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 PDF/A, and JP2 (JPEG 2000; *Joint Photographic Experts Group 2000*), JPEG, TIFF (*Tagged Image File Format*), PNG (*Portable Network Graphics*), XML (*Extensible Markup Language*), SIARD (*Software Independent Archiving of Relational Databases*) 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 9 for further information.

KOST-Val complies with the following requirements.

PDF/A validation: 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[1] versions from 3-Heights™ PDF/A Validator by PDF-Tools and pdfaPilot by callas.

In module J it is possible to configure whether the JBIG2 compression[2] is accepted or not. Module K checks whether the fonts can be searched and extracted (depending on the configuration).

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

JPEG validation: KOST-Val reads a JPEG file (ISO 10918-1) and uses BadPeggy to validate the structure and the content[3].

TIFF validation: 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.

PNG validation: KOST-Val reads a PNG (ISO 15948) and uses pngcheck to validate the structure and content.

XML validation: KOST-Val reads a XML and uses xmllint and XSD to validate the structure and content.

SIARD validation: KOST-Val reads a SIARD (eCH-0165[4] v1 and SIARD v2.1 as well as v2.2[5]) file and validates the structure and the content.

---

[1] The restriction is related mainly to the maximum throughput by 72,000 pages per year. More information on the licenses see Chapter 9. This restriction may be lifted if the 3-Heights™ PDF/A Validator API license by PDF-Tools is acquired and activated with the LicenseManager (resources\3-Heights_PDF-Validator_dll) respectively if pdfaPilot by callas is acquired.

[2] 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.

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

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

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

<u>SIP validation:</u> KOST-Val reads an SIP (eCH-0160[6] v1.0, v1.1 and v1.2) 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

- 64bit Microsoft Windows
- At least 512 MB RAM
- At least 20 GB hard disk space
-

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.

---

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

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.[7]
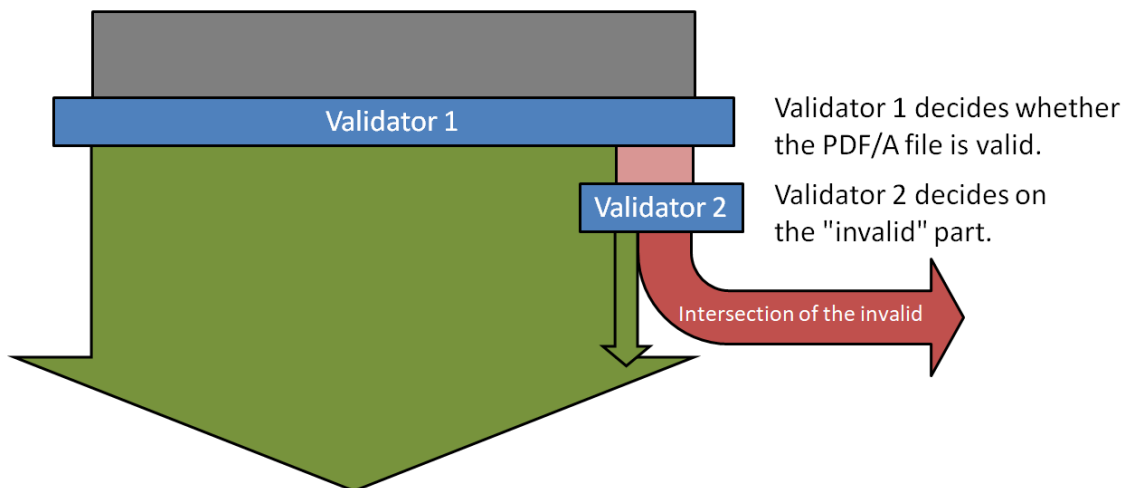
Dual PDF/A validation may only be used if the archive allows potentially invalid PDF/A files to be accepted. If this is not the case, dual PDF/A validation should be avoided.

Both 3-Heights™ PDF/A Validator from PDF-Tools and pdfaPilot from callas are used for dual validation. If only one validator is activated, 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.

---

[7] 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	KOST-Val (version 2.1.3.0 and newer) is only offered in the 64bit installation package KOST-Tools.msi[8].
	https://github.com/KOST-CECO/KOST-Val/releases/latest
	After downloading KOST-Tools the installation package must be executed with administration rights.
	KOST-Val is then available in the start menu under KOST-Tools.

6   Configuration of KOST-Val
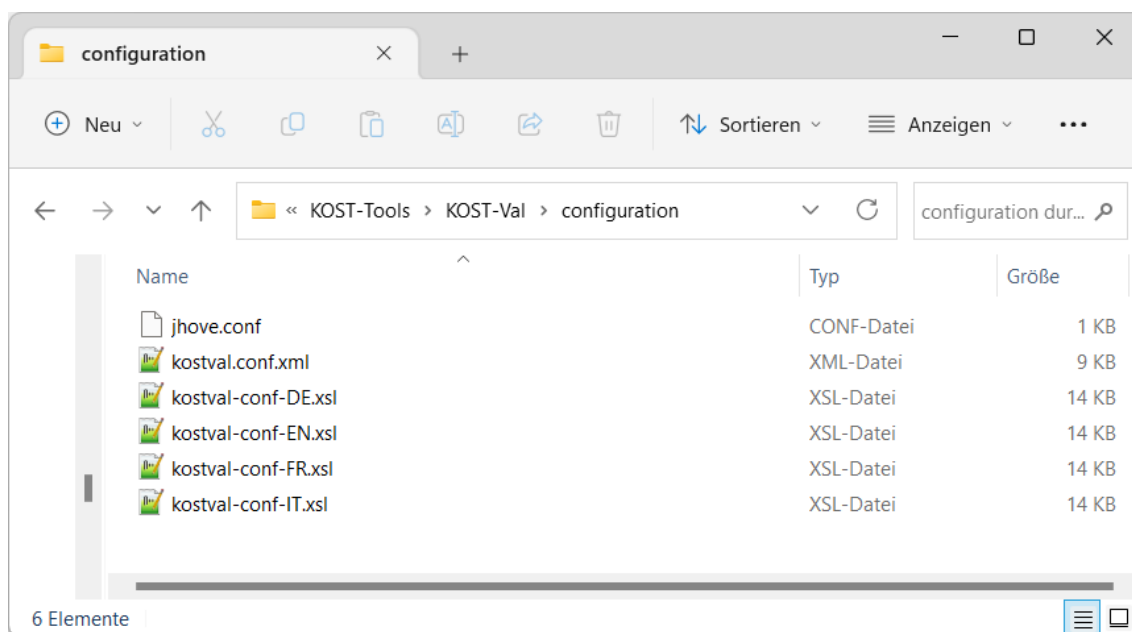
The "configuration" folder contains the file "jhove.conf" that do not need adjustment.

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

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

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

The configuration file "kostval.conf.xml" consists of several parts. The pre-installed configuration allows immediate validation of PDF/A, JP2, JPEG, TIFF, PNG, SIARD and SIP. In addition, FLAC, WAVE, MP3, MKV, MP4, XML, JSON, CSV, XLSX, WARC, HTML and DWG are accepted in folders and SIPs. The following is a short description of the configuration parts.

---

[8] More detailed instructions on the installation and its scope can be found in the KOST-Tools manual.

KOST-Val Configuration

PDF/A: Acceptance and validation [✓]	✓
PDF/A validation with PDF Tools [yes]:	yes
- PDF Tools also detailed errors in English [yes]:	yes
- Validation (searchability and extractability) [tolerant]:	tolerant
PDF/A validation with callas [yes]:	yes
- Error (E) / warning (W) if N entry does not match [W]:	W
Allowed PDF/A versions [1A, 1B, 2A, 2B, 2U]:	1A 1B 2A 2B 2U
Validate PDF/A-3 to PDF/A-2 and generate warning message instead of error [yes]:	yes
JBIG2 compression allowed [yes]:	yes
TXT: Acceptance [(✓)]	(✓)
PDF: Acceptance [✗]	✗
JPEG2000: Acceptance and validation [✓]	✓
JPEG: Acceptance and validation [✓]	✓
TIFF: Acceptance and validation [✓]	✓
Allowed compression algorithms [Uncompressed, CCITT 1D, T4/Group 3 Fax, T6/Group 4 Fax, LZW, PackBits]:	Uncompressed CCITT 1D T4/Group 3 Fax T6/Group 4 Fax LZW PackBits
Allowed color spaces [WhiteIsZero, BlackIsZero, RGB, RGB Palette]:	WhiteIsZero BlackIsZero RGB RGB Palette
Bits per sample allowed [1, 4, 8, 16]:	1 4 8 16
Multipage TIFFs allowed [yes]:	yes
Build in tiles allowed [no]:	no
File sizes of 1000MB (~1GB) and larger allowed [no]:	no
PNG: Acceptance and validation [✓]	✓
FLAC: Acceptance [(✓)]	(✓)
WAVE: Acceptance [(✓)]	(✓)
MP3: Acceptance [(✓)]	(✓)
MKV (FFV1-FLAC): Acceptance [(✓)]	(✓)
MP4 (AVC-MP3): Acceptance [(✓)]	(✓)
XML: Acceptance and validation [(✓)]	(✓)
JSON: Acceptance [(✓)]	(✓)
SIARD: Acceptance and validation [✓]	✓
Allowed SIARD versions [1.0, 2.1, 2.2]:	1.0 2.1 2.2
CSV: Acceptance [(✓)]	(✓)
XLSX: Acceptance [(✓)]	(✓)
ODS: Acceptance [✗]	✗
SIP: Validation [✓]:	✓
Allowed maximum number of characters in path lengths [179]:	179
Specifications for the structure of the SIP name [SIP_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]_\w{3}]:	SIP_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]_\w{3}
Only warning for old documents (Entstehungszeitraum) [no]:	no
Other accepted file formats [WARC, HTML, DWG]:	HTML WARC DWG
Working directory []:	
Input directory []:	
Calculate and output hash value of files. Empty means no calculation and output []:	

KOST-Val - Configuration - PDF/A

Validation setting: PDF/A

PDF Tools	Callas	Versions	Other
✓ PDF Tools	✓ Callas	✓ PDF/A-1a	✓ JBIG2
└ ✓ details	└ ☐ N-Entry	✓ PDF/A-1b	
└ ✓ Font		✓ PDF/A-2a	
└ ✓ Tolerant		✓ PDF/A-2b	
		✓ PDF/A-2u	
		✓ (PDF/A-3 ≈ PDF/A-2)	

KOST-Val - Configuration - TIFF

Validation setting: TIFF

Compression algorithm	Color space	Bits per sample (per channel)	Other
✓ Uncompressed	✓ WhiteIsZero	✓ Bps 1	✓ Multipage
✓ CCITT 1D	✓ BlackIsZero	☐ Bps 2	☐ Tiles
✓ T4/Group 3 Fax	✓ RGB	✓ Bps 4	☐ Size
✓ T6/Group 4 Fax	✓ RGB Palette	✓ Bps 8	
✓ LZW	☐ transparency	✓ Bps 16	
☐ JPEG	☐ CMYK	☐ Bps 32	
☐ Deflate	☐ YCbCr		
✓ PackBits	☐ CIE L*a*b*		

KOST-Val - Configuration - SIARD

Validation setting: SIARD

Versions

✓ SIARD-1.0 (eCH-0165 v1)

✓ SIARD-2.1

✓ SIARD-2.2

KOST-Val - Configuration - SIP

Validation setting: SIP
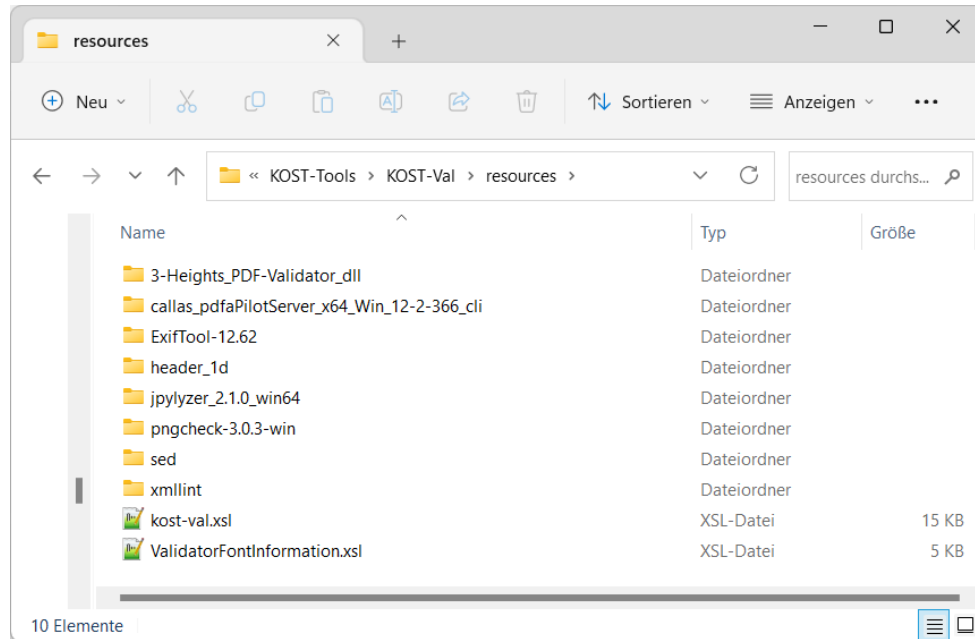
Path length	179
SIP name	SIP_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]_\w{3}

☐ Only warning for old documents (Entstehungszeitraum)

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	Start KOST-Val by clicking on "KOST-Val" in the "KOST-Tools" start menu.
2	1. Specify / select file or folder for validation 2. Adjust configuration and LogType if necessary 3. Start validation
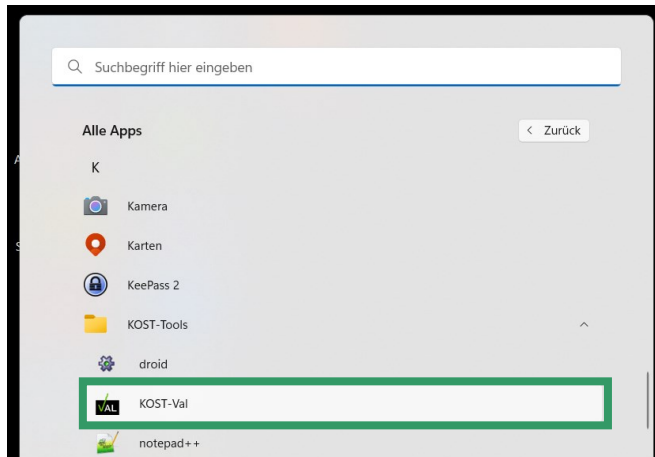
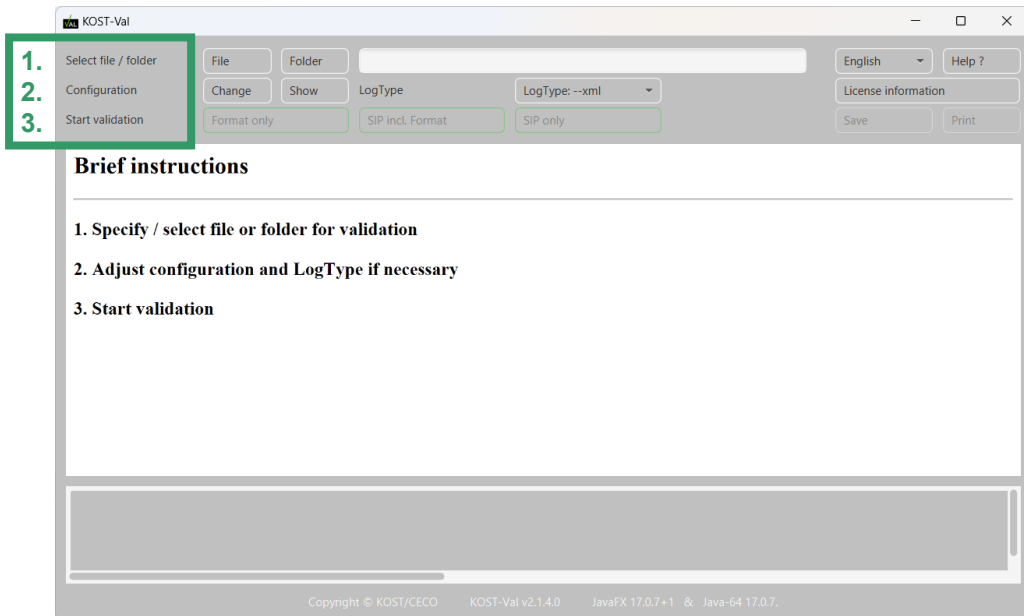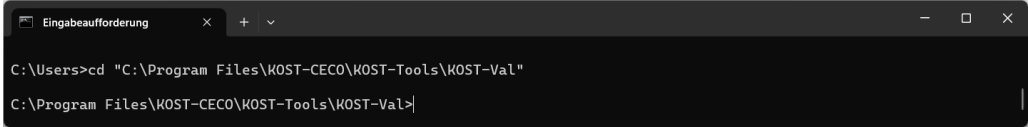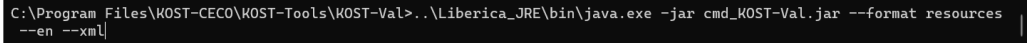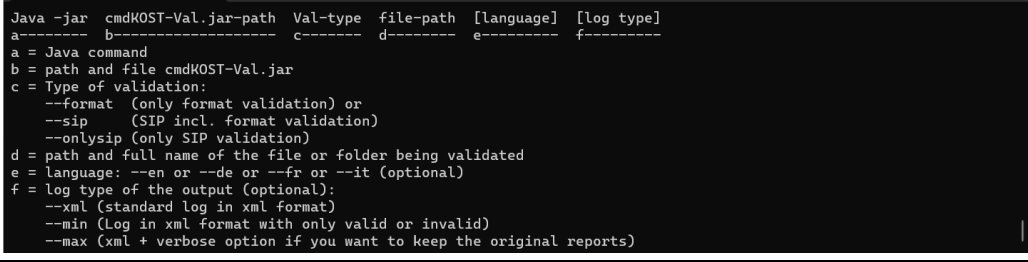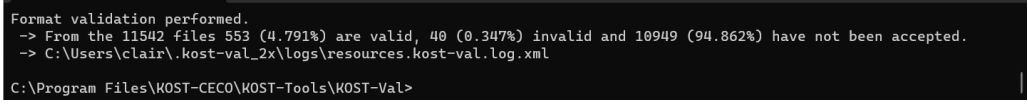|		
	4.	Wait until the validation is finished. The progress / current file is shown in the field below. **4. Format validation is performed.** **Please wait ...** 2512 JPEG: C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val\resources\ExifTool-12.62\t\images\MWG.jpg = Valid 2508 JPEG: C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val\resources\ExifTool-12.62\t\images\Minolta.jpg = Valid 2506 JPEG: C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val\resources\ExifTool-12.62\t\images\Motorola.jpg = Valid 2505 JPEG: C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val\resources\ExifTool-12.62\t\images\Nikon.jpg = Valid
3		At the end of the validation the log file is displayed. This log file contains additional detailed information about the individual invalid validation steps, in particular the affected validation step and the corresponding error. If desired, the KOST-Val-Log file can be saved or printed. **KOST-Val** **Validation: PDF/A, JP2, JPEG, TIFF, PNG, SIARD** Recognition: TXT, FLAC, WAVE, MP3, MKV, MP4, XML, JSON, CSV, XLSX, HTML, WARC, DWG, **From the 11542 files 553 (4.791%) are valid, 40 (0.347%) invalid and 10949 (94.862%) have not been accepted.** **Invalid:** **Validation: TIFF -> C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val\resources\ExifTool-12.62\t\images\GeoTiff.tif** B) Jhove The Jhove validation failed. Status: Well-Formed, but not valid B) Jhove - Jhove ErrorMessage: Projection value out of range: 16017 Format validation performed. -> From the 11542 files 553 (4.791%) are valid, 40 (0.347%) invalid and 10949 (94.862%) have not been accepted. -> C:\Users\clair\.kost-val_2x\logs\resources.kost-val1.log.xml

8.2 Start the validation manually

1	Open a command prompt and change to the desired working directory (`cd "C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val"`)[9].
	`C:\Users>cd "C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val"` `C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val>`

2	Invoke the KOST-Val command (separate command options with spaces).
	`..\Liberica_JRE\bin\java.exe -jar cmd_KOST-Val.jar --format resources --en --xml`
	
	`C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val>..\Liberica_JRE\bin\java.exe -jar cmd_KOST-Val.jar --format resources --en --xml`
	
	Notes:
	Invoking `java -jar` is possible only if the desired Java Runtime Environment (JRE) is the standard version.
	If required, the Java Virtual Memory can be quickly adapted. -Xmx should be adjusted in "Out of Memory" and -Xss at "Stack Overflow" errors (`java -Xmx1024m -Xss128m -jar`).
	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.
	Building KOST-Val command:
	<pre>Java -jar cmdKOST-Val.jar-path Val-type file-path [language] [log type]
	a------- b------------------ c------- d-------- e--------- f---------
	a = Java command
	b = path and file cmdKOST-Val.jar
	c = Type of validation:
	    --format (only format validation) or
	    --sip (SIP incl. format validation)
	    --onlysip (only SIP validation)
	d = path and full name of the file or folder being validated
	e = language: --en or --de or --fr or --it (optional)
	f = log type of the output (optional):
	    --xml (standard log in xml format)
	    --min (Log in xml format with only valid or invalid)
	    --max (xml + verbose option if you want to keep the original reports)</pre>

3	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-Tools\KOST-Val>`) is displayed.
	
	<pre>Format validation performed.
	 -> From the 11542 files 553 (4.791%) are valid, 40 (0.347%) invalid and 10949 (94.862%) have not been accepted.
	 -> C:\Users\clair\.kost-val_2x\logs\resources.kost-val.log.xml
	
	C:\Program Files\KOST-CECO\KOST-Tools\KOST-Val></pre>
	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] To change the drive type, e.g., `c:`.

9 Copyright

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	6.26.0.5	see Chapter 9.1
Apache Commons http://commons.apache.org/ - commons-logging-1.2.jar - commons-io-2.13.jar	 1.2 2.13	Apache License 2.0
Apache Xerces2 http://xerces.apache.org/	2.12.2	Apache License 2.0
BadPeggy http://coderslagoon.com/	2.0	GPL v3 License
Jdom http://www.jdom.org/	2.0.6.1	jdom License
Jhove https://jhove.openpreservation.org/	1.28	LGPL v2.1 License
Spring Framework API https://spring.io/projects/spring-framework	5.3.19	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/	12.62	PERL respective GPL v3.0 License
Jpylyzer http://jpylyzer.openpreservation.org/	2.1.0	LGPL v3.0 License
pdfaPilot CLI https://www.callassoftware.com	12.2.366	see Chapter 9.2
pngcheck http://libpng.org/pub/png/apps/pngcheck	3.0.3	GPL v2 License
GNU sed https://www.gnu.org/software/sed	4.4	GPL v3+ License
Xmllint https://xmllint.com/	20630	MIT License

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

9.1 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. The activation of this license is done with the "LicenseManager.exe", which already exists in "KOST-Val\resources\3-Heights_PDF-Validator_dll".

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

9.2   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.

10 Annex

10.1 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)	JBIG2 check
K	(configurable)	Font validation

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

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

PNG validation: KOST-Val reads a PNG and uses pngcheck to validate the structure and content:

Validation step		Description
A	(exit on error)	Recognition and pngcheck
B		Structure
C		Colors
D		Image data
E		Metadata
F		Others

XML validation: KOST-Val reads a XML and uses xmllint to validate the structure and content:

Validation step		Description
A	(exit on error)	Recognition and xmllint
B	(exit on error)	XML structure
C		XML validation with XSD

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
1g		Package size and number of files
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.

10.2 Functional Principle of Format Validation