KOST

Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen

Ein Gemeinschaftsunternehmen von Schweizer Archiven

KOST-VAL   Abridged Manual

Content

> **!** This Manual is only an abridged Manual.
> Check the German or French Manual for more Information.

1   Preface

KOST-Val is a Java-based application for validating the structure and content of TIFF (*Tagged Image File Format*), SIARD (*Software Independent Archiving of Relational Databases*), PDF/A, and JP2 (JPEG 2000; *Joint Photographic Experts Group 2000*) files as well as Submission Information Packages (SIP) for digital information ingest. It is an open source application under a GPL v3+ licence. KOST-Val uses unmodified components of other manufacturers by embedding them directly into the source code. Users of KOST-Val are requested to adhere to these components 'terms of licence. Please refer to chapter 10 for further information.

KOST-Val complies with the following requirements.

<u>TIFF validation:</u> KOST-Val reads a TIFF file and uses JHOVE to validate the structure, the content, and ExifTool to validate the key properties such as compression, colour space, and multipage. These properties can be configured.

<u>SIARD validation:</u> KOST-Val reads a SIARD (eCH-0165[1] v1 and SIARD v2.1[2]) file and validates the structure and the content.

<u>PDF/A validation:</u> KOST-Val reads a PDF or PDF/A file (ISO 19005-1 and 19005-2) and uses 3-Heights™ PDF/A Validator by PDF-Tools or pdfaPilot by callas to validate the structure and the content of the PDF file. KOST-Val organises the different error messages into main categories such as fonts, graphics, and metadata. KOST-Val supplies only limited[3] versions from 3-Heights™ PDF/A Validator by PDF-Tools and pdfaPilot by callas.

Module J extracts and validates if possible the JPEG and JP2 images contained in the PDF file (depending on the configuration)[4]. It is also possible to configure whether the JBIG2 compression[5] is accepted or not.

Module K checks whether the fonts can be searched and extracted (depending on the configuration).

<u>JP2 validation:</u> KOST-Val reads a JP2 file (ISO 15444) and uses Jpylyzer to validate the structure and the content.

<u>JPEG validation:</u> KOST-Val reads a JPEG file (ISO 10918-1) and uses BadPeggy to validate the structure and the content[6].

---

[1] The specification can be downloaded from the eCH website: https://www.ech.ch/ech/ech-0165.

[2] The specification can be downloaded from the KOST website: https://kost-ceco.ch/cms/index.php?siard_de.

[3] The restriction is related mainly to the maximum throughput by 72,000 pages per year. More information on the licenses see Chapter 10. This restriction may be lifted if the 3-Heights™ PDF/A Validator API license by PDF-Tools is acquired and activated with the license manager respectively if pdfaPilot by callas is acquired.

[4] ISO 19005 does not stipulate that the images contained must be valid. The validation of the images is an archival requirement, especially in the use of PDF/A as an image format.

[5] On the problem of the JBIG2 compression see http://kost-ceco.ch/cms/index.php?jbig2-compression_de. The KOST Preservation Planning Expert Group PPEG recommends to renounce the compression type JBIG2 when creating PDF files until further notice.

[6] KOST-Val further evaluates and interprets the error message "Not a JPEG file".

SIP validation: KOST-Val reads an SIP (eCH-0160[7] v1 and v1.1) and validates the mandatory requirements of the SIP specification. The validated requirements are organised into groups such as folder structure, schema validation, and checksum validation. At the outset, a file format validation is performed.

The results (including information on inconsistencies and errors) are output for every step and written into a validation log.

The validation steps are executed sequentially. Whenever possible the validation shall continue after an error has been detected in order to reduce the number of correction cycles.

Please refer to the Annex for more detailed information on the different formats and validation steps.

2 System requirements

- Microsoft Windows 98 or later
- 128 MB RAM or more (512 MB or more is recommended)
- 20 GB disk space or more
- Java Runtime Environment (JRE) Version 8 resp. 1.8 incl. JavaFX or Liberica Full openJDK[8]

3 Open issues / Feedback

Open issues ranging including bugs, requested features, and questions, are listed on the software development platform GitHub at https://github.com/KOST-CECO/KOST-Val/issues and can also be communicated to kost-val@kost-ceco.ch.

These issues are managed by the development team. Any and all contributions are welcome.

---

[7] The specification can be downloaded from the eCH website: https://www.ech.ch/ech/ech-0160.

[8] The following version was successfully tested: https://download.bell-sw.com/java/14.0.2+13/bellsoft-jdk14.0.2+13-windows-amd64-full.msi

4   Introduction dual PDF/A validation

For PDF/A KOST-Val offers the possibility of a dual validation. A PDF/A file is first checked by a first validator. If the result is invalid, a check by a second validator follows. The PDF/A file is considered valid if at least one of the validators identifies it as valid, and as invalid if both validators identify it as invalid.[9]
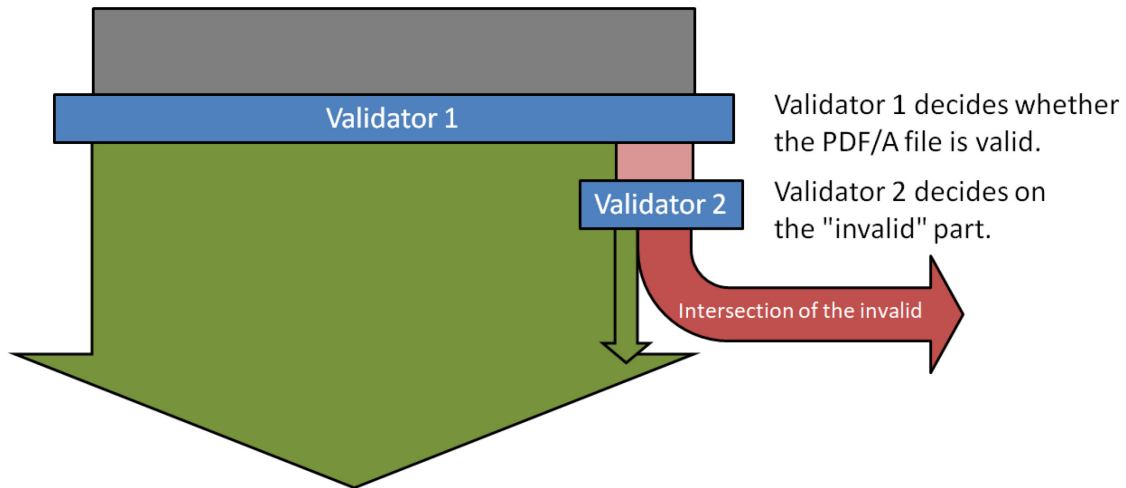
Dual PDF/A validation may only be used if the archive allows potentially invalid PDF/A files to be accepted. If this is not the case, dual PDF/A validation should be avoided.

Both 3-Heights™ PDF/A Validator from PDF-Tools and pdfaPilot from callas are used for dual validation. If only one validator is activated (see 6.1.2), a single validation is automatically performed.

The conceptual basis for dual validation is the observation that even high-quality PDF/A validators can produce different results. This is due on the one hand to the fact that the actual PDF/A standard includes a set of other standards which are not necessarily implemented in the validators in every detail. On the other hand, certain specifications of the standard are formulated in such a way that they can legitimately be implemented in various ways. The fact that all relevant tools implement the specification uniformly and completely remains a pipe dream for the time being. Therefore KOST-Val offers dual validation as an interim solution.

---

[9] The dual validation can only be carried out with high-quality PDF/A validators in this sense. These high requirements are fulfilled among others by the latest versions of 3-Heights™ PDF/A Validator from PDF-Tools and pdfaPilot from callas.

5   Installation of KOST-Val

1	Download KOST-Val and unpack the ZIP file into the relevant folder.
	<table><tr><td>!</td><td>Since KOST-Val Version 1.9.1, the dll files in the 64bit version are in-cluded in the KOST-Val directory. If only a 32bit JRE8 is available, the dll files must be exchanged with the files from the directory "resources\3-Heights_PDF-Validator_dll\Win32". Depending on the au-thorization this is done directly by KOST-Val, otherwise the dll must be exchanged manually.</td></tr></table>

6   Configuration of KOST-Val
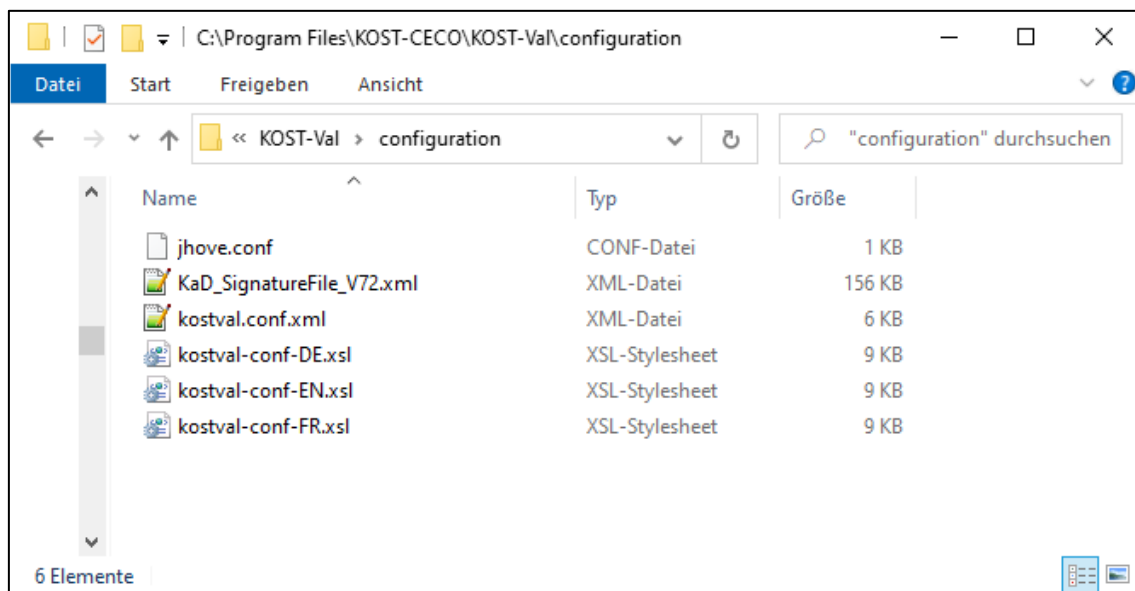
The „configuration" subfolder contains the files "jhove.conf" and "KaD_SignatureFile_V72.xml" that are described in chapter 6.2.

The configuration file "kostval.conf.xml" as well as the three style sheets are copied to the directory "USERHOME/.kost-val_2x/configuration" if not available. All configura-tions of the KOST-Val can be made via GUI.

6.1   Parts of the configuration file "kostval.conf.xml"

Chapter 6.1 has to be updated!

The configuration file "kostval.conf.xml" consists of several parts. The pre-installed con-figuration allows immediate validation of PDF/A, TIFF, SIARD, JP2, JPEG and SIP. The following is a short description of the configuration parts.

6.1.1 General

xml-Tag	Description: default value
\<showprogressonwork\>	Display that KOST-Val works: **yes**

6.1.2 PDF/A

xml-Tag	Description: default value
\<pdftools\>	Specify whether a PDF/A validation with PDF Tools should take place: **yes** If both \<callas\> and \<pdftools\> are set to yes, a dual validation is performed (see also chapter 4).
\<detail\>	Specifies whether an detail error message should be issued by PDF Tools: **yes**
\<callas\>	Specify whether a PDF/A validation with callas should take place: **yes** If both \<callas\> and \<pdftools\> are set to yes, a dual validation is performed (see also chapter 4).
\<nentry\>	Specifies whether an error or warning should be issued by callas if the N entry does not match: **W**
\<pdfa1\>	Specification of minimum conformity level for version 1: **1B**
\<pdfa2\>	Specification of minimum conformity level for version 2: **2U**
\<pdfafont\>	Specifies whether font validation (searchability and extractability) should take place: **no**
\<pdfaimage\>	Indicates if the image validation (JPEG and JP2) should be performed: **no**
\<jbig2allowed\>	Indicates if the JBIG2 compression[10] is allowed: **yes**

---

[10] On the problem of the JBIG2 compression see http://kost-ceco.ch/cms/index.php?jbig2-compression_de. The KOST Preservation Planning Expert Group PPEG recommends to renounce the compression type JBIG2 when creating PDF files until further notice.

6.1.3 SIARD

xml-Tag	Description: default value
<siardvalidation>	Switch for SIARD validation: **yes**

6.1.4 JP2

xml-Tag	Description: default value
<jp2validation>	Switch for JP2 validation: **yes**

6.1.5 JPEG

xml-Tag	Description: default value
<jpegvalidation>	Switch for JPEG validation: **yes**

6.1.6 TIFF[11]

xml-Tag	Description: default value
<tiffvalidation>	Switch for TIFF validation: **yes**
<allowedcompressionX>	Indicates whether compression algorithm X is allowed. By default the following values are allowed: **Uncompressed, CCITT 1D, T4/Group 3 Fax, T6/Group 4 Fax, LZW, PackBits**
<allowedphotointerY>	Indicates whether colour space Y is allowed. By default the following values are allowed: **WhiteIsZero, BlackIsZero, RGB, RGB Palette**
<allowedbitspersampleZ>	Indicates whether Z bits per sample are allowed. By default the following values are allowed: **1, 4, 8, 16**
<allowedmultipage>	Indicates whether multipage TIFF is allowed: **1**
<allowedtiles>	Indicates whether tiled images are allowed: **0**
<allowedsize>	Indicates whether file sizes exceeding 1000MB (~1GB) are allowed: **0**

---

[11] The default values are taken from KOST's Preservation Planning Recommendation (http://kost-ceco.ch/cms/index.php?preservation_de). This uses the Baseline TIFF specification to which it applies restrictions and extensions.

6.1.7 SIP

xml-Tag	Description: default value
<allowedlengthofpaths>	Maximum number of characters in file paths: **179**
<allowedsipname>	Specification of the structure of SIP names: **SIP_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]_\\w{3}**
<allowedformats>	Lists the allowed file formats. The list consists of sub-elements with detailed information. The following are allowed per default: **TXT, PDFA1, PDFA2, TIFF, JP2, JPEG, WAVE, MP3, MP4, MJ2, CSV, SIARD, WARC**

6.1.8 Errors to ignore

xml-Tag	Description: default value
<ignore>	List the detailed error messages from PDF Tools to ignore: **The document does not conform to the requested standard. [PDF Tools: 0x83410612] The value of the key N is 4 but must be 3. [PDF Tools: 0x80410607]**

6.2 "KaD_SignatureFile_V72.xml" and "jhove.conf"

The "configuration" folder also contains the files "KaD_SignatureFile_V72.xml" and "jhove.conf" that do not need adjustment.

"KaD_SignatureFile_V72.xml" is used for format recognition. It is based on DROID, version 28.08.2013, and has been adapted by KOST[12].

"jhove.conf" is used for the internal validation by JHOVE.

---

[12] KOST-Val uses KaD-SignatureFile by KOST instead of the DROID SignatureFile (see https://github.com/KOST-CECO/KaD_SignatureFile). Both files are compatible. KaD-Signature-File comprises only the formats analysed in the KOST Catalogue of archival file formats KaD and permits their recognition in the granularity recommended by KOST. It leads to a significant increase in efficiency as compared to the DROID SignatureFile and thus enhances usability.

7 Resources of KOST-Val

All resources of KOST-Val are stored in the subfolder "resources".
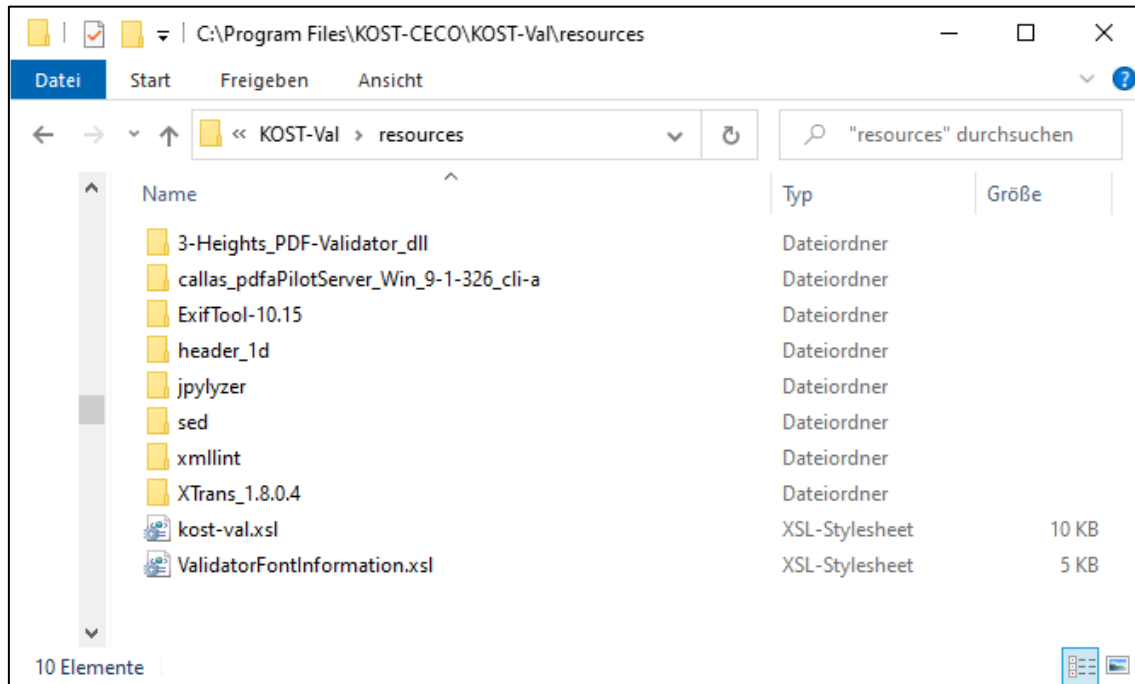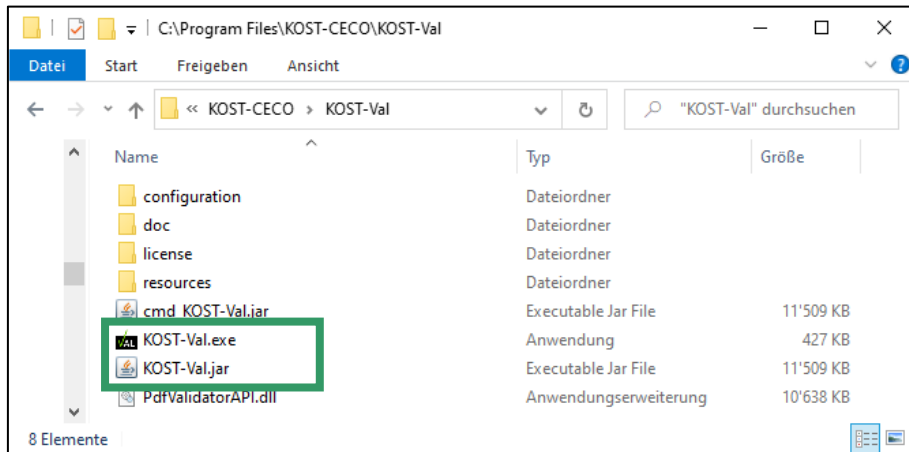
8 Start the validation

!	KOST-Val is not thread safe! That is to say that concurrent instances of KOST-Val cannot be executed without interfering with each other. Concurrent execution of KOST-Val may lead to errors such as a missing working copy.

8.1 KOST-Val GUI

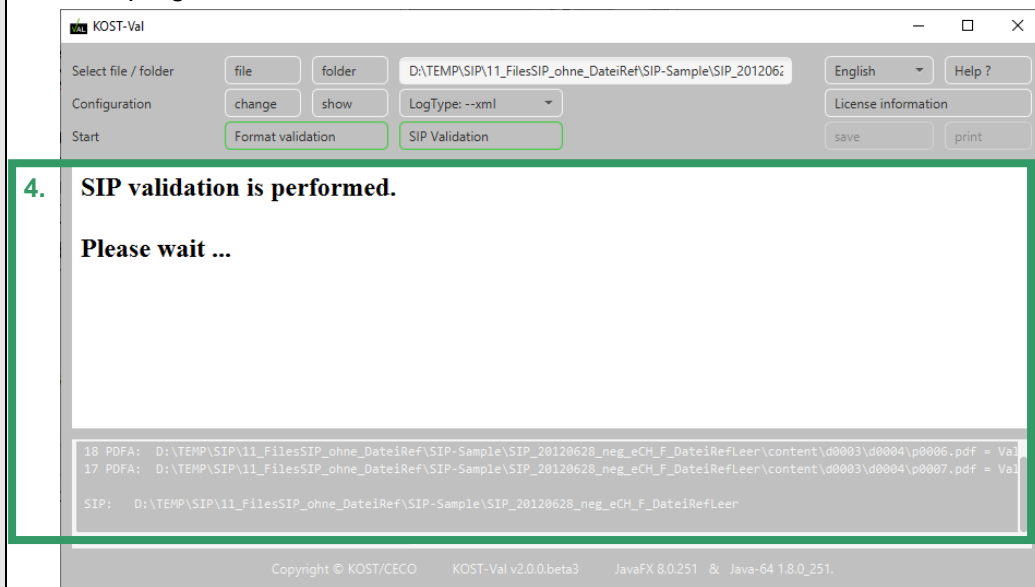| 1	Double click on „KOST-Val.exe"[13] in the folder „KOST-Val" to start KOST-Val.
2	1. Specify / select file or folder for validation 2. Adjust configuration and LogType if necessary 3. Start validation

---
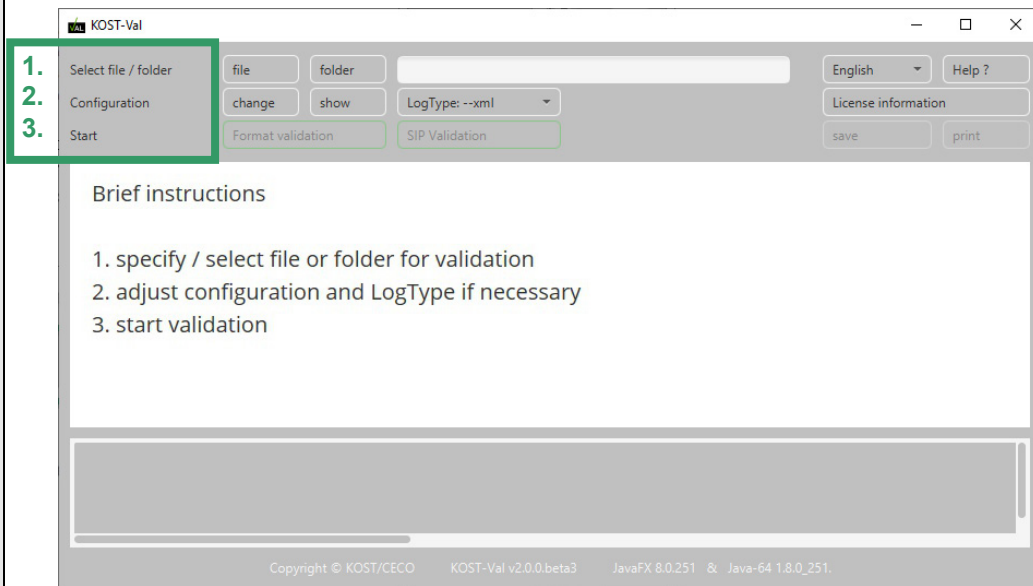
[13] If openJDK is used, the GUI must be started via "KOST-Val.jar" and "Open with...". The java.exe of openJDK must be specified.

	4. Wait until the validation is finished. The progress / current file is shown in the field below.
3	At the end of the validation, the result is displayed. If desired the KOST-Val-Log file can be saved or printed.
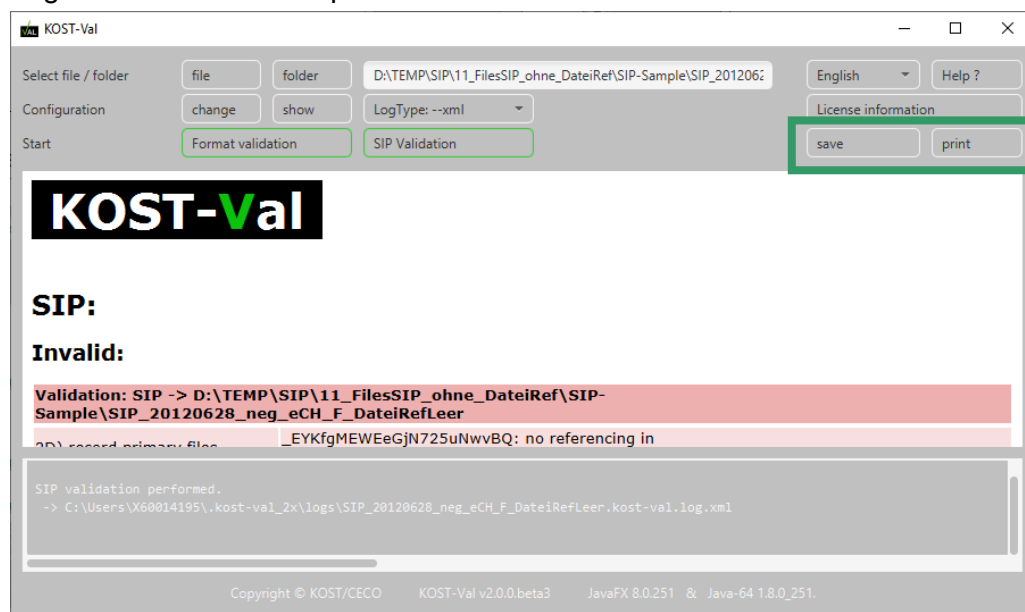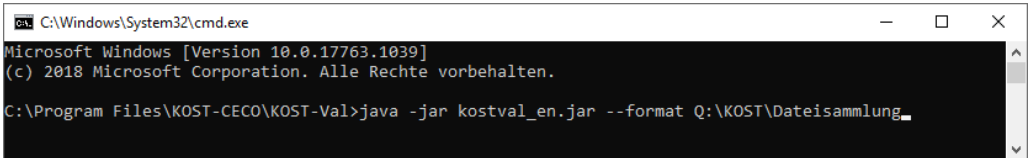
8.2 Start the validation manually

Chapter 8.2 has to be updated!
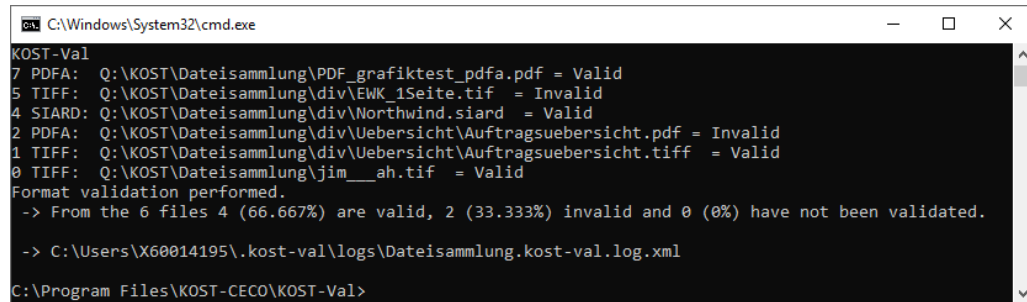
1	Prepare the folder or file to be validated.

2	Open a command prompt and change to the desired working directory (CD "C:\Program Files\KOST-CECO\KOST-Val")[14].

3	Invoke the KOST-Val command (separate command options with spaces).

A java[15] -jar
B kostval_en.jar
C --format (or --sip for SIP validation)
D O:\AMT\Dateisammlung or O:\AMT\Dateisammlung\jim___ah.tif
E nothing or -v (verbose)

A = Java jre8 32bit command (path to java.exe -jar),
B = relative path and file name kostval_en.jar,
C = --format (or --sip for SIP validation),
D = relative path and file name of the folder or file to be validated,
E = verbose option to retain the original reports.

Notes:
A command component that contains spaces needs to be enclosed in quotation marks.

KOST-Val can be invoked from any location. However this may require using absolute paths.

---

[14] To change the drive type, e.g., CD /D.

[15] Invoking `java -jar` is possible only if the desired Java Runtime Environment (JRE) is the standard version.

4	The file has been validated as soon as "Valid" or "Invalid" is displayed in the command window. The folder has been validated as soon as the prompt (C:\Program Files\KOST-CECO\KOST-Val>) is displayed.

```
C:\Windows\System32\cmd.exe                                            —    □    ×
KOST-Val
7 PDFA:  Q:\KOST\Dateisammlung\PDF_grafiktest_pdfa.pdf = Valid
5 TIFF:  Q:\KOST\Dateisammlung\div\EWK_1Seite.tif  = Invalid
4 SIARD: Q:\KOST\Dateisammlung\div\Northwind.siard  = Valid
2 PDFA:  Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.pdf = Invalid
1 TIFF:  Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.tiff  = Valid
0 TIFF:  Q:\KOST\Dateisammlung\jim___ah.tif  = Valid
Format validation performed.
 -> From the 6 files 4 (66.667%) are valid, 2 (33.333%) invalid and 0 (0%) have not been validated.

 -> C:\Users\X60014195\.kost-val\logs\Dateisammlung.kost-val.log.xml

C:\Program Files\KOST-CECO\KOST-Val>
```

Detailed results are available in the file kost-val.log.xml.

The overall result (valid/invalid file) is output as well. In addition, it is visible in the program's exit status in order for the validation to be embedded into an automated process chain. The exit status can take the following values:

    0   everything is ok

    1   incorrect program call

    2   not valid

9 Interpret the validation results

Chapter 9 has to be updated!

1	The validation steps are listed in chapter 11.1.
2	KOST-Val first displays the validation type and the file name in the command prompt output. A validation that has completed without error returns "valid". If a validation identifies an error it returns "Invalid".

```
C:\Windows\System32\cmd.exe                                        —   □   ×

KOST-Val
7 PDFA:  Q:\KOST\Dateisammlung\PDF_grafiktest_pdfa.pdf = Valid
5 TIFF:  Q:\KOST\Dateisammlung\div\EWK_1Seite.tif  = Invalid
4 SIARD:  Q:\KOST\Dateisammlung\div\Northwind.siard  = Valid
2 PDFA:  Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.pdf = Invalid
1 TIFF:  Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.tiff  = Valid
0 TIFF:  Q:\KOST\Dateisammlung\jim___ah.tif  = Valid
Format validation performed.
 -> From the 6 files 4 (66.667%) are valid, 2 (33.333%) invalid and 0 (0%) have not been validated.

 -> C:\Users\X60014195\.kost-val\logs\Dateisammlung.kost-val.log.xml

C:\Program Files\KOST-CECO\KOST-Val>
```

	If the option „-v" (verbose) has been activated, the original validation report is saved as well in the log folder. Since KOST-Val transmits all error messages without modification, this option is of interest only for the specialist user and in particular cases.
3	The log file lists additional detail on every invalid validation step in particular, the affected validation step and the corresponding error.

KOST-Val

Format (PDF/A, TIFF, JP2, SIARD, JPEG):

From the 6 files 4 (66.667%) are valid, 2 (33.333%) invalid and 0 (0%) have not been validated.

Invalid:

Validation: TIFF -> Q:\KOST\Dateisammlung\div\EWK_1Seite.tif

B) Jhove	The Jhove validation has failed (Status: Well-Formed, but not valid).
B) Jhove	- Jhove ErrorMessage: JPEGProc not defined for JPEG compression
C) compression	[EXIF:IFD0] Compression: JPEG (old-style) is not allowed.

Validation: PDFA-2B -> Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.pdf

A) general	PDF/A entry missing [callas]
C) graphics	Device process color used but no PDF/A OutputIntent [callas]
C) graphics	The document contains device-specific color spaces. [PDF Tools: iCategory_3]
C) graphics	- A device-specific color space (DeviceGray) without an appropriate output intent is used. [PDF Tools: 0x03418614]
D) fonts	Font not embedded (and text rendering mode not 3) [callas]
D) fonts	CIDset in subset font is incomplete (font contains glyphs that are not listed) [callas]
D) fonts	The document contains fonts without embedded font programs or encoding information (CMAPs) [PDF Tools: iCategory_8]
D) fonts	- The font ArialMT must be embedded. [PDF Tools: 0x00418704]
D) fonts	- The font Arial-BoldMT must be embedded. [PDF Tools: 0x00418704]
D) fonts	- The font Verdana must be embedded. [PDF Tools: 0x00418704]
D) fonts	- The font Verdana-Bold must be embedded. [PDF Tools: 0x00418704]
D) fonts	- The font Verdana-Italic must be embedded. [PDF Tools: 0x00418704]
H) metadata	The document`s meta data is either missing or inconsistent or corrupt. [PDF Tools: iCategory_16]
H) metadata	- The required XMP property 'pdfaid:part' is missing. [PDF Tools: 0x8341052B]
H) metadata	- The required XMP property 'pdfaid:conformance' is missing. [PDF Tools: 0x8341052B]

Valid:

Validation: PDFA-1B -> Q:\KOST\Dateisammlung\PDF_grafiktest_pdfa.pdf

Validation: SIARD-v1.0 -> Q:\KOST\Dateisammlung\div\Northwind.siard

W) Warning	dataOwner contains the initial value "(...)" and should be filled correctly.
W) Warning	dataOriginTimespan contains the initial value "(...)" and should be filled correctly.

Validation: TIFF -> Q:\KOST\Dateisammlung\div\Uebersicht\Auftragsuebersicht.tiff

Validation: TIFF -> Q:\KOST\Dateisammlung\jim___ah.tif

10 Copyright

Chapter 10 has to be updated!

KOST-Val has been developed by KOST. All rights reserved. KOST-Val has been published by KOST in 2012 under a GNU General Public License v3+.

Notice:	This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

KOST-Val uses the following unmodified components of other manufacturers by embedding them directly into the source code:

Third party application / component	Version	License
3-Heights™ PDF/A Validator API http://www.pdf-tools.com	6.3.1.5	see Chapter 10.1
Apache Commons http://commons.apache.org/ - commons-logging-1.2.jar - commons-lang-3.7.jar - commons-logging-1.2.jar	 1.2 3.7 1.2	Apache License 2.0
Apache log4j http://logging.apache.org/log4j/	1.2.12	Apache License 2.0
Apache Xerces http://xerces.apache.org/	2.7.1	Apache License 2.0
BadPeggy http://coderslagoon.com/	2.0	GPL v3 License
DROID http://digital-preservation.github.io/droid/	5.0.3	3c BSD- License
iText http://itextpdf.com/	5.5.5	AGPL v3 License
Jdom 2.0.0 http://www.jdom.org/	2.0.0	jdom License
Jhove http://hul.harvard.edu/jhove/	1.5	LGPL v2.1 License
Junit 4.12 http://www.junit.org/	4.12	CPL v1.0
Spring Framework API http://static.springsource.org/spring/docs/5.0.x/api/	5.0.8	Apache License 2.0
zip64 http://sourceforge.net/projects/zip64file/	1.02	GPL v2+ License

KOST-Val uses the following unmodified components of other manufacturers which are delivered with KOST-Val:

Third party application / component	Version	License
ExifTool http://www.sno.phy.queensu.ca/~phil/exiftool/	10.15	PERL respective GPL v3.0 License
Jpylyzer http://jpylyzer.openpreservation.org/	1.17.0	LGPL v3.0 License
NSIS v2.46 http://nsis.sourceforge.net/Main_Page	2.46	zlib/libpng License
pdfaPilot CLI https://www.callassoftware.com	7.2	see Chapter 10.2
GNU sed https://www.gnu.org/software/sed	4.4	GPL v3+ License
XML.nsh http://nsis.sourceforge.net/XML_plug-in	2.0	zlib/libpng License
Xmllint http://xmlsoft.org/xmllint.html/	20630	MIT License
XTrans http://sourceforge.net/projects/xtrans/	1.8.0.4	GPL v2 License

Users of KOST-Val are requested to adhere to these components' terms of licence available in the folder KOST-Val\license.

10.1 3-Heights™ PDF/A Validator API License

For the use of the Restricted Version of the 3-Heights™ PDF/A Validator from PDF Tools KOST has made the following individual agreement on the General License Terms with PDF Tools:

2. Individuelle Vereinbarung

Dieses Vertragsverhältnis regelt die Client-Lizenz zwischen der PDF TOOLS als Lizenzgeber und der KOST als Lizenznehmer gemäss nachfolgenden Spezialbestimmungen:

- PDF Tools AG erteilt für KOST eine kostenfreie OEM-Lizenz für das 3-Heights™ PDF/A Validator API als Zusatzfunktion ihrer eigenen Validator-Software (KOST-Val).
- Die Lizenz schliesst den Gebrauch der Software (KOST-Val) durch Gedächtnisinstitutionen, bestehend aus Archiven oder Bibliotheken, deren Zulieferer und der KOST selbst, ein.
- Der OEM-Lizenzschlüssel, welcher fest in KOST-Val eingebunden ist, darf nicht ausserhalb der Applikation (KOST-Val) verwendet werden.
- Die Lizenz ist zeitlich unbegrenzt, jedoch bezüglich Durchsatz pro Installation begrenzt (72'000 Seiten pro Jahr).
- Für die Verteilung der Software (KOST-Val) an den Anwender ist die KOST zuständig.
- Der First Level Support der Anwender erfolgt durch KOST. Second Level Support Fälle leitet KOST an PDF Tools AG weiter.
- Wenn der Anwender weitergehende Bedürfnisse hat, z.B. höherer Durchsatz, Integration in andere Applikationen etc. kauft er die Software (3-Heights™ PDF/A Validator API) direkt bei PDF Tools AG.
- Die KOST darf weiterhin den Quellcode von KOST-Val Open Source publizieren und KOST-Val gratis und ohne Registrierung abgeben.

The following points are decisive for users:
- The license includes the use of the software (KOST-Val) by memory institutions consisting of archives or libraries, their suppliers and KOST itself.
- The OEM licence key, which is firmly integrated in KOST-Val, may not be used outside the application (KOST-Val).
- The license is unlimited in time, but limited in throughput per installation (72'000 pages per year).
- The first level support of the users is provided by KOST. Second Level Support cases are forwarded by KOST to PDF Tools AG.
- If the user has additional requirements, e.g. higher throughput, integration in other applications, etc., he or she can purchase the software (3-Heights™ PDF/A Validator API) directly from PDF Tools AG.

Users of KOST-Val are required to comply with this license agreement.

10.2 pdfaPilot CLI License

For the use of the Restricted Version of the pdfaPilot CLI from callas KOST has agreed the following Individual Agreement on the General License Conditions with callas:

---

2. Individuelle Vereinbarung

Dieses Vertragsverhältnis regelt die Lizenz zwischen der callas software als Lizenzgeber und der KOST als Lizenznehmer gemäss nachfolgenden Spezialbestimmungen:

- callas software erteilt für die KOST eine kostenfreie Lizenz für callas pdfaPilot CLI für Windows zur innerbetrieblichen Nutzung und zur Integration in ihren eigenen Validator „KOST-Val".
- Die Lizenz schliesst die Distribution von KOST-Val an „Anwender" (Gedächtnisinstitutionen, Archive oder Bibliotheken und deren Zulieferer) ein.
- Für die Distribution von KOST-Val an diese Anwender ist die KOST zuständig und darf KOST-Val auch gratis und ohne Registrierung an diese abgeben.
- Die Lizenz ist zeitlich unbegrenzt, jedoch bezüglich Durchsatz pro Installation begrenzt auf 72'000 Seiten pro Jahr.
- Die KOST darf den eigenen Quellcode von KOST-Val Open Source publizieren. callas pdfaPilot CLI ist hiervon ausgenommen.
- First Level Support der Anwender erfolgt durch die KOST. Second Level Support leistet callas software gegenüber der KOST.

The following points are decisive for users:
- The license includes the distribution of KOST-Val to "users" (memory institutions, archives or libraries and their suppliers).
- The license is unlimited in time, but limited in throughput per installation to 72'000 pages per year.
- KOST may publish the own source code of KOST-Val Open Source. callas pdfaPilot CLI is excluded from this.
- First Level Support of the users is provided by KOST. Second level support is provided by callas software to KOST.

The users of KOST-Val are obliged to comply with this license agreement.

11 Annex

11.1 Program structure

KOST-Val is structured according to the following requirements:

Functional requirements:

TIFF validation: KOST-Val reads a TIFF file and uses JHOVE to validate the following:

Validation step		Description
A	(exit on error)	Recognition
B		Jhove
C		Compression
D		Colour space
E		BitsPerSample
F		Multipage
G		Tiles
H		File size

SIARD validation: KOST-Val reads a SIARD file and validates the following:

Validation step		Description
A	(exit on error)	Readability
B	(exit on error)	Primary folder structure
C	(exit on error)	Validation of header
D	(exit on error)	Validation of structure
E		Validation of table columns
F		Validation of table rows
G		Validation of tables
H		Validation of content
I		Recognition of SIARD
J		Additional primary data
W		Warning

PDF/A validation: KOST-Val reads a PDF file and uses 3-Heights™ PDF/A Validator by PDF-Tools or pdfaPilot by callas to validate the following:

Validation step		Description
A	(exit on error)	General
B		Structure
C		Graphics
D		Fonts
E		Transparency
F		Annotations
G		Actions & Interactions
H		Metadata
I		Accessibility
J	(configurable)	Image & JBIG2
K	(configurable)	Font validation

JP2 validation: KOST-Val reads a JP2 file and uses Jpylyzer to validate the following:

Validation step	Description
A (may exit on error)	Recognition and Jpylyzer
B	Metadata
C	Image
D	Extended

JPEG validation: KOST-Val reads a JPWG file and uses BadPeggy to validate the following:

Validation step	Description
A (may exit on error)	Recognition and BadPeggy
B	Corrupt data
C	Invalid file structure
D	Other problems

SIP validation: KOST-Val reads an SIP file and validates the following requirements of the SIP specification:

Validation step	Description (name of step)
1a (exit on error)	Readability
1b (exit on error)	Primary folder structure
1c (exit on error)	Folder and file names
1d (exit on error)	Schema validation of metadata.xml
1e	Determine type of SIP
1f	Primary data in folder
2a	Missing primary data
2b	Additional primary data
2c	Validation of checksums
2d	Recording primary data
3a	Format recognition
3b	Additional formats
3c	Format validation
3d	Validation of range of dates

For every step the results (including information on inconsistencies and errors) are output and written into a validation log.

The overall result (valid/invalid file) is output as well. In addition, it is visible in the program's exit status in order for the validation to be embedded into an automated process chain. The exit status can take the following values:

- 0 everything is ok
- 1 incorrect program call
- 2 not valid

The validation steps are executed sequentially. Whenever possible the validation shall continue after an error has been detected in order to reduce the number of correction cycles.

Non-functional requirements:

External programs or java frameworks are used for particular tasks.

The application has a modular structure that allows for inserting additional validation modules without further ado.

The validation log and exit status permit an easy readout of a single validation result and allow the utilisation of the tool in a process chain.

The console output is limited on the validation module, the final results of either "valid" or "invalid" and the path to the file. All additional information is documented in the log file.

11.2 Functional Principle of Format Validation