Table of Contents

radixPreflight

1. Introduction

The **radixPreflight** app works in combination with the **calibrate masterProfile** that includes all relevant fixups and checks when preparing and preflighting PDF files for print production.

radixPreflight supports two use cases that can be applied individually or combined.

calibrate templates

- defines all required fixups, checks, exceptions and namedFeatures
- can be based on other templates
- check against common standards like **PDF/X-1a**, **PDF/X-4**, **GWG 2015** or **pdfxReady 2.0.**

calibrate jobTicket

- define your own product specific intents for layout, color, folding and so on
- checks and fixups will be automatically be defined depending on the ticket
- define ticket specific features

In addition, we offer a set of features to fulfill the requirements for offset printing, digital printing or large format printing. The **radixPreflight** app makes it easy to automate PDF preflight in an Enfocus Switch environment.

2. Requirements

The requirements for using the app are as following:

- PDF Toolbox 12 CLI or higher

3. Implementation notes

3.1 Profile

If the script runs as app it's possible to choose the automatic mode for the calibrate masterProfile.
Then the masterProfile.kfpx from the app package will be used for processing.

3.2 Ticket

The ticket can be enabled, disabled or set to the automatic mode.

If the automatic mode is chosen the following datasets will be searched:

- **cmpTicket** - if a calibrate masterProfile ticket is appended to the job as dataset

- **resource** and **orderJson** - if an IWS ticket is defined and should be mapped to a cmpTicket

If found, the dataset (ticket) will be used for processing the job. Named features can be defined as usual.

3.3 Private data

3.3.1 Variables used by the app

`calibrate.cmp.currentFileProductRefId` (optional)

This variable can be used if the ticket is defined for two parts (body and cover), but the incoming PDF should only be checked against one of those.

Example: If the cover PDF should be checked, we set this variable to "cover". Else if the body PDF should be checked, we set it to "body".

3.3.2 Variables set by the app

`com.callassoftware.pdftoolbox.returncode`

Return code of the Callas PDF toolbox CLI. (more information)

`com.callassoftware.pdftoolbox.severity`

Severity level of the processed job.

4. Configuration

4.1 App properties

Each app property is described in the app within the help text.

4.1.1 General

General app properties do not effect the processing of the job, but are neccessary to run the app.

- App path
- Profile path
- User path
- License server

4.1.2 Specific

Specific app properties effect the processing of the PDF and define what should be checked and fixed if templates are used. Otherwise if a ticket is used, the calibrate masterProfile decides what have to be done, you define the requirements for the product and maybe some additional features.

4.2 Outgoing connections

4.2.1 Property `Output`

The property `Output` of each outgoing connection defines if the chosen output should be generated or not. For example if an outgoing connection has an `Output` defined as "HTML", a HTML report will be generated.

There are five categories of output:

- Output
- Reports
- Log
- LogExecution
- Tracing

Output

The processed PDF file will be send to this connection.

Reports

There are multiple different reports that can be chosen:

- HTML
- JSON
- XML
- PDF Report
- PDFMask
- PDFLayer
- PDFComments
- Vardump

General report options

`Dataset`

If a dataset name is defined the report will be appended to the processed PDF file (Output) as dataset.
In addition the option `sendToOut` allows us to define whether the report should be send to the outgoing connection or not.

`Suffix`

Defines a suffix for the generated report, that will be added to the end of the job name.

Additional report options

Additional report type depending options will appear if the certain report is selected in enfocus Switch.

`Template path`

If the report template is chosen from library, a list of all available report templates from the app package will be shown.
For PDF-reports only folders and for JSON/HTML-reports only files will be listed.

Log

The log file of the Callas PDF toolbox CLI will be send to this connection.

LogExecution

This option activates the `--logexecution` option of the Callas PDF toolbox for debugging purposes and sends the log folder to the connection.

Tracing

This option acitivates the `--tracing` option of the Callas PDF toolbox and send the tracing folder to the connection.

4.2.2 Property `Severities`

Defines when the chosen output should be send to the connection.

For example if an outgoing connection has the severity level "success" and the preflight result is "success" (all checks passed, fixups didn't fail), the output will be send to the connection. Else if the severity would be "error", it wouldn't.

There are four possible severity levels:

- success
- warning
- error
- processingError

processingError only exists on the following output options:

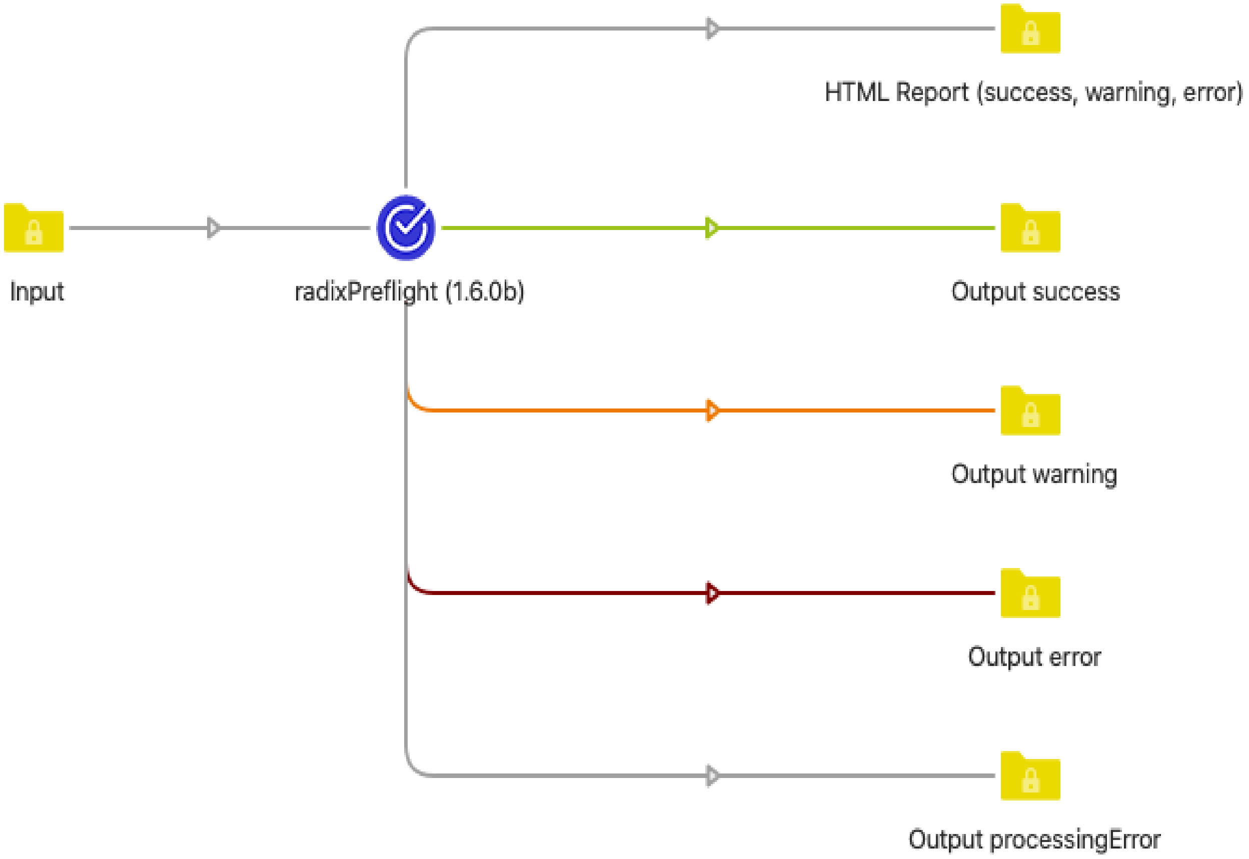
- Output
- Log

---

5. Examples

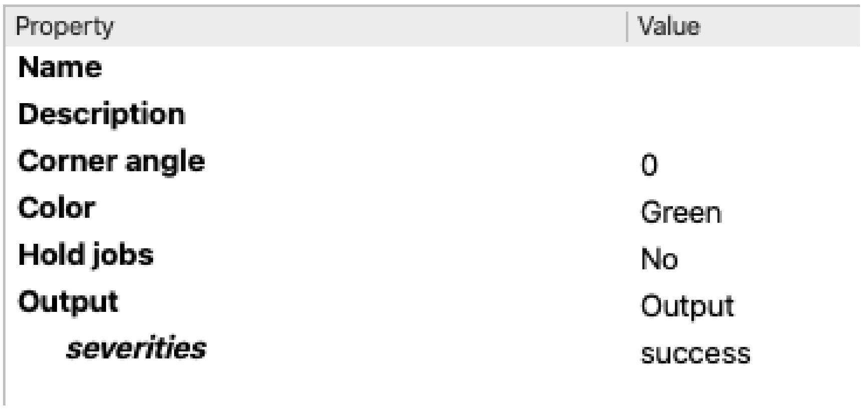
5.1 Flow configuration

The following flow configuration can be used for templates and for job ticket configurations:

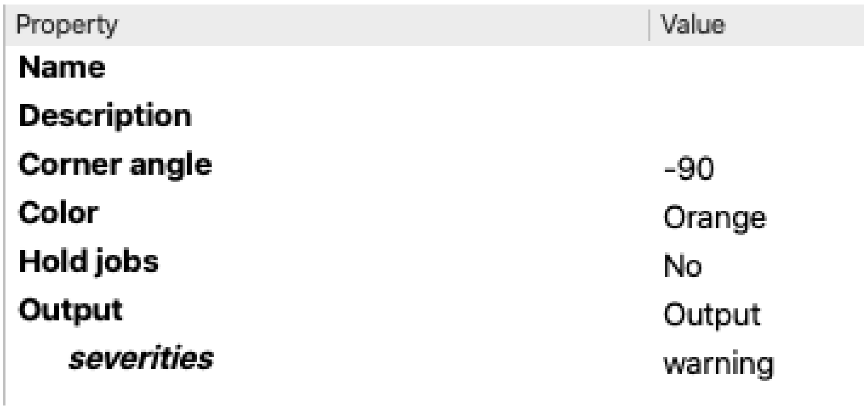


This configuration defines four connections for the processed file (Output), each for every severity level (success, warning, error, processingError). Every flow must have all severity levels for Output defined.

Below are two Output configurations shown:

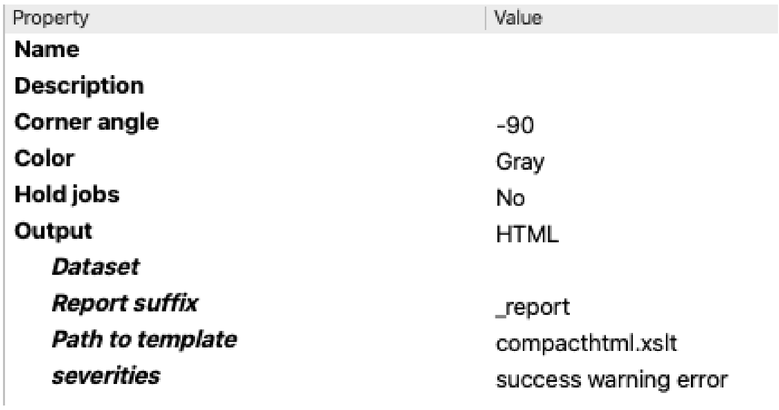


Property	Value
Name	
Description	
Corner angle	0
Color	Green
Hold jobs	No
Output	Output
severities	success



Property	Value
Name	
Description	
Corner angle	-90
Color	Orange
Hold jobs	No
Output	Output
severities	warning

In addition one HTML report is ordered for the severity levels *success*, *warning* and *error*.

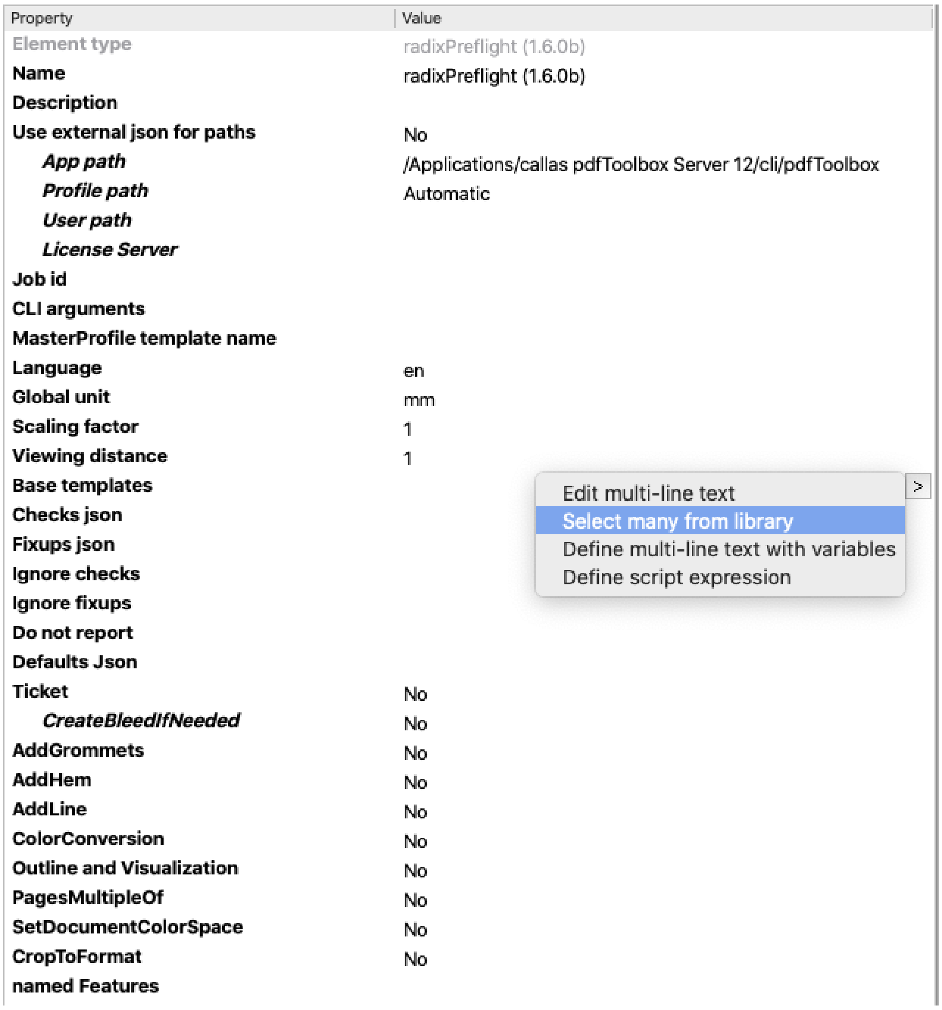


Property	Value
Name	
Description	
Corner angle	-90
Color	Gray
Hold jobs	No
Output	HTML
Dataset	
Report suffix	_report
Path to template	compacthtml.xslt
severities	success warning error

5.2 Working with templates

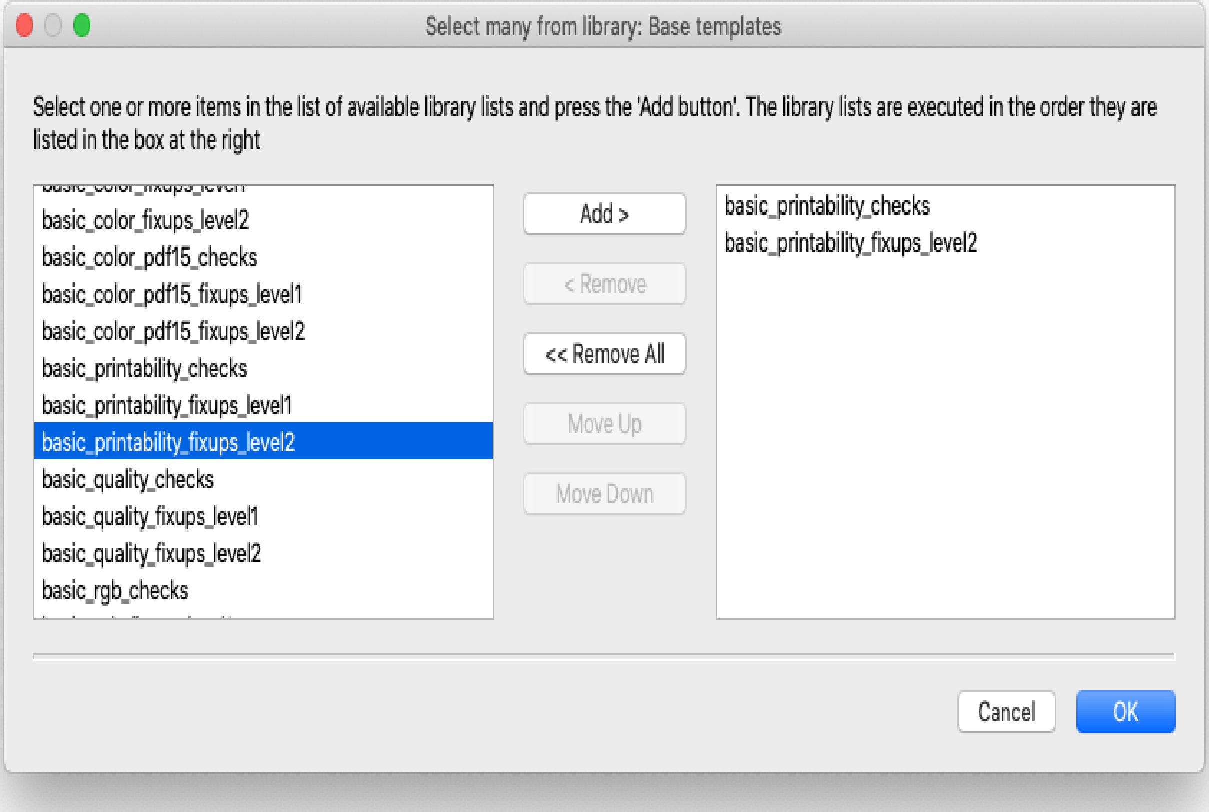
5.2.1 Basic printability

In this example we want to check if the PDF full filles basic printability requirements and if not, fix it to do so. Select "Select many from library" for base templates, to get a list of all templates:



Property	Value
Element type	radixPreflight (1.6.0b)
Name	radixPreflight (1.6.0b)
Description	
Use external json for paths	No
App path	/Applications/callas pdfToolbox Server 12/cli/pdfToolbox
Profile path	Automatic
User path	
License Server	
Job id	
CLI arguments	
MasterProfile template name	
Language	en
Global unit	mm
Scaling factor	1
Viewing distance	1
Base templates	
Checks json	
Fixups json	
Ignore checks	
Ignore fixups	
Do not report	
Defaults Json	
Ticket	No
CreateBleedIfNeeded	No
AddGrommets	No
AddHem	No
AddLine	No
ColorConversion	No
Outline and Visualization	No
PagesMultipleOf	No
SetDocumentColorSpace	No
CropToFormat	No
named Features	

Select "basic_printability_checks" and "basic_printability_fixups_level2":



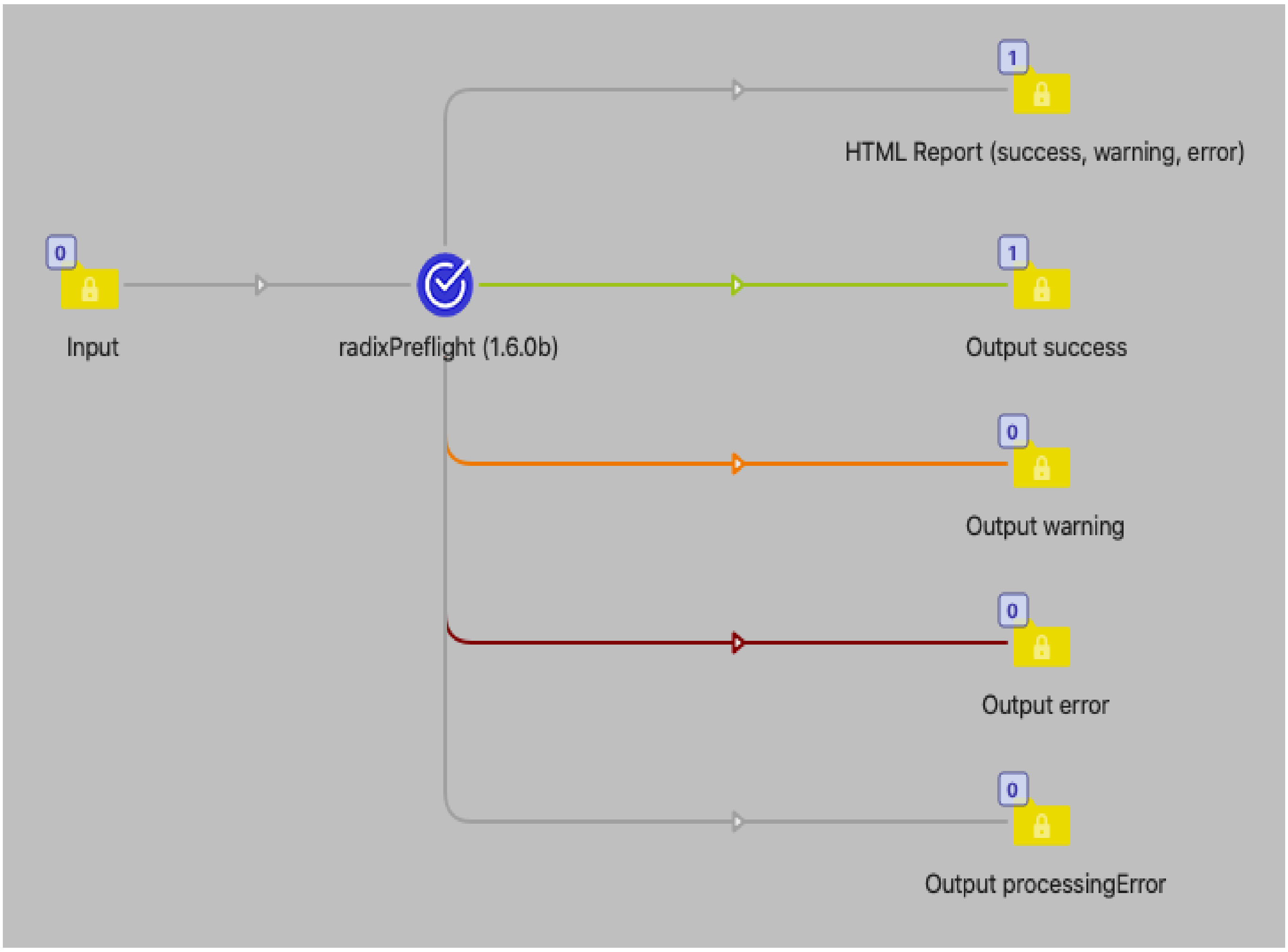
Select many from library: Base templates

Select one or more items in the list of available library lists and press the 'Add button'. The library lists are executed in the order they are listed in the box at the right

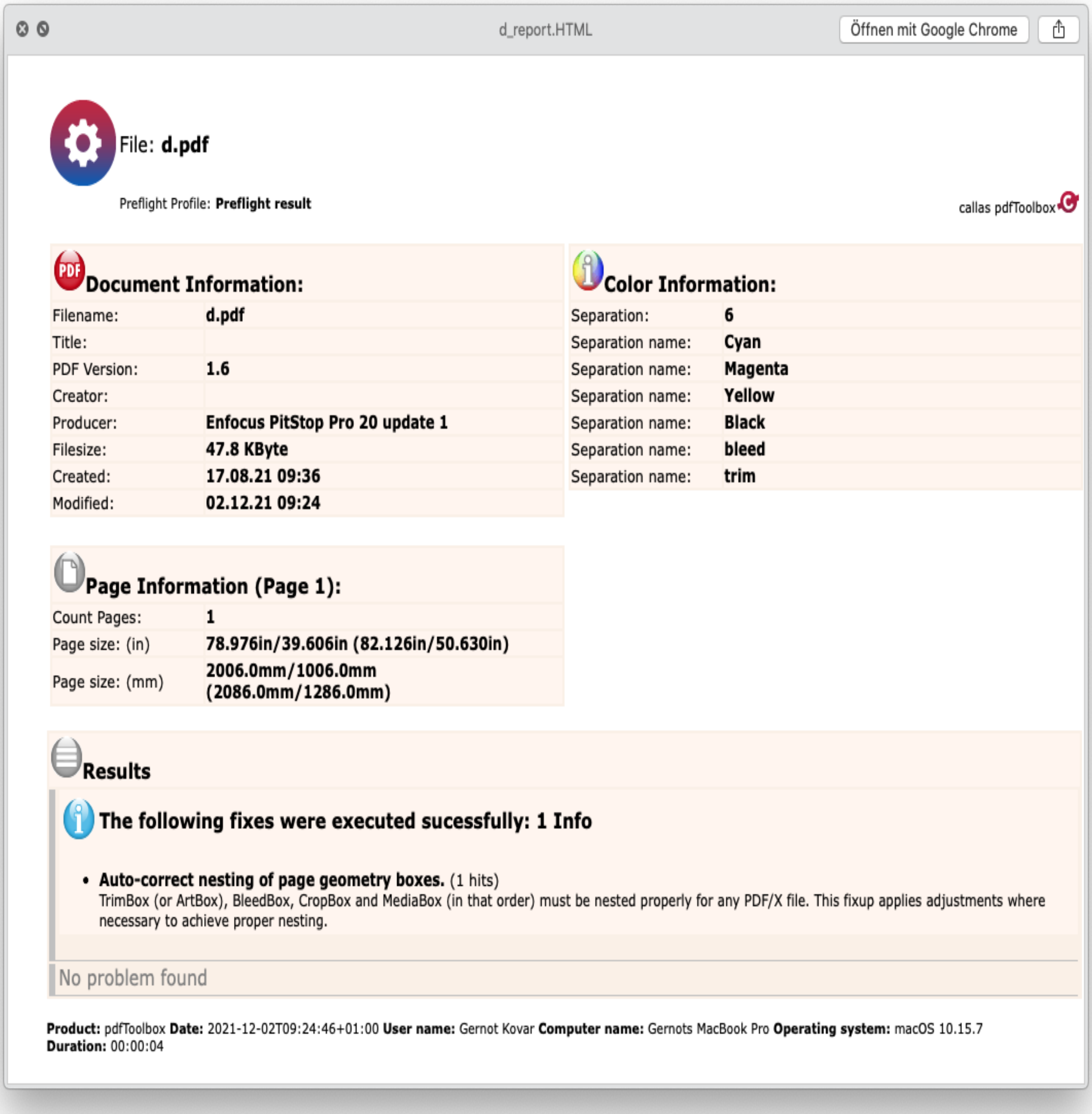
- basic_color_fixups_level1
- basic_color_fixups_level2
- basic_color_pdf15_checks
- basic_color_pdf15_fixups_level1
- basic_color_pdf15_fixups_level2
- basic_printability_checks
- basic_printability_fixups_level1
- basic_printability_fixups_level2
- basic_quality_checks
- basic_quality_fixups_level1
- basic_quality_fixups_level2
- basic_rgb_checks

- basic_printability_checks
- basic_printability_fixups_level2

In that case no warnings, errors or processing errors occured and therefore the processed file is send to "Output success".



As expected the report only shows an information what have been done and does not show any warnings or error:



File: **d.pdf**

Preflight Profile: **Preflight result**

callas pdfToolbox

Document Information:

Filename:	**d.pdf**
Title:	
PDF Version:	**1.6**
Creator:	
Producer:	**Enfocus PitStop Pro 20 update 1**
Filesize:	**47.8 KByte**
Created:	**17.08.21 09:36**
Modified:	**02.12.21 09:24**

Color Information:

Separation:	**6**
Separation name:	**Cyan**
Separation name:	**Magenta**
Separation name:	**Yellow**
Separation name:	**Black**
Separation name:	**bleed**
Separation name:	**trim**

Page Information (Page 1):

Count Pages:	**1**
Page size: (in)	**78.976in/39.606in (82.126in/50.630in)**
Page size: (mm)	**2006.0mm/1006.0mm (2086.0mm/1286.0mm)**

Results

The following fixes were executed sucessfully: 1 Info

- **Auto-correct nesting of page geometry boxes.** (1 hits)
  TrimBox (or ArtBox), BleedBox, CropBox and MediaBox (in that order) must be nested properly for any PDF/X file. This fixup applies adjustments where necessary to achieve proper nesting.

No problem found

Product: pdfToolbox **Date:** 2021-12-02T09:24:46+01:00 **User name:** Gernot Kovar **Computer name:** Gernots MacBook Pro **Operating system:** macOS 10.15.7
Duration: 00:00:04

5.3 Working with a job ticket

5.3.1 Requirements

As already mentioned above, we define the requirements for a product with the ticket.

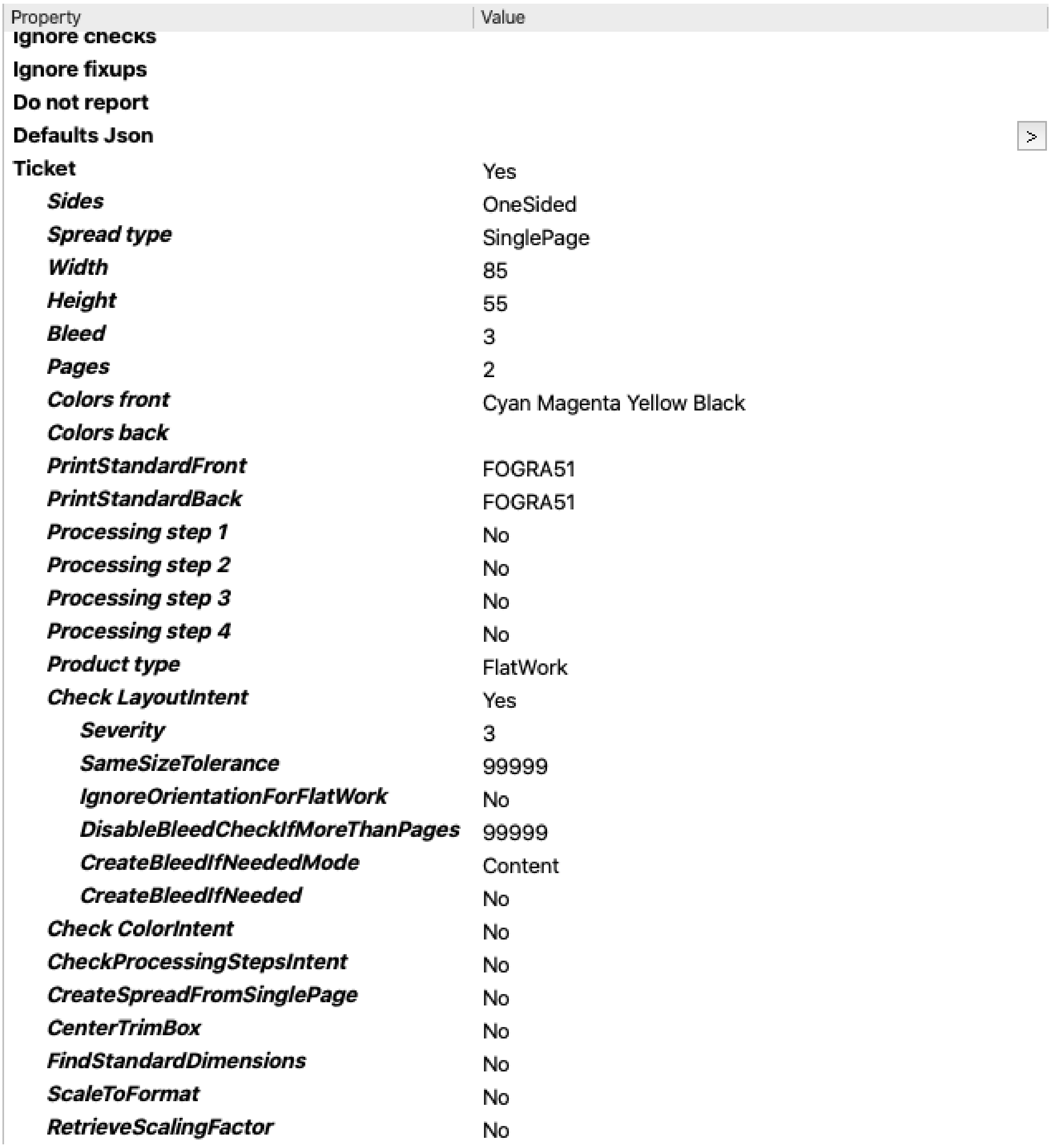
For example if we have a business card:

Expected PDF:

- width: 85mm
- height: 55mm
- bleed: 3mm
- sides: OneSided (no back page).

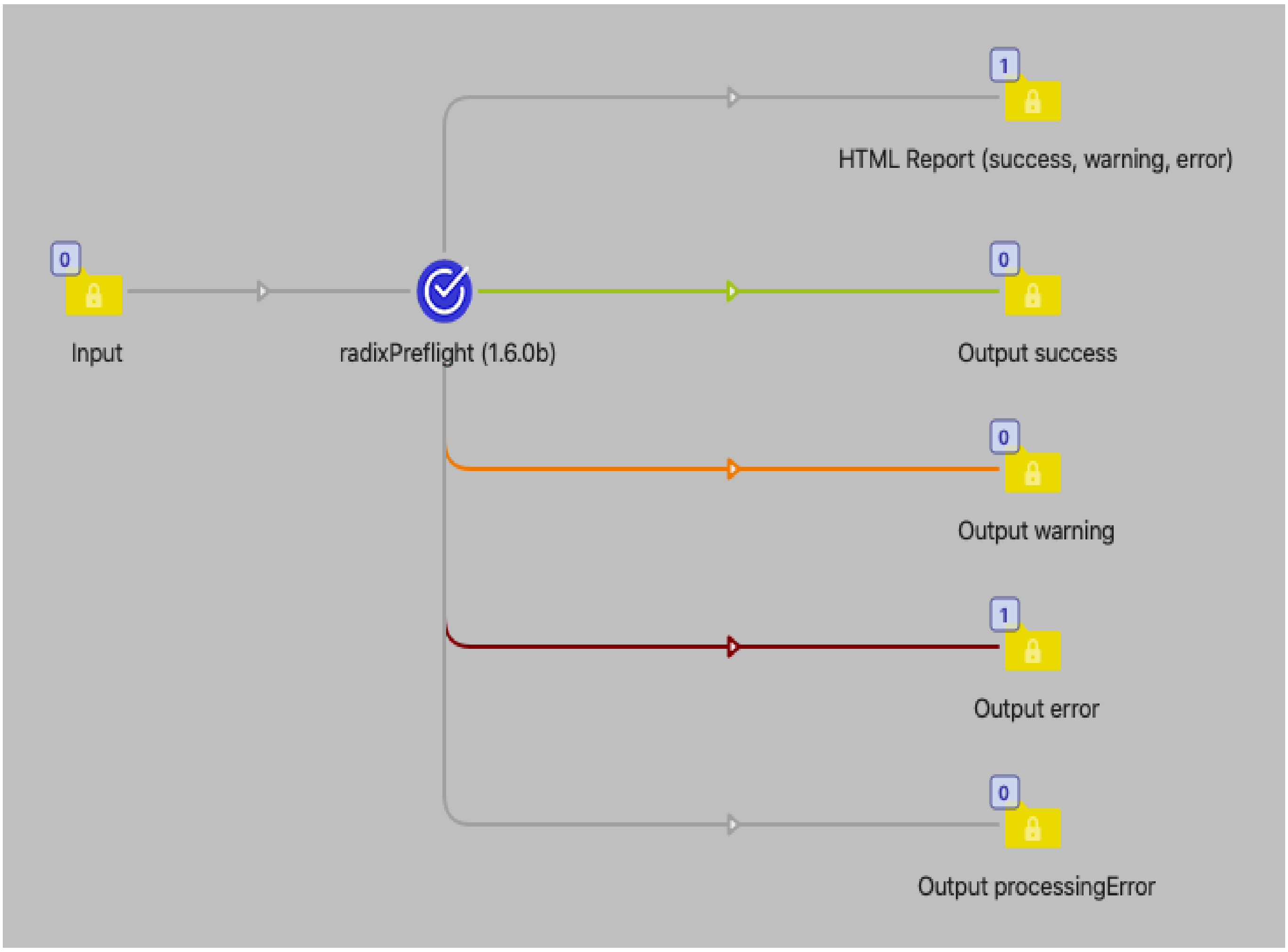
- pages: 2 (The pages property will always be the number of surfaces, even if the PDF has only one page, it must be 2.)

Because we want to check if the PDF meets our layout requirements, we also have to activate the feature "checkLayoutIntent".

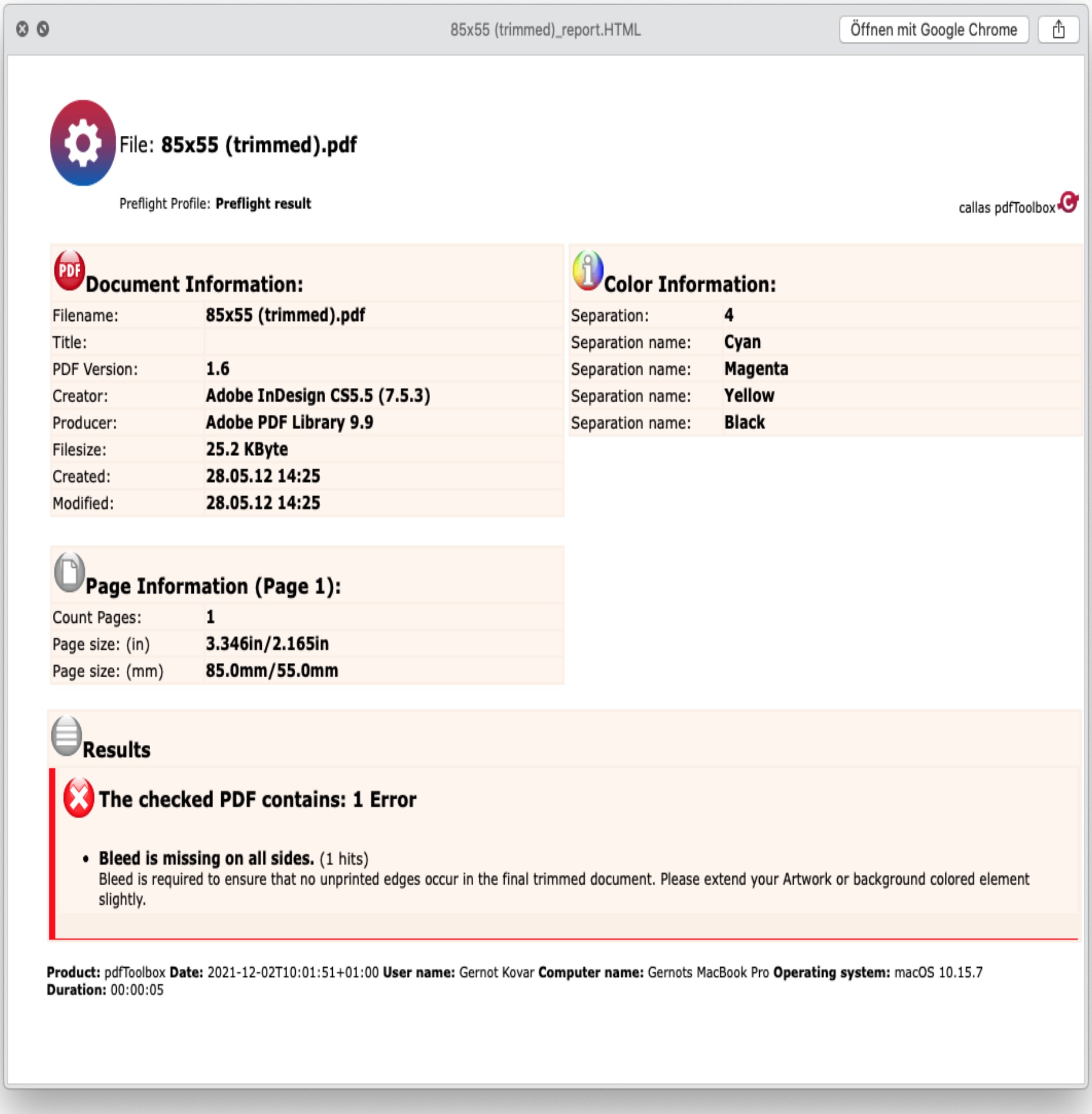


Property	Value
Ignore checks	
Ignore fixups	
Do not report	
Defaults Json	>
Ticket	Yes
Sides	OneSided
Spread type	SinglePage
Width	85
Height	55
Bleed	3
Pages	2
Colors front	Cyan Magenta Yellow Black
Colors back	
PrintStandardFront	FOGRA51
PrintStandardBack	FOGRA51
Processing step 1	No
Processing step 2	No
Processing step 3	No
Processing step 4	No
Product type	FlatWork
Check LayoutIntent	Yes
Severity	3
SameSizeTolerance	99999
IgnoreOrientationForFlatWork	No
DisableBleedCheckIfMoreThanPages	99999
CreateBleedIfNeededMode	Content
CreateBleedIfNeeded	No
Check ColorIntent	No
CheckProcessingStepsIntent	No
CreateSpreadFromSinglePage	No
CenterTrimBox	No
FindStandardDimensions	No
ScaleToFormat	No
RetrieveScalingFactor	No

Our input file did not have any bleed and therefore it was send to "Output Error". It did not full fill our requirements.



The HTML report shows use the result of the preflight:



File: **85x55 (trimmed).pdf**

Preflight Profile: **Preflight result**

Document Information:

Filename:	**85x55 (trimmed).pdf**
Title:	
PDF Version:	**1.6**
Creator:	**Adobe InDesign CS5.5 (7.5.3)**
Producer:	**Adobe PDF Library 9.9**
Filesize:	**25.2 KByte**
Created:	**28.05.12 14:25**
Modified:	**28.05.12 14:25**

Color Information:

Separation:	**4**
Separation name:	**Cyan**
Separation name:	**Magenta**
Separation name:	**Yellow**
Separation name:	**Black**

Page Information (Page 1):

Count Pages:	**1**
Page size: (in)	**3.346in/2.165in**
Page size: (mm)	**85.0mm/55.0mm**

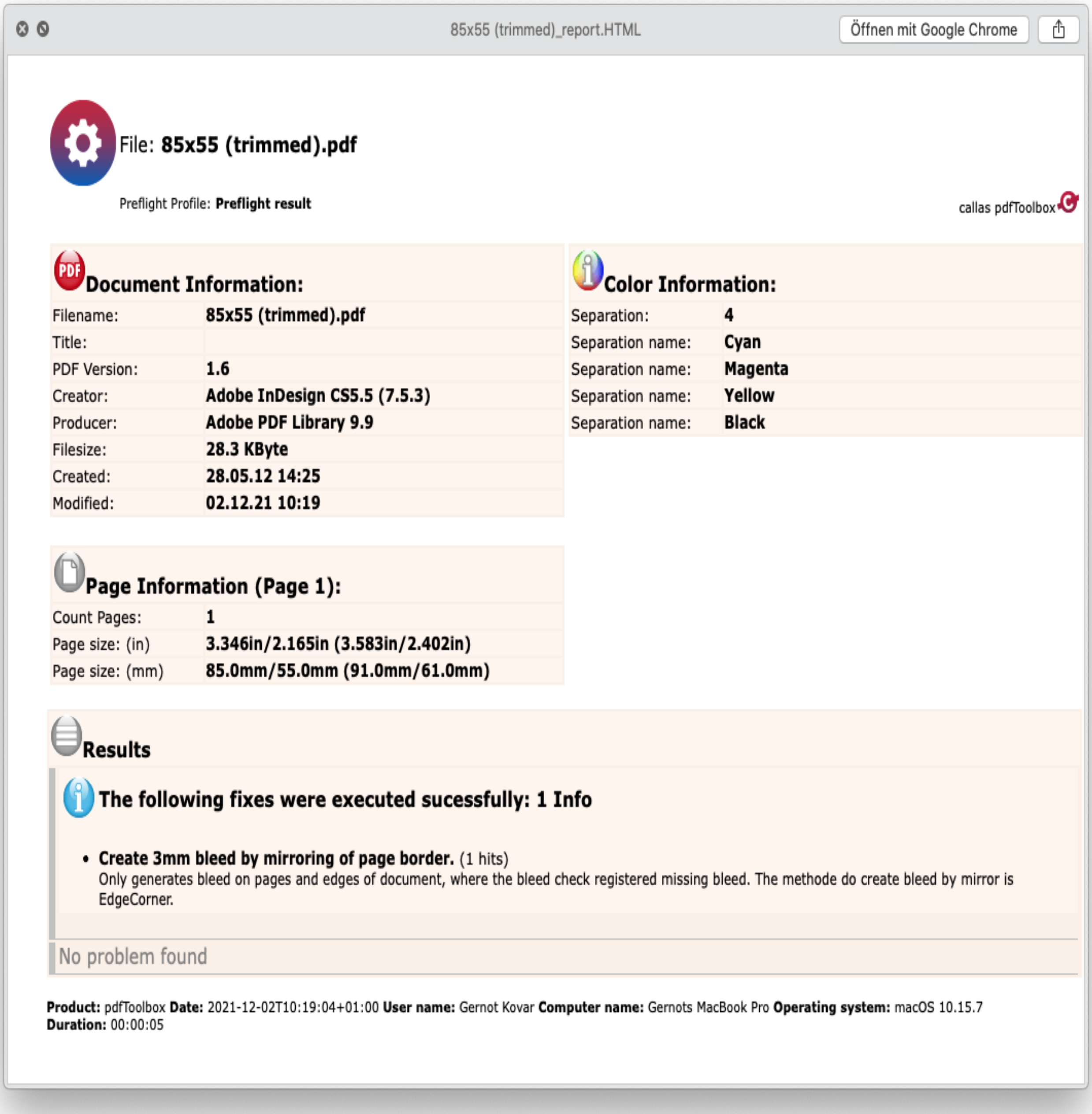
Results

The checked PDF contains: 1 Error

- **Bleed is missing on all sides.** (1 hits)
  Bleed is required to ensure that no unprinted edges occur in the final trimmed document. Please extend your Artwork or background colored element slightly.

Product: pdfToolbox **Date:** 2021-12-02T10:01:51+01:00 **User name:** Gernot Kovar **Computer name:** Gernots MacBook Pro **Operating system:** macOS 10.15.7
Duration: 00:00:05

If we activate "CreateBleedIfNeeded" in the "checkLayoutIntent" feature, the bleed will be created and the file send to "Output success". The HTML report now only shows the information that bleed has been created:



File: **85x55 (trimmed).pdf**

Preflight Profile: **Preflight result**

Document Information:

Filename:	**85x55 (trimmed).pdf**
Title:	
PDF Version:	**1.6**
Creator:	**Adobe InDesign CS5.5 (7.5.3)**
Producer:	**Adobe PDF Library 9.9**
Filesize:	**28.3 KByte**
Created:	**28.05.12 14:25**
Modified:	**02.12.21 10:19**

Color Information:

Separation:	**4**
Separation name:	**Cyan**
Separation name:	**Magenta**
Separation name:	**Yellow**
Separation name:	**Black**

Page Information (Page 1):

Count Pages:	**1**
Page size: (in)	**3.346in/2.165in (3.583in/2.402in)**
Page size: (mm)	**85.0mm/55.0mm (91.0mm/61.0mm)**

Results

The following fixes were executed sucessfully: 1 Info

- **Create 3mm bleed by mirroring of page border.** (1 hits)
  Only generates bleed on pages and edges of document, where the bleed check registered missing bleed. The methode do create bleed by mirror is EdgeCorner.

No problem found

Product: pdfToolbox **Date:** 2021-12-02T10:19:04+01:00 **User name:** Gernot Kovar **Computer name:** Gernots MacBook Pro **Operating system:** macOS 10.15.7 **Duration:** 00:00:05

---

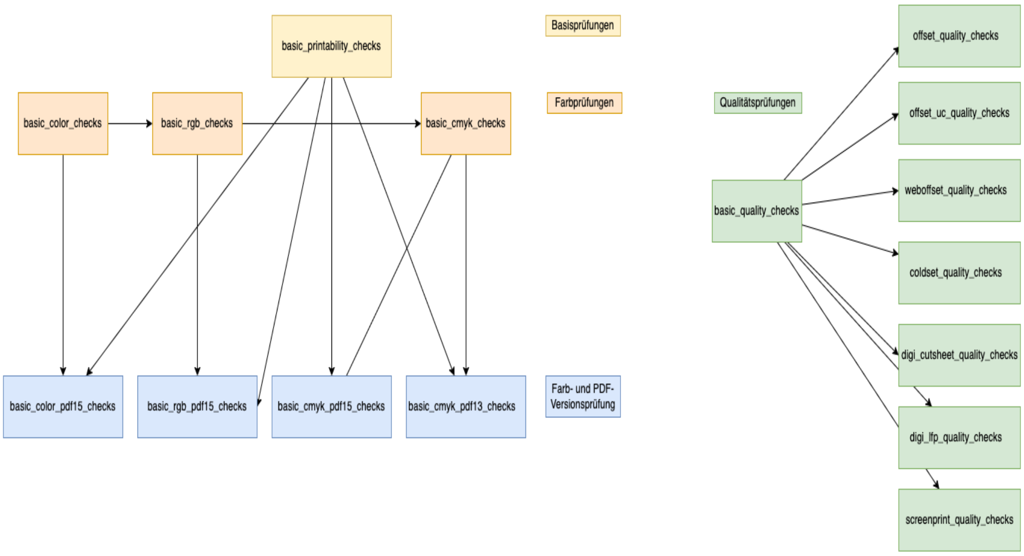
6. Detailed Information

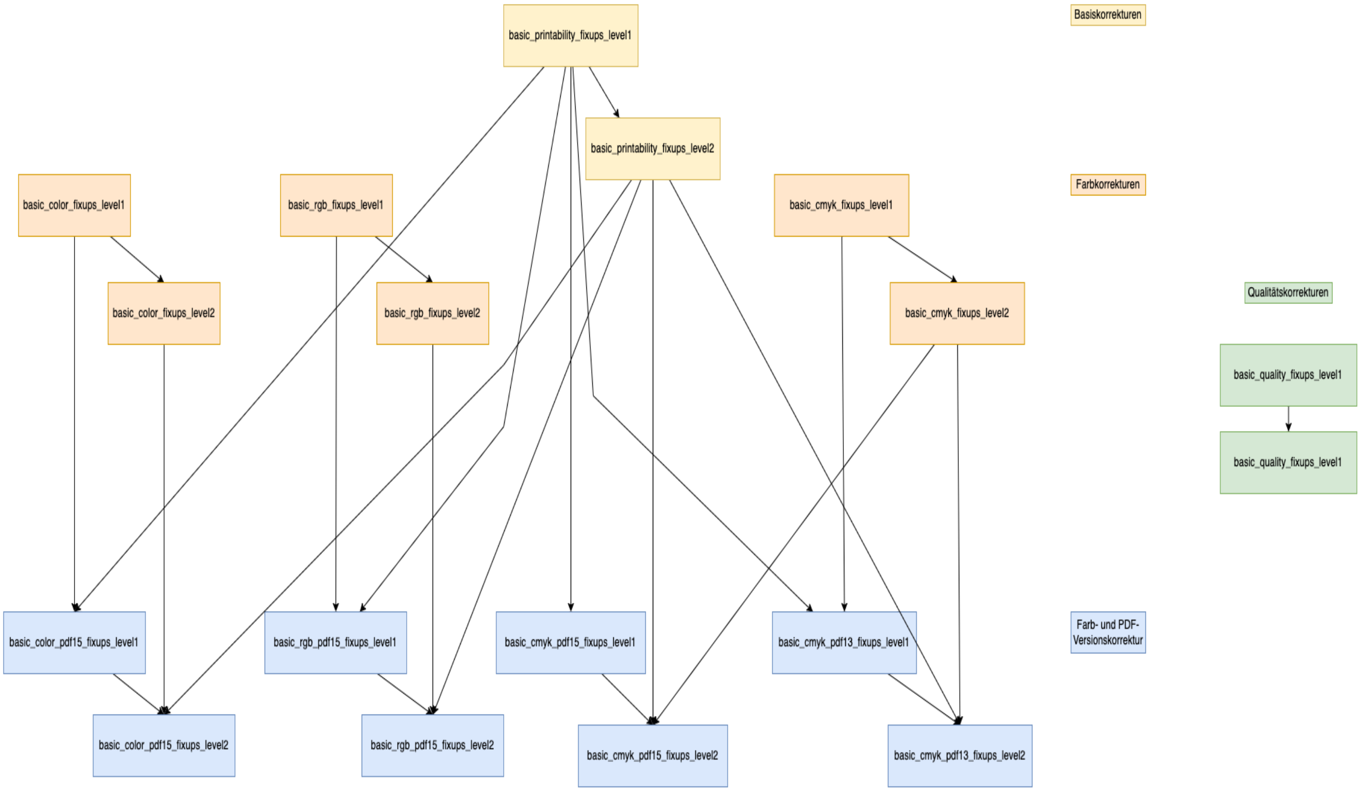
6.1 Templates

A template consists of category-based checks or fixups. This app provides many different templates, which can be put in the following categories:

- Basic
- Colors
- Quality
- PDF versions based

Overview:




basic_printability_fixups_level1
Basiskorrekturen
basic_printability_fixups_level2
basic_color_fixups_level1
basic_rgb_fixups_level1
basic_cmyk_fixups_level1
Farbkorrekturen
basic_color_fixups_level2
basic_rgb_fixups_level2
basic_cmyk_fixups_level2
Qualitätskorrekturen
basic_quality_fixups_level1
basic_quality_fixups_level1
basic_color_pdf15_fixups_level1
basic_rgb_pdf15_fixups_level1
basic_cmyk_pdf15_fixups_level1
basic_cmyk_pdf13_fixups_level1
Farb- und PDF-Versionskorrektur
basic_color_pdf15_fixups_level2
basic_rgb_pdf15_fixups_level2
basic_cmyk_pdf15_fixups_level2
basic_cmyk_pdf13_fixups_level2

6.1.1 Basic

These templates contain basic printability checks and fixups.

Template names:

- **basic_printability_checks**
- **basic_printability_fixups_level1** (fixups marked with Fixup: Level 1)
- **basic_printability_fixups_level2** (fixups marked with Fixup: Level 2)

List of checks and fixups:

- damaged
- secured
- missing font

- missing glyph
- damaged image
- damaged font
- lzw compression
    - Fixup: Level 1: lzwAsZip
- opi
    - Fixup: Level 1 removeOpi
- transfercurve
    - Fixup: Level 1 applyTransfer
- postscriptEmbedded
    - Fixup: Level 1 removePs
- fontWidthUnequal
    - Fixup: Level 2 repairGlyphWidth
- annotations
    - Fixup: Level 2 removeAnnot
- deviceRGB
- trueTypeEncodingInvalid
    - Fixup: Level 2 removeAdditionalEncoding
- spotcolorNameNotUtf8
    - Fixup: Level 1 convertSpotColorNameUtf8
- blendingCsCmyk
- blendingCsCmykSoftmask
- notdefGlyphUsed
    - Fixup: Level 2 fixNotdef
- rotation
    - Fixup: Level 1 applyRotation
- pageScaling
- size and orientation not equal
- CropBoxNeMediaBox
- sourceProfileEqOi
    - Fixup: Level 2 removeIccIfOi
- oiCsNeCmyk
- oiMissingIcc
- unknownPdfOperant
- usesActions

6.1.2 Colors

Checks

Template name: **basic_color_checks**

All color spaces are allowed except RGB without profile and CMYK/Gray with profile.

List of checks:

- jpeg2000DefinesMultipleCs
- spotCount (3 spot colors => warning)

- deviceRgbImage
- deviceRgbTextLineart
- calGrayImage
- calGrayTextLineart
- calRgbImage
- calRgbTextLineart

Template name: **basic_rgb_checks**

Allowed is Gray, CMYK and spot. Images can be defined as ICCbasedRGB. This template extends *basic_color_checks* and therefore includes all the checks of it too.

List of additional checks:

- iccBasedGrayImage
- iccBasedRgbTextLineart
- iccBasedGrayTextLineart
- iccBasedCmykImage
- iccBasedCmykTextLineart
- labTextLineart

Template name: **basic_cmyk_checks**

Allowed is Gray, CMYK and spot. Images defined as ICCbasedRGB are not allowed. This template extends *basic_rgb_checks* and therefore includes all the checks of it too.

List of additional checks:

- iccBasedRgbImage
- labImage

Fixups

As in the examples above we also offer fixups to each check template:

Template_names:

- **basic_color_fixups_level1**
- **basic_color_fixups_level2**
- **basic_rgb_fixups_level1**
- **basic_rgb_fixups_level2**
- **basic_cmyk_fixups_level1**
- **basic_cmyk_fixups_level2**

6.1.3 Quality

The quality templates are provided to check and fix the PDF to full fill the requirements for certain printing methods. Every template for each printing method inherits the checks of the *basic_quality* templates.

Basic quality templates

Template names:

- **basic_quality_checks**
- **basic_quality_fixups_level1**
- **basic_quality_fixups_level2**

General

- emptyPage
- optionalContent - INFO
- courierInsideTrimBox

Overprints

- whiteLineartFillOverprint
- whiteLineartStrokeOverprint
- whiteTextFillOverprint - ERROR
    - Fixup Level 2: weißer Text auf Aussparen
- whiteTextStrokeOverprint - ERROR
    - Fixup Level 2: weißer Text auf Aussparen
- BlackKonly100przKnockoutLteFillText - 12pt
    - Fixup Level 2: schwarzer Text auf Überdrucken
- BlackKonly100przKnockoutLteStrokeLineart - 2pt
- deviceGrayOverprintFill
- deviceGrayOverprintStroke
- deviceGrayBlackKnockoutLteStrokeText - 12pt
- deviceGrayBlackKnockoutLteStrokeLineart - 2pt
- deviceGrayBlackKnockoutLteFillText - 12pt

Resolution

- Platzierte Graustufen und Farbige Bilder
- resolutionHalftoneLevel1 - 200ppi
- resolutionHalftoneLevel2 - 200 - 300ppi - INFO
- Platziertes s/w-Bild auf einer Seite
- resolutionBitmapLevel1 - 550ppi
- resolutionBitmapLevel2 - 550 - 800ppi - INFO
- SingleImage (“die PDF-Seite besteht nur aus einem Bild”)
- resolutionSingleImageLevel1 - 300ppi
- resolutionSingleImageLevel2 - 300-450ppi - INFO

Lines

- minLineWidthTintColor - 100% - 0.25pt
- minLineWidthSingleColor100pzt - 0.125pt
- thinLineMulticolor - 0.25pt
- minLineWidthSingleColor100pzt - 0.25pt

Text

- minTextSizeMultiColor - 8pt
- minTextSizeSingleColor - 5pt
- tintLt100PctText

Color

- spotColorsIdenticalAltanate - ERROR
  - Fixup: Level 1 makeSpotcolorAppearanceConsitstent
- oiMissingIcc
- fillTac - 320%
- strokeTac - 320%
- Lineart400PztFill
- Text400PztFill
- Lineart400PztStroke
- Text400PztStroke
- richBlackTextLineartStroke
- richBlackTextLineartFill
- separatedBlackTextFill
  - Fixup: Level 2 mapCmykAboveToDefinedCmyk
- seperationAll
  - Fixup: Level 2 separationAllToBlack
- spotcolorUsed - INFO
- invalidSpotname
  - Fixup: Level 1 consolidateSpotColorName
- Fixup Level 2: OPM für alle Objekte aktivieren

Printing method templates

List of templates:

- **offset_quality_checks**
- **offset_uc_quality_checks**
- **weboffset_quality_checks**
- **coldset_quality_checks**
- **digi_cutsheet_quality_checks**
- **digi_lfp_quality_checks**
- **screenprint_quality_checks**

6.1.4 PDF versions

PDF 1.3

- No PDF Version higher than 1.4 allowed
- No transparency allowed
- No optional content allowed

List of templates:

- **basic_cmyk_pdf13_checks**

- **basic_cmyk_pdf13_fixups_level1**
- **basic_cmyk_pdf13_fixups_level2**

PDF 1.5

List of templates:

- **basic_color_pdf15_checks**
- **basic_color_pdf15_fixups_level1**
- **basic_color_pdf15_fixups_level2**
- **basic_cmyk_pdf15_checks**
- **basic_cmyk_pdf15_fixups_level1**
- **basic_cmyk_pdf15_fixups_level2**
- **basic_rgb_pdf15_checks**
- **basic_rgb_pdf15_fixups_level1**
- **basic_rgb_pdf15_fixups_level2**

6.1.5 Standards

Check if the PDF full filles the requirementes of a specific ISO or industry standard with the following templates.

List of templates:

- **pdfx_checks**
- **pdfx_fixups_level1**
- **pdfx_fixups_level2**
- **pdfx1a_checks**
- **pdfx1a_fixups_level1**
- **pdfx1a_fixups_level2**
- **pdfx4_checks**
- **pdfx4_fixups_level1**
- **pdfx4_fixups_level2**
- **pdfxready_checks**
- **pdfxready_fixups_level1**
- **pdfxready_fixups_level2**
- **pdfxready_fixups_level3**
- **gwg_digital_cutsheet_2015_cmyk+rgb_sheetfed**
- **gwg_digital_lfp**

6.2 Ticket

A ticket consists of intents, which define the intended values of a product. Click here to get more information about the calibrate jobTicket.

The PDF will be compared with the intended values and if possible fixed or at least reported if it does not match.

Types of intents:

- layoutIntent

- colorIntent
- foldingIntent
- bindingIntent
- mediaIntent
- processingStepsIntent

6.2.1 Intents

LayoutIntent

The layoutIntent includes the following properites:

- sides
- spread type
- width
- height
- pages
- bleed
- finished width
- finsished height
- depth

ColorIntent

The colorIntent includes the following properties:

- colors front
- colors back
- print standard front
- print standard back

6.2.2 Named features (ticket based)

Check LayoutIntent

This feature allows to check that a PDF meets the requirements defined by the layoutIntent. In addition it's possible to define tolerances and fixups if necessary.

Check ColorIntent

This feature allows to check that a PDF meets the requirements defined by the colorIntent.