfToolbox can be selected - in each case suitable for the type of sequence step.



- 1. A Sequence step can be activated or deactivated via the control field "Activate step". Deactivation is useful in cases where a step is to be skipped.
- 2. Variables for activating the Sequence step can be integrated via the small orange triangle.



Connecting Sequence Steps



Sequence steps have four nodes that can be used to connect individual steps. To do this, click on the area of the initial step and lead the displayed connecting line to the desired next step.

Connection type



If you click on a connection line, you can use the checkboxes displayed on the right-hand side to specify the conditions under which a subsequent and connected step is to be triggered (not all connection types mentioned here will necessarily be visible, as they are context-dependent):



- If successful
- In case of error
- Warning
- Info
- If deactivated

You can also carry out the following in postprocessing:

- Create a report, whereby the report type can be set.
- Create a Certify PDF

These two options are only available for certain sequence types.

We introduced the NextGenUI Process plans in pdfToolbox 11. While the usage is the same as before, the UI looks totally different from the previous one. Here you can learn more about the new Process Plan UI:



Actions in Process Plans

As well as containing Profiles, Checks and Fixups, Process Plans can also include some Actions. In this chapter, we'll show you how this can be done.

Note: Not all Actions can be used in Process Plans. Primarily, the Actions available are those which are required for certain workflows, such as for imposition.

Prepare the Action in the Switchboard



For our example, we will prepare the **Image Export** Action for use in a Process Plan.

This Action is a part of the **Document** group, which is contained in the **Essentials** library and elsewhere.



callas pdfToolbox
Saves the PDF file in various image formats.
Image format
JPEG (Quality: High)
Color space
RGB T
Image resolution
pixel (width and height)
150 × 200
Smoothing
Line art, text and images
Simulate overprinting
Batch Execute
Licensed to Christian (callas)

Here you can specify **Settings** for the Action which will later be used in the course of the Process Plan. (Among other things, the saved image file should have a fixed size; here, 150 x 200 pixels.)



callas pdfToolbox
Image export Q ■
Saves the PDF file in various image formats.
Image format
JPEG (Quality: High)
Color space
RGB *
Image resolution
pixel (width and height)
150 × 200
Smoothing
Line art, text and images
Simulate overprinting
✓ Include this action in the "Workspace"
Create new switchboard action
Batch

The **Flyout menu** at the bottom-left provides a range of options for the Action.

Here you can also find the **Create New Switchboard Action...** command.



This will open the Create Action dialog.

Here, if required you can...

- 1. ... Change the Name...
- 2. ... And edit the Description.



3. Click on the OK button to finish creating the action.



The newly created Switchboard Action Image Export 150x200 can now also be accessed through the flyout menu. (You can also rename or delete it from here.)



Utilising the newly configured Action in a Process Plan

lame:			Sequence:		
love annotation out of page area ar	nd 🔻		Profile		
urpose:			O Check		
innotations in PDE will be moved			Fixup Action	Δ	dd
Annotations in PDF will be moved out page. After that it exports the PDF as JPEG with 150x200 Pixels.			Execute powe to the PDF do	rful actions on a PDF document or export it to other formats and perform another sequence st ument afterwards.	ep
			Select a sequen	ce step in the "Sequence" list and click the "Add" button in the selected sequence entry nee step.	to
			🕹 Fixup		
	-		Name:	Move annotations out of page area, remove TrapNet, PrinterMark, 🔹 📃 🗾	- 11
		0	1 On Succe	ss: Continue with next step -	×
		•	On Failure	Continue with next step	- 11
			Action		
			Name:	Image export 150x200 *	
		0	2 On Succe	ss: Continue with next step	X
		9	On Failure	Continue with next step	
uthor:					- 11
mail:					
Unlocked -					
roup:					
Process plans	K			5	

Process Plans allow you to use sequences from the following categories:

- Profile
- Check
- Fixup
- Action
- Variable
- In the example shown, we should first use a Fixup to move any annotations in a PDF outside of the page area. (For each sequence step, you can specify what to do in case of success or failure.)
- 2. The second step uses the **Action** we prepared earlier. Under the Name category, the Action **Image Export 150x200** can be selected.
- 3. The new Process Plan should be given a meaningful Name.



- 4. It is also useful to add a **Comment**. This will appear later in the Profile list when the Profile is activated.
- 5. Click on the OK button to save the Process Plan.

Utilising the newly configured Action in a Process Plan starting pdfToolbox 11

Move annotations and export JF	EG 150x200 3	D û :
Control de la control de l	Notations ou Image export (JPEG, 150 On success On success Indiations Indiations Image export (JPEG, 150 Image export (JPEG, 150 Image export Image export Image export	 Process plan Profile Variable Check Check Action Fikup Create PDF copy Quick Check File pick up File pick up Rename PDF

- In the example, any comments contained in the PDF should first be moved outside the page area with a correction. (For each sequence step you can set what should happen in case of success or failure.)
- 2. The second step uses the action we have prepared. In the Name category, the 150x200 Image Export action is available for selection.
- 3. The new Process plan should be given a meaningful name.
- 4. A comment is also useful. The field for this is available after clicking on the pop-up symbol. The comment appears later in the profile list when the profile is activated.
- 5. The Process plan can be tested in advance by clicking on the Test button.
- 6. Click on the OK button to save the Process plan.





The new Process Plan with the associated Action is now available in the Profile window under the name **Move anno-tations and export JPEG 150x200**.

Click Check and fix-up to execute the Process Plan.







The **Results** window shows all sequence steps executed by the Process Plan, including the integrated **Action**.

Switchboard Actions available as Actions for Process Plans

Not all Switchboard Actions are available as Actions for Process Plans. The reason is, that some Switchboard actions are based on Profiles of Fixups.

The following Switchboard Actions can be used as Actions for Process Plans:

- Arrange:
 - Booklet
 - N-Up
 - Fill page
 - Impose
 - Reader spreads
 - Split in half
 - Step & Repeat
 - New page



- Split or reorder: This action in a Process Plan will create one or multiple file - but this will be written as a separate output and not as the input for the next step
- Present:
 - Handout
 - Passe partout
- Document:
 - Overlay
 - Re-Distill
 - Content export
 - Image export: This action in a Process Plan will create one or multiple file - but this will be written as a separate output and not as the input for the next step
 - PostScript export
 - EPS export
- Prepress
 - Pre-separated pages
 - Extract ICC profiles
- Layers
 - Split layers


Using Quick Check as a step in a Process Plan

When creating or editing a Process Plan, it is possible to insert a Quick Check step in the Process Plan.

For the purpose of this article, a new Process Plan is created:

Create Profile Create Process Plan
Create Process Plan
Edit Profiles Duplicate Profile Delete Profile Export Profile Import Profile Create Profile Summary
Log Profile Execution
Search Libraries Search all Libraries Manage Libraries Export Library Import Library

After entering a suitable name for the new Process Plan, select "Quick Check" from the "Sequence" list and click on the "Add" button in that list entry:



Name:			Sequence:
Quick Check example	•		Process plan
Purpose:			Profile
			>> Action
	-		Variable
			File pick up
			📩 Create PDF copy
			Ouick Check Add
Author:		0	Executes configurable high speed analysis and stores result in app.vars object.
			0
Email:	_		Select a sequence step in the "Sequence" list and click the
Unlocked •			
Group:			
Process plans	0		

A new "Quick Check" step gets inserted as the first item in the list of Process Plan steps.

- In order to configure what the "Quick Check" step is to provide as a result, either a suitable Quick Check-JavaScript variable has to be chosen in the "Name" popup list,
- 2. or a new Quick Check-JavaScript variable has to be created using the "+" icon.



Name:		Sequence:
Quick Check example		≽ Process plan
_		2 Profile
Purpose:		O Check
		🔆 Fixup
		>> Action
	-	▼ Variable
		Eile pick up
Email:		Quick Check Name: On Success: Continue with next step On Success: Continue with next step
Unlocked •		On Failure: Continue with next step
Group:		
Process plans	0	

Alternatively starting pdfToolbox 11

Select "Quick Check" from the "Sequence" list on the right and drag this step onto the area for the Process plan. In the top left field you can enter a suitable name for the new Process plan:





Delete the default connection type between "Start" and "End" (dashed line) and connect "Start" with "Quick Check" and "Quick Check" again with "End".

The Quick Check step is inserted into the process plan as an element.

- In order to set up what the Quick Check step should contribute as a result, either a suitable Quick Check JavaScript variable is selected from the "Name" dropdown menu,
- 2. or a new Quick Check JavaScript variable is created using the "+" symbol.





When creating a new Quick Check-JavaScript variable, the Edit Quick Check window is opened.

Suitable values have to be entered for "Key" (by which the variable can be referenced in the app.vars object), the "Label" (shown in the user interface when displaying a reference to the variable) and the actual "Script".

In the example shown below, a very simple configuration is used that requests that the TrimBox for all pages is retrieved.

	callas pdfToolbox: Edit Variable		
Key:	quick_check_config_example		
Label:	Example for Quick Check configuration variable		
Type:	Simple value • Script		
Scri	t:		
} Sa - - */ var a c my_ } my_	mple configuration for a Quick Check Process Plan step equests TrimBox data for all pages esult will be availale under app.vars.my_quickcheck_example my_config = { pp_vars_sub_path: "my_quickcheck_example", uickcheck_config : ["\$.aggregated.pages.page.geometry.TrimBox: true "] config		
Allo	v Browse button if feasible		
Sho	w evaluation results		
🖌 Us	age	Cancel	ОК

Click OK to save this Quick Check-JavaScript variable, and click OK again to save the new Process Plan.

Whenever this Process Plan is executed, the TrimBox of all pages gets stored in the app.vars object under my_quick_check_example.

Obviously, in this simple form the Process Plan is not very useful. The next part of the article shows a slightly extended variant of this Process Plan, that actually makes some interesting use of the information retrieved by the Quick Check step.

How to use information retrieved by a Quick Check step in a Process Plan

In order to try out the Process Plan presented below, please download it here:



Quick_Check_example.kfpx



78



Quick_Check_example_(Process_Plan_in_pdfToolb.kfpx

After importing the downloaded "Quick_Check_example.kfpx" file and importing it into pdfToolbox Desktop, it should show up as can be seen below:



In order to explore how this Process Plan has been constructed, click on the "Edit..." button next to the "Quick Check example" entry.

The "Edit Process Plan" dialog will open. It contains two steps:

- Quick Check step using a variable named "quick_check_sample_JavaScript_variable" that collects information from the current PDF
- Fixup step "Put file name + file path on upper left of all pages" that picks up the collected information and uses it to place text on each page of the current PDF

In order to have a look at the way the JavaScript variable has been set up,



3. click on the Edit button to the right of the "Name" popup.

		callas pdfToolbox: Edit Process Plan
Name:		Sequence:
Quick Check - practical demo		▹> Process plan 27 Profile
Purpose: Puts file name and file path on the top left of every page in the current PDF (using pink spot colored Courier text). File name and file path are collected using a Quick Check process plan step.		Check Fixup Action Check Gradient Section Check
Using a "Pace text" fixup in a second process plan step, file name and file path are picked up and combined into a suitable text using a JavaScript variable that retrieves the two values from the app.vars JavaScript application object's data structure.		On Failure: Continue with next step
Author: callas software - help.callassoftware.co	m	2 V Fixup
Email: support@callassoftware.com		Name: Put file path + name on upper left of all pages 2 On Success: Continue with next step Image: Second state
Group:	0	Test Cancel OK





Alternate image for Process Plan in pdfToolbox 11

		callas pdfToolbox: Edit Proce	ess Plan		
Quick Check example					● û :
e a ŝ		ପ୍ Reset zoom ପ୍	кл К 1		◎ 照 =
	Tiguick_check_sample_Jav 1 On success On failure If disabled	Put file path and name On success On failure If disabled		Name Put file path and nam	+ 🚡 🔨
				Test	ncel OK

The "Edit Variable" window will open, revealing the setup of the variable "quick_check_sample_JavaScript_variable".

The relevant part of the configuration are these two lines:

"\$.aggregated.file.name: true ",
"\$.aggregated.file.filepath: true ",

which request that Quick Check retrieves information about the file name and the file path of the current PDF. For a detailed description of the configuration set up please check out the article <u>Quick Check configuration syntax</u>.



	callas pdfToolbox: Edit Variable
Key:	quick_check_sample_JavaScript_variable
Label:	Quick Check sample
Type:	Simple value O Script
Scri	pt:
/* Si - - */ var	<pre>ample configuration for a Quick Check Process Plan step requests document info for the document and TrimBox data for all pages adds the Quick Check result to app.vars.quickcheck_sample the name of the variable does not have be 'my_config', any name can be used 'my_config = { app.vars_sub_path: "my_quickcheck_example", quickcheck_config :</pre>
} my	L "\$.params: true", "\$.direct: false", "\$.direct.Info: true", "\$.aggregated.file.name: true ", "\$.aggregated.file.file.path: true ", "\$.aggregated.file.file.path: true ", "\$.aggregated.pages.page.geometry.TrimBox: true "] config
	•
	ow Browse button if feasible
Sho	ow evaluation results
💕 Us	Cancel OK

Close the "Edit Variable" window, and thus go back to the Edit window for the Process Plan.

1. Click on the Edit icon to the right of the Name entry for the Fixup step.



ame: uick Check - practical demo urpose: uts file name and file path on the pp left of every page in the current DF (using pink spot colored ourier text). lie name and file path are ollected using a Quick Check rocess plan step. Ising a "Pace text" fixup in a econd process plan step, file name nd file path are picked up and ombined into a suitable text using JavaScript variable that retrieves he two values from the app.vars avaScript application object's data tructure.	Sequence: Process plan Profile Check Fixup Action Variable Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Select a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence step in the "Sequence" list and click the "Add" button in the selected sequence entry Celect a sequence sequence sequence
uthor: allas software - help.callassoftware.com mail: upport@callassoftware.com Unlocked roup: Imported profiles	 Fixup Name: Put file path + name on upper left of all pages On Success: Continue with next step On Failure: Continue with next step

uick Check example		<u>^</u>
		+ ÷ :
Q Reset zoom Q 💱		● 照 =
start	Name Put file path and nam ✓ ✓ Enable step	+ 🗈 🖍 🚺

The "Edit Fixup" window for the "Put file path + name on upper left of all pages" fixup will open.



The numerous fields in this fixup are filled with values that will make the contents of the "Text" field show up in the upper left of all pages of the current page using Courier font in pink spot color.

The more interesting part of this fixup at least in this context is how the contents of the "Text" is defined.

1. Click on the orange triangle popup to the right of the "Text" field.



A list of already defined variables is now shown, plus various options regarding existing or new variables.

1. Click on 'Edit "place_filepath_text"...'





The "Edit Variable" window for the variable "place_filepath_text" will open.

Important: this not the same variable as the JavaScript variable defined in the QuickCheck step (which configures that Quick Check step), rather, it is a separate JavaScript variable that makes use of the information retrieved by the Quick Check step.

As that information will only be available if the fixup gets executed as part of a Process Plan, where in a previous step a suitable QuickCheck step has been executed. Some precautions are taken in the JavaScript code that give a kind of error message "filename @ filepath: WARNING: required variables undefined" if the fixup is executed on its own or in a different Process Plan or Profile. This is managed by the if branch of the JavaScript code:

```
if (
        typeof app.vars.my_quickcheck_example
           == 'undefined' ||
        typeof app.vars.my_quickcheck_example.aggregated
 == 'undefined' ||
        typeof app.vars.my_quickcheck_example.aggregated.file
                                                                                ==
'undefined' ||
        typeof app.vars.my_quickcheck_example.aggregated.file.filepath == 'unde-
fined' ||
                                                                         == 'unde-
        typeof app.vars.my_quickcheck_example.aggregated.file.name
fined'
)
{
        var myvar = "filename @ filepath: WARNING: required variables undefined";
}
```



The intended main portion of the JavaScript variable is contained in the else branch:

```
else {
    var myvar = "filename: «"
        + app.vars.my_quickcheck_example.aggregated.file.
name
        + "»<br/>filepath: «"
        + app.vars.my_quickcheck_example.aggregated.file.
filepath
        + "»" ;
}
```

This else branch creates a string built from two variables, app.vars.my_quickcheck_example.aggregated.file.name and app.vars.my_quickcheck_example.aggregated.file.filepath, concatenated with other bits of text into a nicely formatted string.

Kev:	place_filepath_text		
Label	Poturo filo poth		
Label:	Return nie path		-
Type:	Simple value Script	Value type:	: Str
Sori	at.		
if () { } elsa } my	<pre>typeof app.vars.my_quickcheck_example == 'undefined' typeof app.vars.my_quickcheck_example.aggregated.file == 'undefined' typeof app.vars.my_quickcheck_example.aggregated.file.filepath == 'undefined' typeof app.vars.my_quickcheck_example.aggregated.file.name == 'undefined' var myvar = "filename @ filepath: WARNING: required variables undefined"; e (var myvar = "filename: «" + app.vars.my_quickcheck_example.aggregated.file.name + "soft, filepath: «" + app.vars.my_quickcheck_example.aggregated.file.name + "soft, filepath: «" + app.vars.my_quickcheck_example.aggregated.file.name + "soft, filepath: «" + app.vars.my_quickcheck_example.aggregated.file.filepath + "soft, filepath: "soft, filepath + "so</pre>	* 11 * 11 * 11 * 11 * 11 * 11	0
Allo	w Browse button if feasible		
She	ow evaluation results		

After closing this "Edit Variable" window, and then also the "Edit Fixup" window, the "Edit Process Plan" window will be shown again. Instead of also closing this window and trying the Process Plan on one of your files,



1. use the "Test" feature by clicking on the "Test" button

Name:	Sequence:	
Quick Check - practical demo	Process plan	
Purpose:	O Check	- 11
Puts file name and file path on the top left of every page in the current PDF (using pink spot colored Courier text). File name and file path are collected using a Quick Check process plan step. Using a "Pace text" fixup in a second process plan step, file name and file path are picked up and combined into a suitable text using a JavaScript variable that retrieves the two values from the app.vars JavaScript application object's data	 Fixup Action Variable Ele alak un Select a sequence step in the "Sequence" list and click the "Add" button in the selected sequence e Quick Check Name: [quick_check_sample_JavaScript_variable] In On Success: Continue with next step 	ntry
Author:	On Failure: Continue with next step	
callas software - help.callassoftware.com	🔶 Fixup	
Email:	Name: Put file path + name on upper left of all pages 🔹 🛨 📃 🗹	
support@callassoftware.com	2 On Success: Continue with next step	×
Torup:	On Failure: Continue with next step	
Imported profiles	Test Cancel OK	

	callas por looidox: Edit Process Plan	
uick Check example		
	Q Reset zoom Q 🕻 🕻	◎ 図 三
start	* sample_Jav	Name + 💽 / Put file path and name on upp 🗸



This will open the "Test profile" window – a very convenient environment to try Process Plans (or Profiles, Fixups and Checks) on a copy of a currently open PDF file without the risk of inadvertently breaking your files ("Test" processing is always carried out on temporary copies of your PDF files), and without tedious open/edit/close/apply cycles, and without any need to clean up processed copies of your PDF files.

As you can see in the screen shot some text is showing up in the upper left of the sample file used here (a PDF consisting of an empty A 4 sized page, and named accordingly).





Using Process Plans in Process Plans

pdfToolbox 10 adds a new type of step to Process Plans to allow running a Process Plan inside of another Process Plan.

		callas pdfToolbox: Edit Process Plan	
Name:		Sequence:	
Rotate file and create landscape proof Purpose:	V	Process plan Profile Check Check Action Variable File pick up	
Author:	•	Select a sequence step in the "Sequence" list and click the "Add" button in the selected sequence Fixup Name: Rotate pages to force landscape • • • • • • • • • • • • • • • • • • •	×
Email: Chlocked Group: Process plans			
		Test Cancel OK	

- 1. Double-clicking "Process Plan" or clicking the "add" button next to it, adds a Process Plan step to the Process Plan you're editing.
- 2. A Process Plan step is identified by the yellow Process Plan icon and its title.
- 3. The "Name" property has a pull down list showing all Process Plans in the current library. This allows you to select the Process Plan you want to use. The buttons on the right make it easy to create a new Process Plan, duplicate the currently showing Process Plan or edit the currently showing Process Plan.
- 4. If the result after running this Process Plan step is "success", this options allows you to choose what needs to happen next. This is similar to all other Process Plan steps.
- 5. If the result after running this Process Plan step is not "success", this options allows you to choose what needs





to happen next. This is similar to all other Process Plan steps.

Alternatively in pdfToolbox 11

Rotate file and create landscape proof	callas pdfToolbox: Edit Process Plan	
Q Reset zoom Q :: Q Reset zoom Q :: <th>Rotate file and create landscape proof</th> <th>□ ô :</th>	Rotate file and create landscape proof	□ ô :
Image: start interview Image: start Image: start interview Image: star	C C Reset zoom Q	● 照 =
Test Cancel OK	Rotate pages to force I Or success Or failure If disabled () () () () () () () () () ()	 Process plan Profile Variable Check Action Fixup Create PDF copy Create PDF copy File pick up File pick up Rename PDF

1. The type "Process plan" is also listed in the list of sequence steps. This step "Process Plan" can be added via the list with the sequence steps on the right by drag&drop.

2. A process plan step can be recognized by the yellow process plan icon.



90





3. The Name property has a pop-up menu that displays the list of all process plans that comprise the selected library. This allows you to select the Process plan you want to use.

4. The buttons above allow you to easily create a new Process plan, duplicate the currently displayed Process plan, or edit the currently displayed Process plan.

New in pdfToolbox 11





5. After the Process plan is set up, you can set the connection types to determine what is to happen afterwards. This is similar to other Process plan steps.

Remarks

- This feature allows you to put frequently used steps into their own Process Plan and re-use them easily in other Process Plans.
- Take care not to create infinite loops.



Generating additional PDF output from a Process Plan step

pdfToolbox 10 onwards, you can add a new Process Plan step to generate additional PDF output files from a Process Plan. Where before, a single Process Plan would always have a single PDF output file, using this new "Create PDF copy" step, a single Process Plan can now save additional PDF output files. See how it works in pdfToolbox 11 (go below for pdfToolbox 10 UI)



1. The process plan step is called "Create PDF copy". As with all other steps, the step can be added via the list with the sequence steps on the right using Drag&Drop.



C callas



2. The new step is included in the Process plan as shown in the figure. By default, this step results in an additional PDF file with the same name as the usual result of the complete Process plan (in the same folder where the regular result file is created).

For "Create PDF copy" a number of properties are provided in the column on the right that can be set.

3. The "Prefix" property allows you to optionally specify a prefix for the name of the PDF document created by this step.

4. The "Suffix" property allows you to optionally specify a suffix for the name of the PDF document generated by this step.

5. The "Folder" property allows you to optionally specify a folder for the PDF document generated by this step. If this is used, a folder with the specified name is created and the output file created in this step is saved in this folder.

6. The "Overwrite existing file" checkbox allows existing files to be overwritten by files of the same name created in this step. Variables are possible for this (orange triangle).



7. The option "Replace file name with prefix and suffix" allows changes to the file name. Variables are possible for this (orange triangle).

8. Finally, the step in this category can be activated or deactivated.



9. By clicking on the connection type, you can set the conditions under which the next connected step is to take place. These are available here: "If successful", "If error" and "If deactivated".



Alternatively in pdfToolbox 10 or lower

Name	Callas por looidox: Edit Process Plan Sequence:
Create proof and production file Purpose:	 Fixup Action Variable File pick up Create PDF copy Creates a copy of the current PDF as an additional output file. Suffix or prefix for the output
Author: Email: Inlocked Group: Process plans	Imame can be defined. Else: Create PDF copy Prefix: Suffix: Folder: Overwrite existing file On Success: Continue with next step On Failure: Continue with next step Test Cancel OK

- 1. The Process Plan step is called "Create PDF copy". As all other steps, double-clicking or clicking the "Add" button the right of the name, inserts the step at the end of your Process Plan.
- 2. This is how this new step looks in a Process Plan. By default, this step outputs an additional PDF file with the same name as the regular result of the complete Process Plan and next to it (in the same folder as where the regular output file is generated).
- 3. This property allows specifying an optional prefix that will be added to the name of the PDF file generated by this step.
- 4. This property allows specifying an optional suffix that will be added to the name of the PDF file generated by this step.
- 5. This property allows specifying an optional folder for the output file generated by this step. If used, a folder with the specified name will be created and the output file generated by this step will be saved into this folder.



- 6. Select this checkbox to overwrite any file with the same name as the file generated by this step.
- 7. These options allows setting up what needs to happen next based on success or failure of this step.

Remarks

- The Process Plan always generates an output PDF files, whether or not you use such a step. Using this step simply generates an *extra* output PDF file.
- You can use this step multiple times in the same Process Plan if you need to generate multiple output files at different stages of your Process Plan.



Using the PDF input file from a previous Process Plan step

In pdfToolbox 10, there is a new type of step for Process Plans that allows you to return to the PDF file you had at the input of a previous step in the Process Plan. This has the effect that you can run one or more Process Plan steps, and if you don't like the result, return to the file you had *before* you ran those steps. Below are the steps corresponding to pdfToolbox 11 (go down for pdfToolbox 10 UI steps):



- 1. This Process plan step is called "Pick up file". As with all other steps, the step can be added via the list with the sequence steps on the right using Drag&Drop.
- 2. This is how the new step appears in the Process plan.
- 3. The dashed connection type specifies which input is to be returned from which step.



	callas p	dfToolbox: Edit Process I	Plan		
in a B	Q I	Reset zoom 🍳 🏅	л У		● 照 =
	Convert fonts to curves On error On warning On info On success If disabled	ع File pi On success On failure If disabled	bick up	Transition types On error On info If disabled Post processing Create report Select	On warning On success

4. The connection types allow you to set what should happen in case of success or error in this step.

5. Available here: "On success", "On error", "On warning", "On info" and "Disabled".

99

Steps for pdfToolbox 10

	callas pdfToolbox: Edit Process Plan
Name:	Sequence:
Solve font issues with alternatives	🔶 Fixup
_	>> Action
Purpose:	Variable
	Tile pick up Add
	Replaces the currently processed PDF with the input from a previous step. If that fails because the referenced step has never been executed the current PDF is processed as set up in "On Failure".
	Celest a conjunce step in the "Conjunce" list and sligh the "Add" butter in the celested conjunce.
	select a sequence step in the sequence list and click the Add button in the selected sequence
	28 Profile
	Name: Outline fonts with visual differences check 🔹 📃 🖉
	Do not apply Fixups
	On Error: Continue with next step
7	1 On Warning: Continue with next step
Author:	On Info: Continue with next step
Email:	On Success: Exit sequence
	1 File nick up
Tunlocked 🔹	from input to step 1: Outline tonts with visual differences check
Group:	2 On Success: Continue with next step
Process plans	On Failure: Continue with next step
	Test Cancel OK

- The Process Plan step is called "File pick up". As all other steps, double-clicking or clicking the "Add" button the right of the name, inserts the step at the end of your Process Plan.
- 2. This is how this new step looks in a Process Plan.
- 3. Specifies to the input of which step you want to return
- 4. These options allows setting up what needs to happen next based on success or failure of this step.

Remarks

- You can use this step multiple times in the same Process Plan.
- When you select the first sequence step as the selection in "Pick up PDF from", this has the same effect as returning to the original PDF file you started the Process Plan on.

Switch on/off a Process Plan step via a Variable (11.0)

As of pdfToolbox 11, you can switch on/off a Process Plan step using a Variable or a simple checkbox. Instead of adding an additional Variable as a Process Plan step to manipulate the steps, it can be achieved using the variable inside the Process Plan step for convenience.

	ପ୍ Reset zoom ପ୍ ୁି ଦୁ	● 照 ≡
start I	Con error On warning On into On success If disabled	Name + Magazine Ads O not apply Fixups Enable step
		Test Cancel OK

In the screenshot above, the Profile 'Sheetfed offset (CMYK and RGB) (GWG 2015)' is the first step of our example Process Plan. pdfToolbox 11 offers a checkbox/Variable option to switch on or off this Process Plan step (marked with a red rectangle). In this example with a checked checkbox, the Process Plan step (Profile) is switched ON (Obviously, an unchecked checkbox will lead to this Process Plan step being switched off).

Assigning a variable to the Process Plan step

You can assign a variable to the Process Plan step (using the inverted orange triangle next to the checkbox). When a variable is selected, the checkbox cannot be used (a dash [-] appears in the checkbox) as shown in the screenshot below. The screenshot also shows the menu with available Variables, which is shown when you click on the orange Variables symbol between the checkbox and "Enable step".



If you run this Process Plan, it will prompt you to switch on or off the first step of the Process plan as shown in the screenshot below.



callas po	fToolbox: Specify variable value	
use the default	es or leave unchanged values.	
Enable step 1		
	Capool	OK
	Cancel	OK

 Watch an extended explanation of 'Switch on/off a Process Plan step via a Variable' here:



New Process Plan step "Rename" (11.0)

pdfToolbox 11 introduced a new Process plan step 'Rename' which, as the name suggests, renames the current PDF in the Process plan workflow.

cess Plan step "Rename" Q Reset zoom Q :: Start i
Q Reset zoom Q Check start i i Process plan start i Process plan i Rename PDF I On success i On check On success i On check i end i i Fixup i Create PDF coov
start start
Cuick Check Cuick Check File pick up Carrier PDF
Test Cancel OK





- 1. Prefix: A defined Prefix is prepended to the PDF file name
- 2. Suffix: A defined Suffix is appended to the PDF file name
- 3. Replace file name with Prefix and suffix: If set, the current file name is not included in the new file name, i.e. the new file name will be Prefix + Suffix
- 4. Overwrite existing file: If set, the destination file is overwritten, if it exists





So a setting like the one above would add "_processed" behind the file name, but before the file type suffix.

Using a file called "Yummy_Pumpkin_Yoghurt_x.pdf" would result into "Yummy_Pumpkin_Yoghurt_x_processed.pdf" like on the screenshot of the result dialog below:





Our CTO Ulrich Frotscher talks about 'Rename' step in Process Plans:



DPart metadata


Display DPart metadata

DPart metadata is associated with pages or page ranges. It was specified in PDF/VT first and is defined in PDF 2.0 as well. it is intended to be used in automation to allow for processing pages in the same PDF in different ways. This can be done in pdfToolbox using QuickCheck which is explained here...

But pdfToolbox also is as of today the only tool that can easily display such metadata.

When a PDF has DPart metadata a button is indicating this at the bottom of the pdfToolbox window.

A suite of sample files including the one below is available from the PDF Association:

https://www.pdfa.org/resource/cal-poly-pdfvt-test-suite/



Clicking on this button opens the DPart viewer.





	callas pdfToolbox: DPart Explorer
 File name: Tra Record 1 Record 5 Page Page Page Page Record 2 Record 2 Record 3 	rel V1.0.1 - 10.pdf -4 -8 ecord":2,"RecordValues":("15":"***","77u-Q1VTVE9NRVIgRk4gQ0VOVEVS75i2":"","77u-Q1VTV 5 5 7 8 - 12 3 - 16 7 - 20 1 - 24 5 - 28 9 - 32 3 - 36 7 - 40
_	Export

It shows Records, leafs and DPM nodes. Records are expected to be present in each DPart hierarchy on at least one level to define where records are differentiated. pdfToolbox shows right after the name (in this example "Record" the pages with which the respective record is associated.

DPM nodes contain the actual metadata which may in the PDF use any format (key - value pairs, XML, PDF syntax) but is in pdfToolbox converted into JSON.

Leafs hold the page associations which may be page ranges, but are in this example only single pages. The leaf entries contain links to the respective pages in the DPart viewer.





An Export button allows for saving the structure as a JSON file to the disk.



G Here you can get to know everything about DPartner and DPart metadata:



111



Use DPart metadata in a Process Plan via QuickCheck

You can extract DPart metadata from a PDF file into a JSON structure via QuickCheck, e.g. as the first step in a Process Plan and then use this structure in the following steps.

The attached Process Plan demonstrates how to do that.

Remove_pages_with_DPart_"Male_Cerebellum".kfpx

It is very specific to the DPart structure and can only be used with one of the "Cerebellum ..." sample files from the "Cal Poly Graphic Communications PDF/VT Test File Suite" that is available from the PDF Association: <u>https://www.pdfa.org/resource/cal-poly-pdfvt-test-suite/</u>

When you open the PDF in pdfToolbox you may first analyze the DPart structure (as described in <u>Display DPart metadata</u>).

	callas pdfToolbox: DPart Explorer
🔻 🗾 File nam	ne: Cerebellum V1.0.1 - 100.pdf
🔻 🗾 Root	
🔻 🕤 Re	ecord 1 - 13
▼ 🗖	DPM
	{"Record":1,"RecordValues":{"77u-R2VuZGVy75i2" Female Cerebellum", 77u-R2Vu
	Page 1
	Page 2
	Page 3
	Page 4
	Page 5
	Page 6
II - 12	Page 7
II	Page 8
II - 13	Page 9
II - 12	rage to Page 11
II - 53	Fage 12
II - 63	Page 12
	Fage 15
	DPM
	A "Record":2 "Record Values":/"77u-R2\/u7G\/u7G\/u75i2" Male Cerebellum" 77u-R2\/u7G
11 b S	Page 14
II - 63	Page 15
II - 63	Page 16
II - 15	Page 17
1 - FS	Page 18
1 × ×	Page 19
X	Page 20
→ ×	Page 21
	Page 22
▶ ∖	Page 23
▶ 	Page 24
> ×	Page 25
• ×	Page 26
🕨 💿 Re	ecord 27 - 39
🕨 💿 Re	ecord 40 - 52
🕨 💿 Re	ecord 53 - 65
🕨 💿 Re	ecord 66 - 78
P 💀 Re	ecord 79 - 91
Re Re	ecord 92 - 104
Re Re	BCORD 105 - 117
	BCORD 118 - 130
	scord 144 - 156
	acord 157 - 169
O Re	30010 107 - 108
	Export



When you open some DPM entries you will see that some have a value with "Male Cerebellum" while others have "Female Cerebellum".

In this example we use this metadata to decide what to do and in this example that is to remove all pages that are associated with "Male Cerebellum".

When you apply the Process Plan above to one of the "Cerebellum ..." files it will reduce the number of pages and after that operation many of the records will not have any page associations anymore, since the pages have been removed.



You can immediately identify records with pages and without pages since the DPart viewer displays page numbers after each record. If you open those entries and look into the DPM entries you will see that only Female Cerebellums are left.



JavaScript enhancements



Using object coordinates from a hit in a Process Plan

Beginning with pdfToolbox 11 coordinates for all objects identified by a hit are available in the pdfToolbox JavaScript object app.doc. In this example this information is used to print image resolutions on all images in a PDF file.



Mark_images_with_resolution.kfpx

You can use this Process Plan with any file including this one:



The Process Plan identifies all images and their image resolutions in step 1. In the second step the results are copied into app.vars.imageHits and in the final step the coordinates are used to place the image resolutions at the top and centered on each image.

An alternative approach is used in this Process Plan.

Mark_images_with_resolution_in_a_loop_via_a_r.kfpx

It prints the resolutions in a loop which is much slower than in the Process Plan above where a combined template is created and then put on top of the page. But since it lacks the complexity of the combined template it may be easier to adjust.

New properties in app.doc.result.checks[i].hits

- "llx": lower left x coordinate of snippet bounding box (pt)
- "Ily": lower left y coordinate of snippet bounding box (pt)
- "urx": upper right x coordinate of snippet bounding box (pt)
- "ury": upper right y coordinate of snippet bounding box (pt)



- "type": Type of snippets:
 - "Fill"
 - "Stroke"
 - "StrokeFill"
 - "TextFill"
 - "TextOutline"
 - "TextOutlineFill"
 - "TextInvisible"
 - "InlineImage"
 - "XObj"
 - "Image"
 - "FormXObj"
 - "PostScript"
 - "Shade"
 - "Unknown"
 - "InvalidCmd"



Map (spot and process) colors using script variables

Introduction

When you want to configure the Fixups "Map colors" and "Map spot and process colors" via JavaScript you face some problems, mainly that you can't add additional settings via JavaScript (in the UI you would use the [+] button) and that you have to predefine variable for all dynamic fields.

pdfToolbox 11 introduces two new Fixups: "Map colors using script variables" and "Map spot and process colors using script variables" that are configured with a single JSON object.



Usually you will assign a variable to it (via the orange triangle as usual) and create the value of the variable as a JSON structure according to the example in the UI.



Possible values in the JSON structure

Map spot and process colors using script variables

Source color name:	equal to	V	
Destination:	Map or rename		
Destination color name:		-	
Destination/Alternate color model:	Use source color		
Destination/Alternate color value:		\bigtriangledown	
Overprint:	Unchanged 👻 💌		
Apply to:	None 🔻 🔻 📃 🖉		
🔽 🔻 Ignore upper/lower case			
Keep alternate color definition of destination color if present in the PDF			

Name	Possible values	Corresponds to ("Map spot and process colors")
source		
operator	1 2 3 4	matches with RegEx does not match with RegEx equal to unequal to
name	<any proper="" string=""></any>	
ignorecase	true false	
keepalternatecolor	true false	
destination		
operation	ConvertToCMYK MapOrRename KeepName ConvertToDestination	Convert to CMYK Map or rename Change alternate color Convert to destination
model	Automatic	Use source color



Name	Possible values	Corresponds to ("Map spot and process colors")
	CMYKPercent CMYKZeroToOne GrayPercent GrayZeroToOne Lab RGBPercent RGBZeroTo255 RGBZeroToOne	CMYK (%) CMYK (0.01.0) Gray (%) Gray (0.01.0) 0.0 is black Lab (0100,-128127,-128127) RGB (%) RGB (0255) RGB (0.01.0)
name	<any proper="" string=""></any>	
value	array of numbers, length ac- cording to color space	
overprint	On Off Unchanged	
applyto	None Images VectorAndText	None All images All vector and text objects

Map colors using script variables

Source color model:	СМҮК (%)		▼						
Source color value:	0	▼	0	▼	0	▼	0	▼	
Tolerance:	0							▼	
Include intermediate color values									
Destination/Alternate color model:	Use source color		-						
Destination/Alternate color value:	0	\bigtriangledown	0	\bigtriangledown	0	\bigtriangledown	0	∇	
Create as spot color									
Spot color name:								\bigtriangledown	
Apply to:	None		-	+					
Spot color tint value:	100							% 🛡	
Overprint:	Unchanged 🔻 🔻								

C callas

Name	Possible values	Corresponds to ("Map colors")
source		
model	CMYKPercent CMYKZeroToOne GrayPercent GrayZeroToOne RGBPercent RGBZeroTo255 RGBZeroToOne	CMYK (%) CMYK (0.01.0) Gray (%) Gray (0.01.0) 0.0 is black RGB (%) RGB (0255) RGB (0.01.0)
value	array of numbers, length accord- ing to color space	
tolerance	number	
intermediate	true false	
destination		
model	Automatic CMYKPercent CMYKZeroToOne GrayPercent GrayZeroToOne RGBPercent RGBZeroTo255 RGBZeroToOne	Use source color CMYK (%) CMYK (0.01.0) Gray (%) Gray (0.01.0) 0.0 is black RGB (%) RGB (0255) RGB (0.01.0)
value	array of numbers, length accord- ing to color space	
applyto	None Images VectorAndText	None All images All vector and text objects
spotcolor		
create	true false	
name	<any proper="" string=""></any>	
tintvalue	number	



Name	Possible values	Corresponds to ("Map colors")
overprint	On Off Unchanged	
keepalternatecolor	true false	

Example: Convert colors using wildcards

To give an example we have created a Fixup to convert colors which allows for using a "wildcard" in one colorant. That means you could convert e.g. all colors using C50 M50 and K50 to something else, keeping all Y values as they are.

P

Map_colors_with_a_wildcard.kfpx

This PDF has a number of color patches, all using C75, M50, K0 and varying values for Y.



Text_Script_Mapping_original.pdf





If you apply the Fixup and enter C75, M50, K0 for the input values of the respective colorants and a "?" (wildcard) for Y you can convert C,M and K to whatever values you enter (the output value for Y does not matter, the original values will be kept).

In this example we have used C0, M95, K10 for output.

New in pdfToolbox 11







One more thing:

You may as well use more than one wildcard ("?"). The problem is that it will then process much longer, in one test that we made it took roughly 30 minutes. So you should only do so if you have time...



Convert colors



Convert RGB to CMYK using custom tolerance for gray

In RGB gray is indicated by same (or similar) color values for all three colorants. The tolerance is important when you convert RGB to CMYK, at least when you want to keep gray so that RGB gray would only use the K colorant (or DeviceGray or Separation Gray depending on the other settings in the Convert colors Fixup). This is controlled via the "Preserve black objects" checkbox in Convert colors.

The internal tolerance in Convert colors for that is 2%. But pdfToolbox 11 introduces a new Process Plan "Convert RGB to CMYK (PSO Coated v3) with specified R=G=B tolerance" which allows you to modify this tolerance.

	•	callas pdfToolbox	K	
◄		Profiles	sparency 1 🔻	
De	fault	· 208	R=	8
∇	Convert colors			
	Sonvert RGB to CMYK	(PSO Coated v3) with specifie	d R=G=B tolerance	Edit
	Converts RGB objects objects that are converted objects that are converted objects that are converted objects and the second objects are converted objects and the second objects are converted objects and the second objects are converted obje	to CMYK (PSO Coated v3) with rted to CMYK gray. Spot colors	a specified tolerance for are maintained.	or R=G=B, i.e.
- F	Further Options		2ª Analyze	🚼 Analyze and fix
1 -				
Ŀ	Preflight only pages 1	to 1		
		Licensed to Dietrich (o	allas)	

Via a variable you can specify the tolerance.





How it works

The first step uses the Convert colors Fixup with a filter Check. The filter uses the "Difference between colorant channels" property with a variable that defines the maximum difference between the R,G and B colorant to be matched by the filter. All such colors are converted to gray (Dot Gain 15%).

In the second step the remaining colors are converted to CMYK without preserving black objects. that means the tolerance can be either above or below 2% since what colors are considered to be gray is determined in the first step.

Example

This test chart has a R=G=B=50% patch in the center (indicated with a black stroke) and color shifts of 2.5% towards the outer edges. This can be used to visualize the effects of the Process Plan.



RGB_Testchart_box2pt.pdf





When you apply the Process Plan with a tolerance value of 25% the result looks like this.



127





Watch Dietrich von Seggern talk about RGB to Gray conversion in the video below:



Convert colors: Advanced settings (previously "Policies")

The regular ICC based color conversion is extended by some useful options. This includes obtainment of pure black, recognition of RGB black or adaption of tone value increase. Additional parameters for controlling these properties can be defined in the "Advanced settings" tab, that can be selected in the "Convert colors" Fixup.

You will find details on parameters and example configurations on the following pages.

Old, deprecated configuration using Policy files

Before pdfToolbox 11, these options were defined by config files, which were completely text based and could be modified with a simple text editor. Exported color conversion profiles were containing these additional files. You'll find the documentation of this deprecated implementation here: <u>Policies</u>



Overview

	callas pdfToolbox: Edit Fixup		
Name:	Fixup category: Type	of fixup: Find	
Convert color to PSO Coated v3 (ECI) (keep spot colors) Comment: Converts the current PDF document to CMYK, using PSO Coated v3 (ECI) as destination profile. All spot colors are maintained.	All Color spaces, spot colors, inks Pages Document info and Metadata Interactive elements and properties Document Page contents Layers	Adjust colors for PDF based ISO standards Adjust dot gain Condense into single spot color Downer colors Downer color Convert [®] fixup offers very powerful and leable options for converting colors in many ways, allowing for highly customizable color conversion ettings. Downert colors using Devicel ink profiles.	
•	Convert colors:	gs Afranced settings 4. SeparationAsDeviceCMYK • • • • 4. SeparationAsDeviceCMYK • • • • 4. SeparationAsDeviceCMYK • • • • • • • • • • • • • • • • • • •	
Custom ID (for Reports):	Value: Apply dot gain difference 👻 💌	- +	
Apply this fixup to:	Parameter: Destination-SetTransparency8lendSpaceTo Value: Set to destination as device color space	Dest • •	
Embedded files Hide in list of Fixups This fixup is not locked.	S	Switch to list view Switch to standard view	
Usage		Test Cancel OK	

All parameters defined in the "Advanced settings"-tab take precedence over all settings defined in the user interface. Please note that nonetheless only objects triggered due to the "Conversion settings" in the user interface will be changed.

The following parameters are available (please consult section <u>Parameter</u> for a detailed explanation):

- <u>RenderingIntent</u>
- <u>C eq M eq Y is Black</u>
- <u>C_eq_M_eq_Y_is_Black_ExcludeBlendModes</u>
- <u>R_eq_G_eq_B_is_Black</u>
- <u>R_eq_G_eq_B_is_Black_ExcludeBlendModes</u>
- <u>SetGrayColorSpaceTo</u>
- <u>a eq b eq 0 is Black</u>
- <u>a_eq_b_eq_0_is_Black_ExcludeBlendModes</u>
- <u>HandleProcessColorInDeviceN_SeparationAsDeviceCMYK</u>
- <u>AdvancedColorConversion</u>
- <u>CompressionMethod</u>
- JPEGQuality



<u>SetTransparencyBlendSpaceToDest</u>

Parameters

RenderingIntent

Keys

- CMYKGraph-RenderingIntent
- CMYKImage-RenderingIntent
- RGBGraph-RenderingIntent
- RGBImage-RenderingIntent
- GrayGraph-RenderingIntent
- GrayImage-RenderingIntent
- LabGraph-RenderingIntent
- LabImage-RenderingIntent

Values

Use Rendering Intent of document	Rendering Intent spec- ified in the PDF is used (default)
RelativeColorimetic	Relative colorimetric
AbsoluteColorimetric	Absolute colorimetric
Perceptual	Perceptual
Saturation	Saturation

Description

This parameter defines which Rendering Intent is used for a color conversion. This parameter overrides the option Rendering Intent of the conversion settings.

• Note: If there is no Rendering Intent explicitly defined in the PDF the default is "Relative Colorimetric".



C_eq_M_eq_Y_is_Black

Keys

- CMYKGraph-C_eq_M_eq_Y_is_Black
- CMYKImage-C_eq_M_eq_Y_is_Black

Values

lgnore para- meter	Convert as defined for color space CMYK (de- fault)
DeviceGray	Use De- viceGray
DeviceCMYK Black	Use De- viceCMYK Black
Separation "Black"	Use Separa- tion Black

If only the black channel is used, the object will be converted according to the settings for color space Gray and stored in the defined color space afterwards. This parameter overrides the option Preserve black objects of the destination settings for CMYK.

C_eq_M_eq_Y_isBlack_ExcludeBlendModes

Keys

- CMYKGraph-C_eq_M_eq_Y_is_Black_ExcludeBlendModes
- CMYKImage-C_eq_M_eq_Y_is_Black_ExcludeBlendModes

Values



Description

Objects, which are using a Blend mode listed in this parameter, will be processed using the normal color conversion and no preservation of black objects will take place.

This might be appropriate for objects (with empty colorant channels), that are using certain blend mode, which are using these empty channels to achieve a special effect with underlying objects.

If these empty channels would be replaced by a single colorant color space, these effects would not work anymore, so they must be preserved in some certain cases.

Please see article <u>"Processing black objects"</u> for further details.

R_eq_G_eq_B_is_Black

Keys

- RGBGraph-R_eq_G_eq_B_is_Black
- RGBImage-R_eq_G_eq_B_is_Black

Values

Ignore para- meter	Convert as defined for color space RGB (de- fault)
DeviceGray	Use De- viceGray
DeviceCMYK Black	Use De- viceCMYK Black



Senaration	lise Senara-
"Black"	tion Black
Diack	tion black

Description

This parameter only applies to device dependent RGB color – either if the object uses DeviceRGB or if it uses ICC based RGB or CalRGB but the parameter StripSourceProfile is set to On. It only has effect if ConvertToDestination is set to On. If not set to NoChange, instead of carrying out an ICC based color conversion, RGB colors where each of the Red Green and Blue values are equal, will be converted directly to the corresponding black tint value. For images this will only have an effect if all pixels in the image have equal Red, Green and Blue values.

The color conversion engine uses an internal tolerance of 3%, this mean the differences between R and G, G and B as well as for R and B must be <=3%.

R_eq_G_eq_B_is_Black_ExcludeBlendModes

Keys

- RGBImage-R_eq_G_eq_B_is_Black_ExcludeBlendModes
- RGBGraph-R_eq_G_eq_B_is_Black_ExcludeBlendModes

Values

Blend mode	Name of Blend modes to exclude from color conversion To exclude multiple Blend modes, one setting for each blend mode has to be configured
------------	---

Description

Objects, which are using a Blend mode listed in this parameter, will be processed using the normal color conversion and no preservation of black objects will take place. This might be appropriate for objects (with empty colorant channels), that are using certain blend mode, which are using these empty channels to achieve a special effect with un-



derlying objects.

If these empty channels would be replaced by a single colorant color space, these effects would not work anymore, so they must be preserved in some certain cases.

Please see article <u>"Processing black objects"</u> for further details.

SetGrayColorSpaceTo

Keys

- GrayGraph-SetGrayColorSpaceTo
- GrayImage-SetGrayColorSpaceTo

Values

Keep color space for Gray	Convert as defined for color space Gray (de- fault)
DeviceGray	Use De- viceGray
DeviceCMYK Black	Use De- viceCMYK Black
Separation "Black"	Use Separa- tion Black

Description

This parameter applies to gray vector and image objects. It indicates which color space to use for gray objects after the conversion. For example, it may be necessary to encode a DeviceGray object as Separation Black object, so that overprinting also works for subjacent CMYK objects.

• Note: DeviceGray cannot overprint CMYK, even if overprint is set to true, whereas Separation Black does over-



136

print Cyan, Magenta and Yellow once overprint is set to true.

• Note that both DeviceGray and Separation Black do overprint spot colors if overprint is set to true.

a_eq_b_eq_0_is_Black

Keys

- LabGraph-a_eq_b_eq_0_is_Black
- LabImage-a_eq_b_eq_0_is_Black

Values

lgnore para- meter	Convert as defined for color space Lab (de- fault)
DeviceGray	Use De- viceGray
DeviceCMYK Black	Use De- viceCMYK Black
Separation "Black"	Use Separa- tion Black

Description

If a and b is 0 the luminosity value is directly transferred to the designated color space.

The color conversion engine uses an internal tolerance of 1%, this means the differences between L and a as well as for L and b must be <= 1%.

a_eq_b_eq_0_is_Black_ExcludeBlendModes

Keys

LabGraph-a_eq_b_eq_0_is_Black_LimitToBlendModes



LabImage-a_eq_b_eq_0_is_Black_LimitToBlendModes

Values

Blend mode	Name of Blend modes to exclude from color conversion To exclude multiple Blend modes, one setting for each blend mode has to be configured
------------	---

Description

Objects, which are using a Blend mode listed in this parameter, will be processed using the normal color conversion and no preservation of black objects will take place.

This might be appropriate for objects (with empty colorant channels), that are using certain blend mode, which are using these empty channels to achieve a special effect with underlying objects.

If these empty channels would be replaced by a single colorant color space, these effects would not work anymore, so they must be preserved in some certain cases.

Please see article <u>"Processing black objects"</u> for further details.

HandleProcessColorInDeviceN_SeparationAsDeviceCMYK

Keys

- CMYKGraph-HandleProcessColorInDeviceN_SeparationAsDeviceCMYK
- CMYKImage-HandleProcessColorInDeviceN_SeparationAsDeviceCMYK

Values

ors defined in DeviceN or Separa-	col- ed PN a- pot
-----------------------------------	-------------------------------



tion as spot	colors (de-
colors	fault)
Treat	Treat
process col-	process col-
ors defined	ors defined
in DeviceN	in DeviceN
or Separa-	or Separa-
tion as De-	tion as De-
viceCMYK	viceCMYK

Description

This parameter defines the way process colors (Cyan, Magenta, Yellow, Black) in DeviceN or Separation are treated. If set to On, they are treated as DeviceCMYK objects. Not defined color channels are set to 0% in the respective color. To keep the overprint settings, the original color space (Separation or DeviceN) is recreated after the conversion if possible.

Use the AdvancedColorConversion parameters to define the treatment of newly added color channels.

AdvancedColorConversion

Keys

- CMYKGraph-AdvancedColorConversion
- CMYKImage-AdvancedColorConversion
- GrayGraph-AdvancedColorConversion
- GrayImage-AdvancedColorConversion

Values

No spe- cial treat- ment	Convert as defined for destination color space
Apply dot	Apply dot
gain dif-	gain differ-
ference	ence



C callas

Description

If set to "No special treatment", tints in colorants that were originally 0 are preserved.

Handles all black objects. Black objects are using only the K channel of CMYK or DeviceGray or Separation Black or DeviceN with only one channel named Black.

If necessary, new channels will be added for DeviceN (Separation color space becomes DeviceN after the conversion). With AdjustDotGain the dot gain difference between source and destination profile is calcuated and all color values are modified so that the dot gain difference is compensated. This is useful mainly for standard PCS conversions.

CompressionMethod

Keys

Destination-CompressionMethod

Values

Keep compression method	Keep compression method of original (default)
Compress all to ZIP	Compress all handled images with ZIP
Compress all to JPEG	Compress all handled images using JPEG

Description

Specifies the method for recompression of converted images.

 Note: All processed images are decompressed for color conversion tasks. If recompressing with JPEG the quality level is to be defined with the parameter <u>Destina-</u> <u>tion-JPEGQuality</u>. Please keep in mind that every JPEG compression leads to the loss of quality.



JPEGQuality

Keys

• Destination-JPEGQuality

Values

Minimum	Minimum JPEG quality compression (20)
Low	Low JPEG quality compression (40)
Medium	Medium JPEG quality compression (60)
High	High JPEG quality compression (80)
Maximum	Maximum JPEG quality compression (100)

Description

Allows for specifying the quality level in which JPEG objects are saved after the conversion.

• Note: The predefined values are interpreted as a percentage declaration between 0 and 100. The value in brackets are giving a rough comparision to the equivalent setting in Adobe Photoshop.

SetTransparencyBlendSpaceToDest

Keys

Destination-SetTransparencyBlendSpaceToDest

Values

Leave unchanged	Leaves transparency blend color space unchanged



Set to destination ICC profile	Sets destination color space as transparency blend color space
Set to destination as device color space	Sets destination color space as transparency blend color space; if destination color space is ICC based, the respec- tive device color space is set as transparency blend color space
Set to destination if equal to Output Intent	Sets destination color space as transparency blend color space; if destination color space is ICC based and the out- put intent matches the destination profile, the re- spective device color space is set as transparency blend color space

Description

Allows for specifying the transparency blend color space after the color conversion.

Implementation notes

When importing old "Convert colors"-Fixups, the contained Policy file will show up in the "Advanced conversion policy" Pop-Up in the "Destination"-tab.





It is possible to edit the Policy file using the parameters described on <u>this page</u>. You can even select another existing Policy file and export such Fixups

To avoid contradictionary settings, no configuration can be done in the "Advanced settings"-tab in such Fixups.

No matter if the old, deprecated Policy files or the new "Advanced settings" are used in the Fixup, both will work in pdfToolbox 11 and later.

Watch Dietrich von Seggern talk about 'Convert colors: Advanced settings' in the video below:



Processing black objects with Advanced settings

General

Black and tinted black (Gray) objects in PDF files can be rendered in different ways.

Common color spaces

DeviceGray

Gray color space with 1 channel 0% represents black in this color space

DeviceCMYK

CMYK color space with 4 channels C, M and Y are 0 %, where only the black (K) channel is used

Separation Black

Color space with 1 channel

Uses only the K channel of the process colors for output

DeviceN Black

Color space with 1 channel

Uses only the K channel of the process colors for output

Other color spaces

There also is a special form of black in the following color spaces:



DeviceRGB

If all color components have the same value, the visual representation is tinted black.

Lab

If the color values for a and b equal 0, the visual representation is tinted black.

ICC based color spaces

Additionally there are the following color spaces:

ICCbasedGray, ICCbasedCMYK, ICCbasedRGB

When processing black it is recommended to ignore ICC profiles, since this would lead to a conversion of black using the PCS of the ICC profile resulting in rich black defined in CMYK (assuming the destination profile to be a CMYK profile).

Default handling

Depending on the color space the black is defined in, it will normally be processed like all the other colors in this color space; e.g. a black in DeviceCMYK will be processed like the other CMYK values.

Within the scope of print production though it is preferable to treat every black the same, no matter how it is defined internally. To process black elements properly according to the settings in the gray conversion, the following parameters are available:

- Gray...._SetGrayColorSpace
- RGB....._R_eq_G_eq_B_is_Black
- Lab....._a_eq_b_eq_0_is_Black
- CMYK....._C_eq_M_eq_Y_is_Black

Note:

Please keep in mind that some of the parameters mentioned in this article have a prefix (e.g. "RGBImage-..." or "CMYK-Graph-...") which will be shown in the "Advanced settings"-


tab of the "Convert colors"-Fixup. Details can be found in the article: <u>Advanced settings for Con-</u> <u>vert colors</u>

When is black considered as black?

The color conversion engine uses a small tolerance when detecting black objects. So the objects considered as "black" might have a slightly different colorant value, but as their visual appearance will be gray when printed, they'll be handled the same way as if there colorant values were completely the same (also called "gray balance").

The used tolerances for the different colorspaces are:

Lab: Tolerance = 1% ((|L - a| <= 0.01) & (|L - b| <= 0.01)) = Gray

RGB and calRGB: Tolerance = 3% ((|R - G| <= 0.03) & (|G - B| <= 0.03) & (|R - B| <= 0.03)) = Gray

CMYK: Tolerance = 0% (C == 0) & (M == 0) & (Y == 0) = Gray

Setting up

Behavior of "Black" defined in common color spaces during color conversion

In the usual cases it is desirable, that black/gray elements are only represented in the K channel after any conversion. Therefore you may use the option "Preserve Black" in the Fixup "Convert colors". This guarantees that all black objects are converted according to the gray treatment as required.

The parameters set by the option "Preserve Black" are:

- GrayGraph_SetGrayColorSpaceTo Separation "Black"
- GrayImage_SetGrayColorSpaceTo Separation "Black"
- CMYKGraph_C_eq_M_eq_Y_is_Black Separation "Black"

C callas

 CMYKImage_C_eq_M_eq_Y_is_Black Separation "Black"

The parameters set in most predefined Convert colors Fixups in pdfToolbox (overruling the parameters set by the option "Preserve Black") are:

- GrayGraph_SetGrayColorSpace Keep color space for Gray
- GrayImage_SetGrayColorSpace Keep color space for Gray
- Gray_AdvancedColorConversion Apply dot gain difference

Elements defined in DeviceGray remain DeviceGray after conversion. Gray elements defined in DeviceCMYK/Separation Black or DeviceN Black are mapped to Separation Black after the conversion.

NOTE:

It is recommended to define the 3 parameters also when setting up a new, own "Convert colors"-Fixup with the activated "Preserve Black" option.

Conversion of RGB/Lab Black/Gray to CMYK

The parameters set by the option "Preserve Black" are:

- RGBGraph_R_eq_G_eq_B_is_Black Separation "Black"
- RGBImage_R_eq_G_eq_B_is_Black Separation "Black"
- LabGraph_a_eq_b_eq_0_is_Black Separation "Black"
- LabImage_a_eq_b_eq_0_is_Black Separation "Black"

The option "Preserve Black" converts any RGB and/or Lab black to Separation Black, if objects of this color space are converted to CMYK. This conversion applies for images as well as for text and vector objects. During this conversion RGB gray scales are mapped 1:1 to Separation Black. This means, a 50% RGB gray (127/127/127) will be 50% Separation Black after conversion.

Tone value adjustments for conversions amongst different printing conditions

If files are converted amongst different printing conditions the different dot gain might be counterbalanced. The follow-



ing parameters (which should be used in most cases and are contained in almost all predefined "Convert colors"-Fixups) allow to compensate the different dot gain:

- GrayGraph-AdvancedColorConversion Apply dot gain difference
- GrayImage-AdvancedColorConversion Apply dot gain difference

The difference in dot gain between the gray source ICC profile and the destination ICC profile will be compensated. 100% Black remains unchanged.

Changing the destination color space of Gray

By changing the parameters in the "Advanced settings"-tab, it is possible to define the destination color space of gray elements with the following parameters:

- Gray....._SetGrayColorSpace
- RGB....._R_eq_G_eq_B_is_Black
- Lab....._a_eq_b_eq_0_is_Black
- CMYK....._C_eq_M_eq_Y_is_Black

Ignore parameter	Element will be converted according to the settings for the corresponding color space
DeviceGray	Element will be converted according to the settings for the color space "Gray" and stored in DeviceGray
DeviceCMYK Black	Element will be converted according to the settings for the color space "Gray" and stored in DeviceCMYK – Black channel only
Separation "Black"	Element will be converted according to the settings for the color space "Gray" and stored in Separation Black
Keep color space for Gray	Convert as defined for color space Gray (default) (only available for "GraySetGrayColorSpaceTo")

 Attention: In DeviceGray, a defined Overprinting flag will have no influence on subjacent CMYK elements. These will always be knocked out.

Preserve black: Exclude certain BlendModes from special



handling (pdfToolbox 9.2)

Some special transparency effects are based on the interaction of color channels and therefore, a conversion of objects using a gray balance of a process color (e.g. RGB, CMYK or Lab) to pure gray scale may result in visually changed results. As these effects differ from the "construction", the used BlendMode and color space of a PDF a general solution is not possible and excluding certain BlendModes depending on the input files is necessary.

- RGBImage-R_eq_G_eq_B_is_Black_ExcludeBlendModes
- RGBGraph-R_eq_G_eq_B_is_Black_ExcludeBlendModes
- CMYKGraph-C_eq_M_eq_Y_is_Black_ExcludeBlendModes
- CMYKImage-C_eq_M_eq_Y_is_Black_ExcludeBlendModes
- LabGraph-a_eq_b_eq_0_is_Black_ExcludeBlendModes
- LabImage-a_eq_b_eq_0_is_Black_ExcludeBlendModes

All BlendModes can be excluded, which means they are converted to a process color space. Normally the gray balance is preserved by converting it to grayscale, but this can result in a changed visual appearance with some special usage of certain BlendModes (e.g. ColorDodge).

Possible values are: Normal, Compatible, Multiply, Screen, Overlay, Darken, Lighten, ColorDodge, ColorBurn, HardLight, SoftLight, Difference, Exclusion, Hue, Saturation, Color or Luminosity.

If more than one BlendMode shall be excluded, the values have to be separated either by a space, a comma or a semicolon.



148

C callas

New Server functionalities



Using Job Tickets

Usually, a Job of a Server needs incoming files in the specified "In" folder to start processing as defined in the respective Job settings.

The "Job Ticket mode" makes it possible to put just a ".jobticket"-file into the input hotfolder of the Server (instead of a PDF), which contains all necessary information about the file to be processed as well as optionally additional parameters like e.g. a Profile, additional CLI parameters or variables and their values.



When setting up a Job, just select "Job Ticket mode" instead of a Profil.

Using this mode, the Server will only process files with the extension ".jobticket" in the input folder of this Job. All other files will remain in the input folder and will not be processed.

The ".jobticket" file has to be formatted in JSON, containing all needed information for processing a file.

150

Job Ticket for Profile-based processing

As already mentioned, instead of placing an input file, which will be processed based on the fixed settings of a Job, the way a file is processed in a "Job Ticket"-Job is completely flexible - and there is no need to place the input file into the input folder as well.

All required components (like the input file, the Profile and maybe ressources referenced by Variables) can remain in their original location and have to be just referenced in the Job Ticket.

Each Job Ticket has to start with the "type", which must be "jobticket".

Within the "params" section, almost all CLI parameters of pdfToolbox can be used.

Minimum content:

- As the Job itself does not define a Profile, a valid path to a Profile must be defined.
- Path to the file to be processed

All other parameters are optional.

Reports have to be defined in the .jobticket file, as they can not become set in the Job.

By default, the report will be saved into the folder of the respective severity of each process (Error, Warning, Info or Success).

However, the path to a report can also be defined in the .jobticket-file.





Job Ticket for Actions

When an Action (e.g. Save as image, ReDistill, Impose) shall be performed, the syntax of the .jobticket-file is quite similar. Instead of defining a path to a Profile, the respective Action command and the needed options have to be defined:

```
{
         "type": "jobticket",
        "params":
        [
                 "--saveasimg",
                 "E:\\Job2\\in\\Testfile.pdf",
                 "--imgformat=jpeg",
                 "--colorspace=CMYK",
                 "--resolution=150"
        ]
}
{
        "type": "jobticket",
        "params":
        Γ
                 "--mergepdf",
                 "E:/Job_4711/in/file_001.pdf",
                 "E:/Job_4711/in/file_002.pdf",
                 "E:/Job_4711/in/file_003.pdf",
                 "E:/Job_4711/in/file_004.pdf"
        ]
}
```

How to test a Job Tickets

To test the behavior of a .jobticket-file, they can be used with pdfToolbox on the command line:

pdfToolbox --jobticket <input.jobticket>



Syntax issues to be considered

Be aware of JSON-related typological issue and other implementation-based issues:

- Don't use (") but use (")
- Escape (\) on Windows → (\\) or use a normal slash (/) instead
- Escape Quotes (") \rightarrow (\") when used as string
- Spaces () must not be escaped

Limitations

The following output options are not supported:

- Outputfile
- Outputfolder
- Overwrite

Other currently not supported options:

- Cachefolder
- All kinds of distributed processing options
- Timeout settings

Everything from 'How to' to 'Do's and Dont's' of Job tickets in this video:



Sidecar files and their use when processing files with a Server-Job

With sidecar files, it is possible to add a file to the input hotfolder (that has the input PDF) which contains additional information in the form of Variables. This information can be used and modified during the hotfolder processing. The new (updated) values can be used to influence processing in later hotfolder steps.

Using Sidecar files

- When sidecar files shall be used, no new entry in the configuration of a Job is necessary - if a *.sidecar-file exists beside a PDF, their content is just used for processing.
- Values of the "variables" in the .sidecar-file are used as Variables for every Job.
- To use new values (e.g. calculated or set in previous hotfolder jobs) for Variables, they have to become grabbed up from the results area and it must be handled in the JavaScript Variable in that way, that their value overwrite the existing Variables during runtime.
- A .sidecar-file can not contain additional CLI commands or options.
- A .sidecar-file should exist at least simultaneously to the input file inside the input folder. It is recommended to copy the .sidecar-file before the input file to ensure the Server is aware of the sidecar file when scanning the input folder.

Syntax of Sidecar files

As already mentioned, optionally a .sidecar-file can be copied into the In folder beside the input file. The content of the .sidecar-file will be taken into account for processing. But the general parameters from the Job settings will control the way a file is handled.

The ".sidecar" file has to be formatted in JSON, containing all additional information for processing a file.





The .sidecar-file must have the exact, case-sensitive same name as the original file, e.g.:

- Inputfile.pdf
- Inputfile.pdf.sidecar

```
{
   "type":"sidecar",
   "variables":
   [
      {
        "key":"placetext",
        "value":"Some text from the initial sidecar file"
      },
      {
        "value":"Some text from the initial sidecar file"
      },
      {
        "key":"activate_fixup",
        "1"
      }
  ]
}
```

Each Job Ticket has to start with the "type", which must be "sidecar".

Within the "variables" section new values for Variables can be defined, which will then be used for the specific file.

Updated content after a Job

The normal "variables" part will not be updated, if a Variable has changed during processing.

However, all changed Variable as well as details to every performed processing step is added to the .sidecar-file as shown in the sample below:



"variables" : { "placetext" : "Some text from the initial sidecar file", "placetext_new" : "Some text from the initial sidecar file and some additional text", "script" : "Some text from the initial sidecar file and some additional text" } }], "type" : "sidecar", "variables" : Γ { "key" : "placetext", "value" : "Some text from the initial sidecar file" }, { "key":"activate_fixup", "1" } 1 }

In this sample, a new Variable "placetext_new" is created by a JavaScript-Variable in the Profile. It uses the supplied, new "placetext"-Variable and just adds some additional text.

Create Sidecar files from a Server-Job

It is possible to create a .sidecar-file if not existing by activating the respective option within the four settings for each severity. Creating such a .sidecar-file enables a pick-up of results by following steps.



		callas pdfToolb	oox: Job	
٢		callas pdfToolb	ox Server	
Name:	Jobticket Job			
Description:				
Profile:	Job Ticket mo	de		1 · · · · ·
Folders:				
Overview In folder	1	Folders:	/Hotfolder/Job_1/Success	s Choose
On succe	ess	Report settings:		•
On info On warning		Use checkpoint:	Send to checkpoint	
On error Processed files		Checkpoint name:		
		🗹 Create sidecar f	ile if not existing	
Options:				
Proces	s subfolder			
Overw	rite existing files			
V Log co	mplete CLI outp	out		
Keep c	original			
Additional CL	I parameters:			
			anal.	
		OK Ca	ancei	
	-	_	_	

Download of the used sample Profile

Place_text_and_add_some_text_to_a_new_Variabl.kfpx

Everything from 'How to' to 'Do's and Dont's' of Sidecar files in this video:



Ask-at-runtime dialog



Introduction

When simple variables are used in pdfToolbox Desktop (in a check, fixup, profile, process plan), pdfToolbox Desktop brings up the ask-at-runtime dialog window before running the check, fixup... In previous versions of pdfToolbox, it looked like this:

Enter now values or loove unch	anged to use the default values
Enter new values of leave unch	anged to use the default values.
Height of the trimbox (mm):	300
Width of the trimbox (in mm):	: 500
Inspect variables	
	Cancel

While most of the dialog window was fixed, the inner section always grew to display a list of all variables that are used. The user could now modify the values for all variables (initially shown are the default values for each variable) and then click "OK" to run the check, fixup...

Limitations

The ask-at-runtime dialog came with a number of serious restrictions:

• The dialog always looked the same and could not be customised in any way



- The order of the variables in the dialog window could not easily be modified
- Any kind of additional grouping or structure, or any modification to the type of input fields used was impossible
- Only the most basic validation was available

New in pdfToolbox 11

As of pdfToolbox 11, pdfToolbox Desktop switched to a different ask-at-runtime dialog window as shown below.

callas por looibo	bx: Specify variable value
Enter new values or use the default value	leave unchanged to es.
Height of the trimbox (mm):	
300	÷
Width of the trimbox (in mm):	
500	÷
	Cancel OK

While the overall functionality of the dialog window is the same, it uses an HTML template. This makes it so that the default window looks much better than it did in pdfToolbox 10, but it also allows modifications much more easily.

160



Working with ask-at-runtime templates

The default template for the ask-at-runtime dialog window is embedded in the pdfToolbox Desktop executable and shouldn't be modified. While this article doesn't cover editing templates, it does describe how to select them, create a new one to start modifications from or find the templates to edit them.

Selecting a template to use

Checks or fixups always use the default template. When creating a profile, it is possible to select the profile that will be used. Look for the Dialog property under the general profile properties (Labeled (1) in the window shown below).

Imported profiles > Ø Finibuox is not 5000 by 300 mm Document Pages Images Colors Standards compliance Standards compliance Standards compliance Convert to standard Convert to standard for embedded files Convert to standard for embedded files Convert to standard for embedded files Standards compliance Standards complia
Purpose: Discussion is not body by 300 mm Discussion Pages Images Colors Fonts Standards compliance Standards compliance for embedded files Convert to standard for embedded files Dialog: Mitch and height Hide in list of Profiles
Autor: Standards compliance Convert to standard for embedded files Convert to standard for embedded files Cotom ficuss Summary Summary Custom ficuss Liss Summary Custom ficuss Liss Summary Custom ficus Custom ficus
Convert to standard for embedded files Convert to standard for embedded files Custom fitups Summary Script Image: Imported profiles Custom fitups Custom fitups Cus
Custom checks Custom checks Custom fixups Summary Summary United and height Dialog: Width and height Dialog: Width and height
Cusulin indpos Script:
Usage Test Cancel OK

C callas

Available templates

Opening the "Dialog" pull-down menu reveals the following choices:

Variables:
Script: _
Dial Width and height - Default
Add due date Default_3 Width and height
Open folder with configuration files New

- 1. Select "Default" to use the default pdfToolbox Desktop profile for this profile.
- 2. This section displays the currently available templates and lets you choose any of these templates for this profile.
- 3. "Open folder with configuration files" opens the folder where the current templates (see (2)) are stored. Use this to edit a template or to create a copy.
- "New..." creates a new template. The template will be a copy of the default pdfToolbox Desktop template. You will be able to find it in the folder with configuration files (see option (3)).

Debugging a template

It is possible to connect the Chrome debugger to a running instance of the ask-at-runtime dialog window. In order to do so, you need to install an additional JSON file in the pdfToolbox Desktop preferences folder first. Locate the "Settings" folder in the pdfToolbox preferences folder.



1 The preferences folder can be found here:

- On Mac: /Users/<USERNAME>/Library/Preferences/ callas software/callas pdfToolbox <VERSION>
- On Windows: C:\Users\<USERNAME>\AppData\ Roaming\callas software\callas pdfToolbox <VER-SION>

In this settings folder, copy the attached JSON file:



Make sure the file is called "extras.json" and is placed inside the "Settings" folder. After installing it, restart pdfToolbox Desktop if it was running. Then open an ask-at-runtime dialog window and with it open, switch to Google Chrome. In Chrome, surf to:

http://localhost:8080

The port number can be modified in the "extras.json" file if desired.



1

C localhost:34568/AAR/TMP/755657ff-0e4c-4d23-a01a-c1a8da68f60f/index.html	🕞 🛅 Elements Console »
	html <html></html>
Enter new values or leave unchanged to use the default values.	<pre>> cheads_c/heads w vbody> == id > vdiv class="header">_v/div> > vdiv class="scril_container">_v/div> <script src<br="" type="text/jwascript">"scripts/class_arms.jf">>cf"></pre></td></tr><tr><td>Height of the trimbox (mm):</td><td><pre><script type="text/javascript" src= "scripts/popper.min.jg"></script> cscripts/popper.min.jg"> cscript type="text/javascript" src= </pre>
300	" <u>scripts/tippy_all_min.js</u> "> <script src="</td" type="text/javascript"></tr><tr><td>Width of the trimbox (in mm):</td><td>"<u>scripts/mustache.js</u>"></script> <script src="<br" type="text/javascript">"script"(internal.is"></script>
500	<pre><script src="<br" type="text/javascript">"script/templates.js"></script> <script src="<br" type="text/javascript">"scripts/helper.js"></script> <!--==</pre--></pre>
	html body
	Styles Computed Event Listeners >>
	Filter :hov .cls +
	element.style {
	<pre>bdg { <u>style.css:21</u> font-family: Arial, Helvetica, sans- serif; margin: 0; padding: 0; display: flex; flex-direction: column; flex-direction: column; background-color: im#e8e8e6; height: 100vh; } *, otbefore, *:after { <u>style.css:12</u> backstrond-color=back</pre>
)
	body { user agent stylesheet

Doing this connects the Chrome debugger to the running askat-runtime dialog window. Making changes to the CSS will now be reflected in the ask-at-runtime dialog window inside of pdfToolbox Desktop in real time!

