



Guide d'application

Table des matières

1	Préambule	2
2	Configurations système requises	3
3	Problèmes en suspens / Commentaires / Souhaits de modifications.....	3
4	Introduction à la double validation PDF/A.....	4
5	Installation de KOST-Val	5
6	Configuration de KOST-Val.....	5
6.1	Composants du fichier de configuration « kostval.conf.xml »	5
6.2	« KaD_SignatureFile_V72.xml » et « jhove.conf »	7
7	Ressources de KOST-Val	7
8	Exécuter la validation	8
8.1	KOST-Val avec l'interface graphique (GUI)	8
8.2	Démarrer la validation manuellement	10
9	Description des étapes de validation.....	12
9.1	TIFF	12
9.2	SIARD	13
9.3	PDF/A	14
9.1	JP2.....	15
9.2	JPEG.....	15
9.3	SIP	16
10	Droits d'auteur.....	17
10.1	Licence « 3-Heights™ PDF/A Validator API »	18
10.2	Licence « pdfaPilot CLI ».....	19
11	Annexe.....	20
11.1	Structure du programme.....	20
11.2	Principe de la validation de format.....	22

1 Préambule

KOST-Val est une application java pour la validation de la configuration et du contenu des fichiers TIFF (*Tagged Image File Format*), SIARD (*Software Independent Archiving of Relational Databases*), PDF/A, JP2 (JPEG 2000 ; *Joint Photographic Experts Group 2000*) JPEG ainsi que d'un paquet d'informations à verser (SIP ; *Submission Information Package*) pour le versement d'informations numériques. Cette application est sous licence GPL3+ et elle est mise à disposition du public par le CECO en Open Source. KOST-Val se base sur les composants inchangés d'autres fabricants qui sont intégrés directement dans le code source de KOST-Val. Les utilisateurs de KOST-Val sont tenus de suivre les dispositions de licence de tous ces composants. Des informations supplémentaires sont disponibles au chapitre 10.

KOST-Val remplit les exigences décrites ci-après.

Validation TIFF : KOST-Val lit un fichier TIFF et valide avec l'aide de Jhove la structure et le contenu ainsi que, avec l'aide de ExifTool, les caractéristiques clés telles que compression, espace colorimétrique et multipage. Les caractéristiques peuvent être configurées.

Validation SIARD : KOST-Val lit un fichier SIARD (eCH-0165¹ v1 et SIARD v2.1²) et valide la structure et le contenu.

Validation PDF/A : KOST-Val lit un fichier PDF ou PDF/A (ISO 19005-1 et 19005-2) et valide avec l'aide du « 3-Heights™ PDF/A Validator » de PDF-Tools ou avec l'aide du « pdfaPilot » de callas la structure et le contenu du fichier PDF. KOST-Val répartit les divers messages d'erreur dans les principaux groupes tels que polices, graphique et métadonnées. KOST-Val fournit seulement des versions limitée³ de « 3-Heights™ PDF/A Validator » de PDF-Tools et « pdfaPilot » de callas.

Dans le module J il peut être défini si la compression JBIG2⁴ est acceptée ou non. Le module K contrôle, selon la configuration, si les fonts peuvent être recherchées et extraites ou non.

Validation JP2 : KOST-Val lit un fichier JP2 (ISO 15444) et valide avec l'aide de Jpylyzer la structure et le contenu.

Validation JPEG : KOST-Val lit un fichier JPEG (ISO 10918-1) et valide avec l'aide de BadPeggy la structure et le contenu⁵.

¹ La spécification peut être téléchargée à partir du site eCH : <https://www.ech.ch/ech/ech-0165>.

² La spécification peut être téléchargée à partir du site CECO : https://kost-ceco.ch/cms/index.php?siard_de.

³ La restriction est principalement liée à la capacité maximale de 72'000 pages par an. Plus d'informations sur les licences sont dans chapitre 10. Cette restriction peut être enlevée en acquérant une licence du « 3-Heights™ PDF/A Validator API » de PDF-Tools et en l'activant avec le LicenseManager (resources\3-Heights_PDF-Validator_dll) respectivement par acquérant « pdfaPilot » de callas.

⁴ Au sujet du problème de la compression JBIG2 voir http://kost-ceco.ch/cms/index.php?jbig2-compression_fr. Le Groupe d'experts du preservation planning PPEG recommande pour le moment de ne pas utiliser la compression JBIG2 lors de la création de fichiers PDF.

⁵ Le message d'erreur « Not a JPEG file » est raffiné par KOST-Val et sortie différencié.

Validation SIP : KOST-Val lit un SIP (eCH-0160⁶ v1 et v1.1) et valide les points suivants de la spécification SIP. Les différents points sont validés dans des groupes tels que la structure des données, la validation du schéma et validation de la somme de contrôle. Auparavant une validation du format est effectuée.

Les résultats (y compris les messages d'incohérences ou d'erreurs) de chaque étape sont écrits dans un fichier de log.

Les étapes de validation / les vérifications sont exécutées les unes après les autres. Lorsque cela est possible, la validation continue en cas d'erreur afin de réduire le nombre de cycles de correction.

Des informations plus détaillées sur les formats individuels et des étapes de validation sont présentées dans l'annexe.

2 Configurations système requises

- Microsoft Windows 98 ou plus récente
- Au moins 128 MB de RAM (512 MB ou plus recommandé)
- Au moins 20 GB d'espace sur le disque dur
- Java Runtime Environment (JRE), version 8 respectivement 1.8 incl. JavaFX ou-Liberica Full openJDK⁷

3 Problèmes en suspens / Commentaires / Souhaits de modifications

Les points ouverts (erreurs, bogues, souhaits de modifications) sont visibles sur la plate-forme de développement GitHub (<https://github.com/KOST-CECO/KOST-Val/issues>) et peuvent être rapportés à kost-val@kost-ceco.ch.

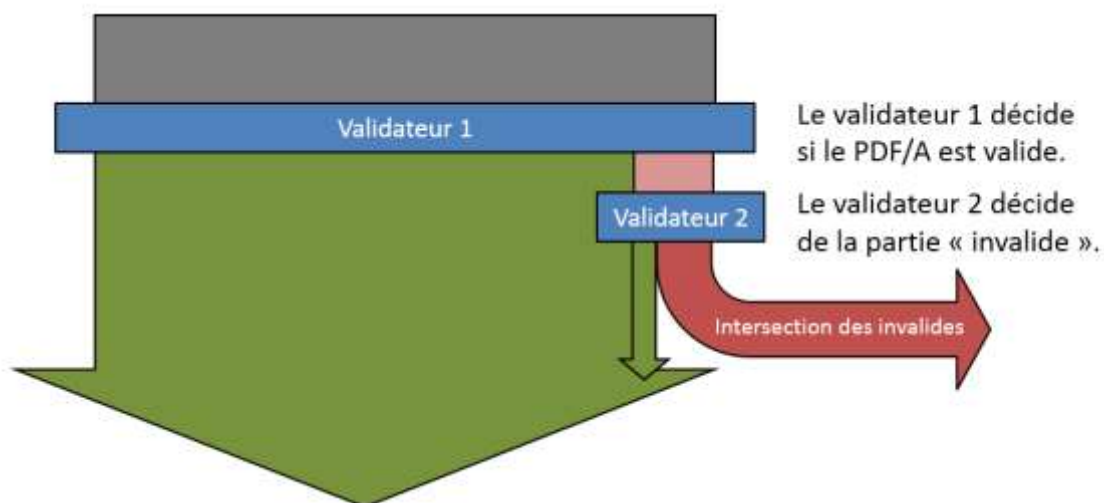
Cette liste peut être complétée par tous et est traitée par l'équipe de développement.

⁶ La spécification peut être téléchargée à partir du site eCH : <https://www.ech.ch/eCH/eCH-0160>.

⁷ La version suivante a été testée avec succès: <https://download.bell-sw.com/java/14.0.2+13/bellsoft-jdk14.0.2+13-windows-amd64-full.msi>

4 Introduction à la double validation PDF/A

Pour le PDF/A, le CECO propose la possibilité d'une double validation. Dans ce processus, un fichier PDF/A est d'abord contrôlé par une première validation. En cas de résultat invalide, un contrôle au moyen d'un deuxième validateur est effectué. Le PDF/A est considéré comme valide si au moins un des validateurs l'a identifié comme valide et comme invalide si les deux validateurs l'ont identifié comme invalide.⁸



La double validation PDF/A ne peut être utilisée que si les archives autorisent la prise en charge de fichiers PDF/A potentiellement invalides. Si ce n'est pas le cas, il ne faut pas faire la double validation.

Pour la double validation, « 3-Heights™ PDF/A Validator » de PDF-Tools et « pdfaPilot » de callas sont utilisés. Si seulement un validateur est activé, il effectuera automatiquement une validation simple.

Le constat que même des validateurs PDF/A de qualité peuvent donner des résultats différents est à la base de la conception de la double validation. Cela tient d'une part au fait que la norme PDF/A originelle implique un ensemble d'autres normes et standards qui ne sont pas implémentés jusque dans les derniers détails de manière obligatoire. D'autre part, certains objectifs de la norme sont formulés de telle sorte qu'ils peuvent légitimement être implantés de différentes manières. Pouvoir implémenter de manière homogène et exhaustive tous les outils importants de la spécification est de la musique d'avenir. C'est pourquoi la double validation de KOST-Val offre une solution provisoire.

⁸ C'est pourquoi la double validation ne peut être effectuée qu'avec des validateurs PDF/A de qualité. Les versions les plus récentes de « 3-Heights™ PDF/A Validator » de PDF Tools et de « pdfaPilot » de callas répondent à ces exigences élevées.

5 Installation de KOST-Val

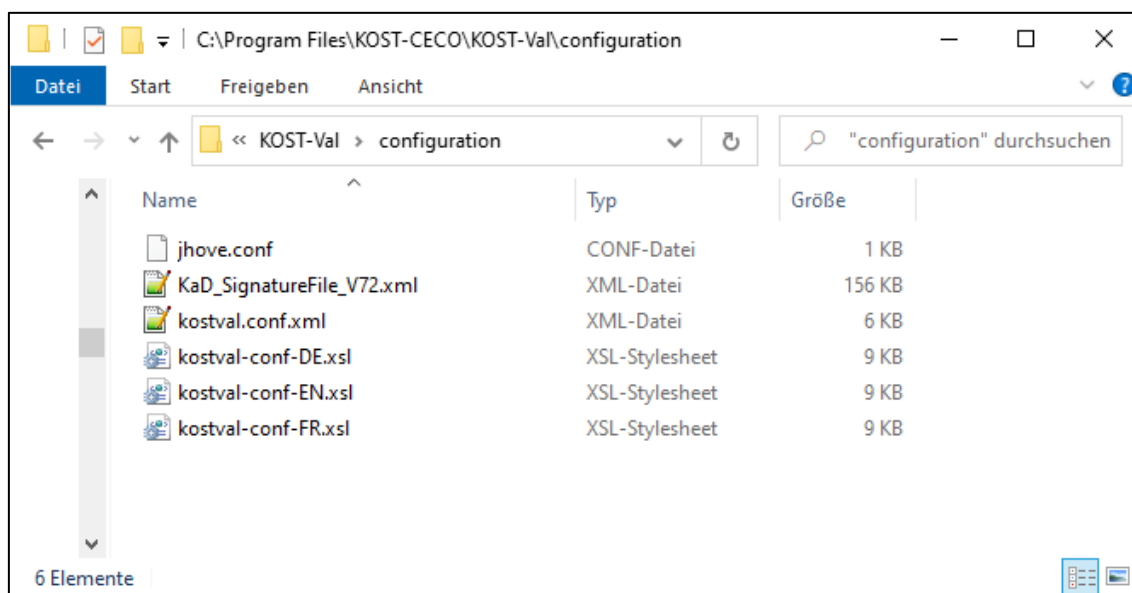
1

Télécharger KOST-Val et l'extraire dans le répertoire souhaité.



Depuis la version 1.9.1 de KOST-Val, les fichiers dll en 64 bits sont inclus dans le répertoire KOST-Val. Si seul un JRE8 32 bits est disponible, les fichiers dll doivent être échangés avec les fichiers du répertoire "resources\3-Heights_PDF-Validator_dll\Win32". Selon l'autorisation, cela est fait directement par KOST-Val, sinon la dll doit être échangée manuellement.

6 Configuration de KOST-Val



Le fichier « KaD_SignatureFile_V72.xml » et le fichier « jhove.conf » sont enregistrés dans le dossier « configuration ». Ils sont décrits dans le chapitre 0.

Le fichier de configuration « kostval.conf.xml » ainsi que les trois stylesheets sont copiés dans le répertoire « USERHOME/.kost-val_2x/configuration » s'ils ne sont pas correctement disponibles. Toutes les configurations du KOST-Val peuvent être effectuées via l'interface graphique (GUI).

6.1 Composants du fichier de configuration « kostval.conf.xml »

Le fichier de configuration « kostval.conf.xml » est divisé en différentes parties. La configuration délivrée permet la validation immédiate de PDF/A, TIFF, SIARD, JP2, JPEG et SIP. Ci-après une brève description des composants.

KOST-Val - Configuration

☒ JPEG

☒ JPEG2000

☒ SIARD
 ☒ SIARD-1.0 (eCRi-0165 v1)
 ☒ SIARD-2.1

☒ PDF/A
 ☒ PDF Tools
 ☒ Callas
 ☒ PDF/A-1a
 ☒ PDF/A-2a
 ☒ Font
 ☐ Image
 ☐ JBIG2
 ☒ details
 ☐ N-Entry
 ☒ PDF/A-1b
 ☒ PDF/A-2b
 ☒ Tolerant
 ☒ PDF/A-2u

☒ TIFF

Algorithme de compression
 Espace couleur
 Bits par échantillon (par canal)
 Divers

☒ Uncompressed
☒ LZW
☒ WhiteIsZero
☐ transparency
☒ 8ps 1
☒ 8ps 8
☒ Multipage
☒ CCITT 1D
☐ JPEG
☒ BlackIsZero
☐ CMYK
☐ 8ps 2
☒ 8ps 16
☐ Tiles
☒ T4/Group 3 Fax
☐ Deflate
☒ RGB
☐ YCbCr
☒ 8ps 4
☐ 8ps 32
☐ Size
☒ T6/Group 4 Fax
☒ PackBits
☒ RGB Palette
☐ CIE L*a*b*

☒ eCRi-0160
 Longueur du chemin: 179
 Nom du SIP: SIP_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]_w{3}
 PUID: TXT PDFa1 PDFa2 TIFF JP2 JPEG WAVE MP3 MP4 MJ2 MPEG2 CSV SIARD WARC

Arbeitsverzeichnis

Inputverzeichnis

KOST-Val Configuration

Précise si une validation JPEG doit avoir lieu [yes]:	yes
Précise si une validation JP2 doit avoir lieu [yes]:	yes
Précise si une validation SIARD doit avoir lieu [yes]:	yes
Précise quelles versions SIARD sont autorisées [1.0, 2.1]:	1.0 2.1
Informations sur la validation PDF/A [yes]:	yes
Précise si une validation PDF/A avec les outils PDF doit avoir lieu [yes]:	yes
Précise si les PDF Tools doivent également produire des erreurs détaillées en anglais [yes]:	yes
Précise si une validation PDF/A avec callas doit avoir lieu [yes]:	yes
Précise si callas doit émettre une erreur (E) ou un avertissement (W) si l'entrée N ne correspond pas [W]:	W
Précise quelles versions PDF/A sont autorisées [1A, 1B, 2A, 2B, 2U]:	1A 1B 2A 2B 2U
Précise si une validation de la police (recherche et extractibilité) doit avoir lieu [tolerant]:	tolerant
Précise si la validation des images (JPEG et JP2) doit avoir lieu [no]:	no
Précise si la compression JBIG2 est autorisée [yes]:	yes
Précise si une validation TIFF doit avoir lieu [yes]:	yes
Spécifie les algorithmes de compression autorisés [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
ASpécification des espaces de couleur autorisés [WhiteIsZero, BlackIsZero, RGB, RGB Palette]:	WhiteIsZero BlackIsZero RGB RGB Palette
Précise le nombre de bits par sample [1, 4, 8, 16]:	1 4 8 16
Précise si les TIFF multipages sont autorisés ou non [yes]:	yes
Précise si la construction en carreaux est autorisée ou non [no]:	no
Précise si les fichiers de 1000 Mo (~1 Go) et plus sont autorisés ou non [no]:	no
Détails de la validation SIP [yes]:	yes
Nombre maximal de caractères autorisés dans les chemins d'accès [179]:	179
Spécifications pour la structure du nom SIP [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}
Liste des formats de fichiers autorisés [TXT, PDFa1, PDFa2, TIFF, JP2, JPEG, WAVE, MP3, MP4, MJ2, CSV, SIARD, WARC]:	TXT PDFa1 PDFa2 TIFF JP2 JPEG WAVE MP3 MP4 MJ2 MPEG2 CSV SIARD WARC
Répertoire de travail []:	
Répertoire d'entrée []:	

6.2 « KaD_SignatureFile_V72.xml » et « jhove.conf »

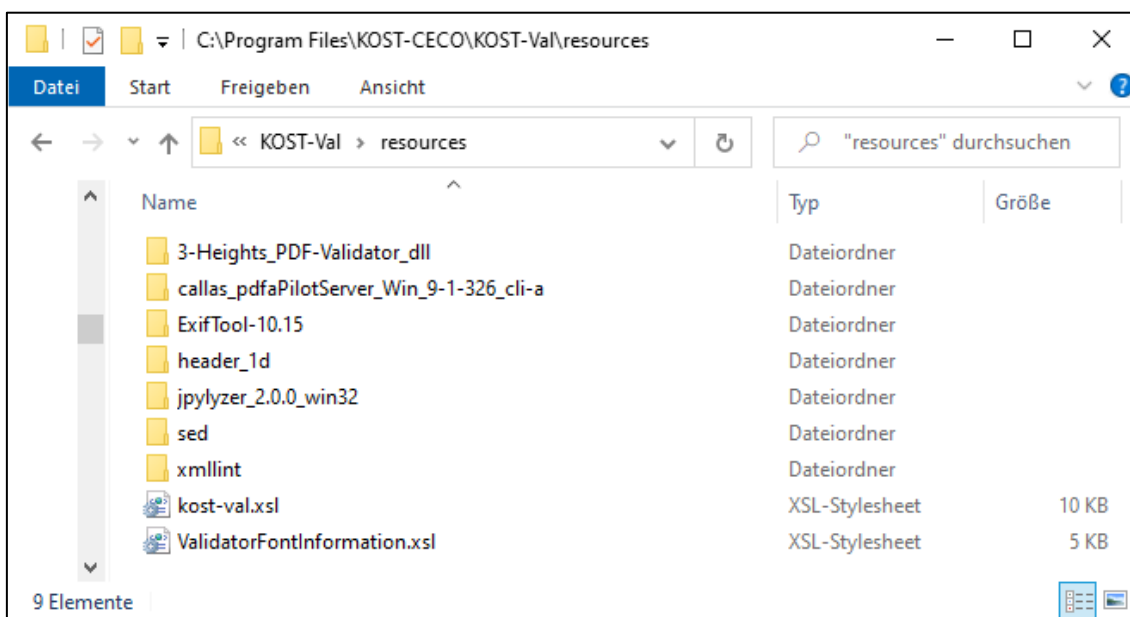
Le fichier « KaD_SignatureFile_V72.xml » et le fichier « jhove.conf » sont aussi enregistrés dans le dossier « configuration », qui n'a pas besoin d'être ajusté.

« KaD_SignatureFile_V72.xml » est nécessaire pour la reconnaissance du format. Il comprend l'état du DROID au 28.08.2012 et a été en plus adapté par le CECO dans le domaine de la reconnaissance des formats de données d'archivage⁹.

« jhove.conf » est requis pour la validation de Jhove intégré dans KOST-Val.

7 Ressources de KOST-Val

Toutes les ressources de KOST-Val sont stockées dans le sous-dossier « ressources ».



⁹ A la place du fichier de signature de DROID, KOST-Val utilise un fichier de signature compatible développé par le CECO (voir aussi https://github.com/KOST-CECO/KaD_SignatureFile), KaD-SignatureFile. Ce dernier ne comprend que les formats analysés dans le Cfa et permet de les reconnaître avec la granularité recommandée par le CECO. Il aboutit ainsi à un gain significatif d'efficacité et à une facilité d'utilisation augmentée par rapport au fichier de signature de DROID.

8 Exécuter la validation

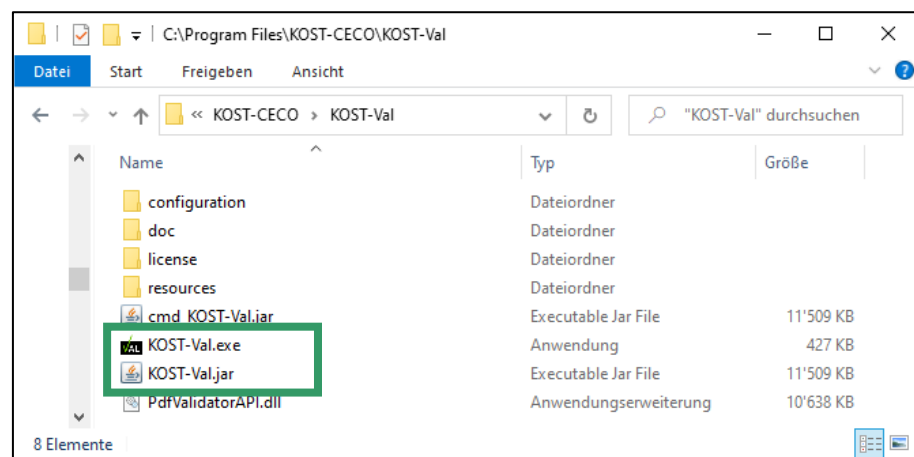


KOST-Val n'est pas thread-safe!

Cela signifie que KOST-Val ne peut pas être exécuté plusieurs fois simultanément sans interférences. L'exécution simultanée de plusieurs instances de KOST-Val peut entraîner des erreurs comme une copie de travail manquante.

8.1 KOST-Val avec l'interface graphique (GUI)

- 1** Lancer KOST-Val en double-cliquant sur « KOST-Val.exe »¹⁰ dans le dossier « KOST-Val ».



- 2**
1. Spécifier / sélectionner le fichier ou le dossier à valider
 2. Ajuster la configuration et le LogType si nécessaire
 3. Commencer la validation

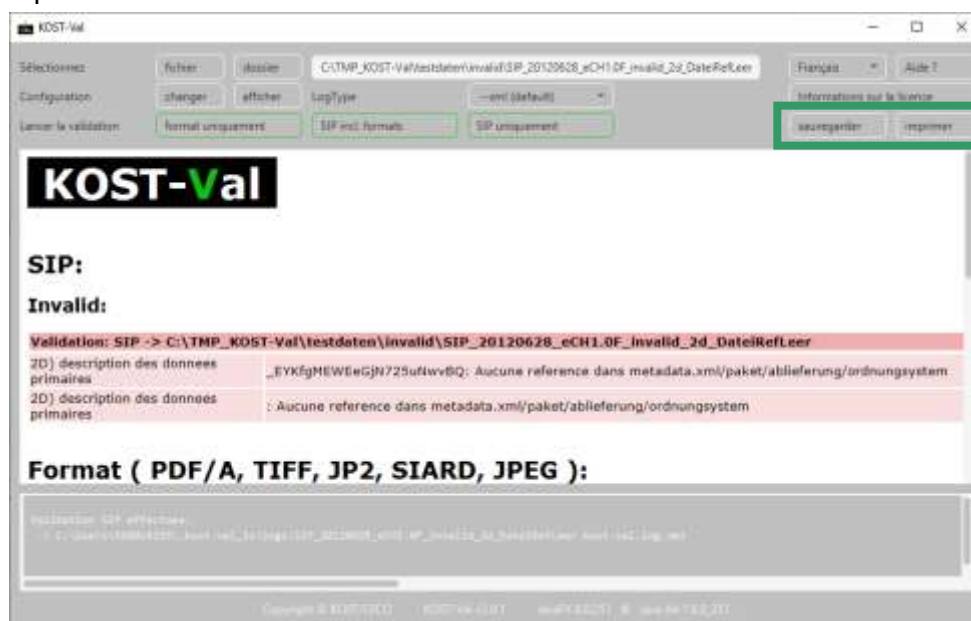


¹⁰ Si openJDK⁷ est utilisé, l'interface graphique doit être lancée via "KOST-Val.jar" et "Ouvrir avec...". Le java.exe de openJDK doit être indiqué.

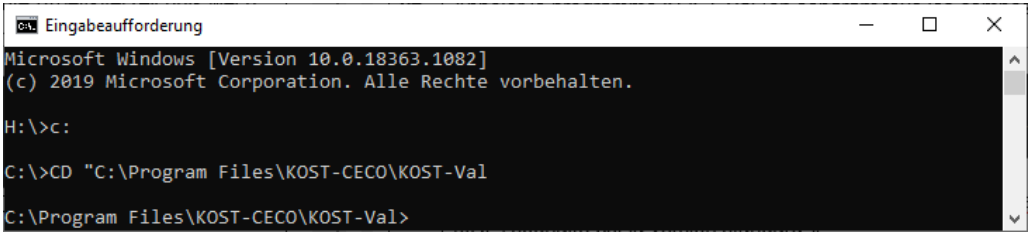
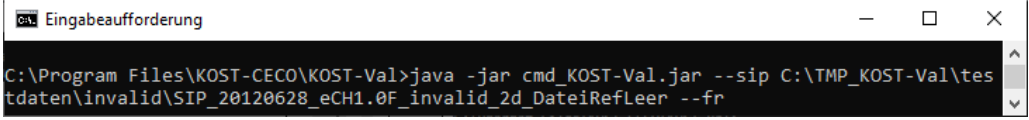
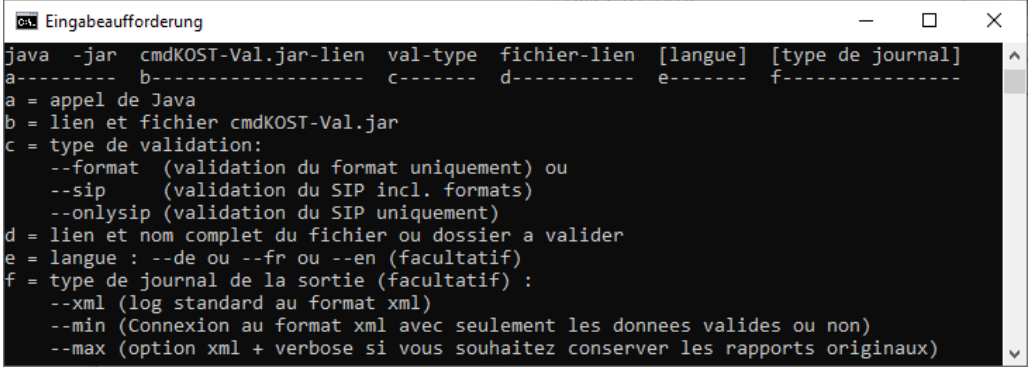
4. Attendre que la validation soit terminée.
Le dossier en cours est indiqué dans le champ inférieur.



- 3 A la fin de la validation, le fichier journal est affiché. Il contient des informations détaillées supplémentaires sur les différentes étapes de validation non valables, en particulier l'étape de validation concernée et l'erreur correspondante. Si vous le souhaitez, le fichier journal de KOST Val peut être enregistré ou imprimé.



8.2 Démarrer la validation manuellement

1	<p>Ouvrir la fenêtre d'exécution et aller dans le répertoire de travail (CD "C:\Program Files\KOST-CECO\KOST-Val")¹¹.</p> 
2	<p>Appeler le programme KOST-Val (en séparant tous les composants d'entrée par des espaces).</p> <pre>java -jar cmd_KOST-Val.jar --sip C:\TMP_KOST-Val\testdaten\invalid\SIP_20120628_eCH1.0F_invalid_2d_DateiRefLeer --fr</pre>  <p>Notes :</p> <p>L'entrée <code>java -jar</code> est seulement possible si le Java Runtime Environment (JRE) souhaité est la version standard.</p> <p>Si nécessaire, la valeur de la mémoire virtuelle de Java peut être rapidement adaptée. <code>-Xmx</code> doit être adaptée sur « Out of Memory » et <code>-Xss</code> à des erreurs « Stack Overflow » (<code>java -Xmx1024m -Xss128m -jar</code>).</p> <p>Quand un élément d'entrée contient des espaces, celui-ci doit être inscrit entre guillemets.</p> <p>Il est également possible d'appeler KOST-Val depuis tout autre emplacement. Cependant, cela nécessite l'utilisation de liens absolus.</p> <p>Commande pour KOST-Val:</p> 

¹¹ Le disque est par exemple changé avec `c:.`

3

La validation du fichier est terminée dès que « Valid » ou « Invalid » est affiché dans la fenêtre cmd. Le dossier a été validé une fois que le prompt (C:\Program Files\KOST-CECO\KOST-Val>) apparaît.

```

Eingabeaufforderung
17 PDF: C:\TMP_KOST-Val\testdaten\invalid\SIP_20120628_eCH1.0F_invalid_2d_DateiRefLeer\content\d0003\d0004\p0007.pdf = Valid

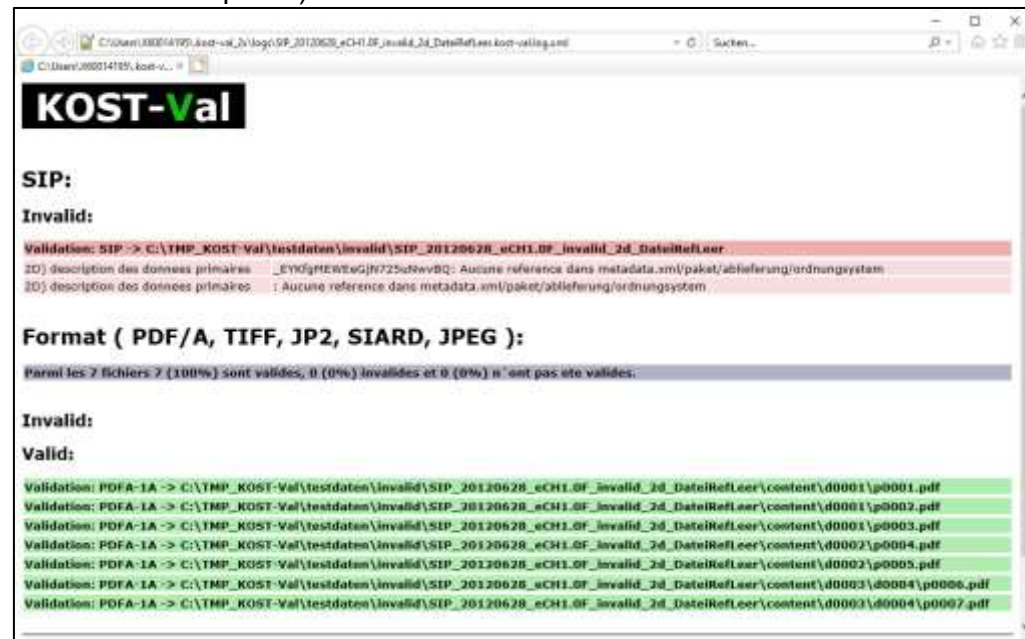
SIP: C:\TMP_KOST-Val\testdaten\invalid\SIP_20120628_eCH1.0F_invalid_2d_DateiRefLeer
Invalid

Validation SIP effectuée.
-> C:\Users\X60014195\.kost-val_2x\logs\SIP_20120628_eCH1.0F_invalid_2d_DateiRefLeer.kost-val.log.xml

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

```

Les résultats détaillés se trouvent dans le fichier « kost-val.log.xml » (ouvrir avec Internet Explorer).



Le résultat de la validation globale (fichier correct/erroné) est aussi délivré et visible dans le statut *exit* du programme, afin que la validation puisse être liée dans la chaîne de traitement automatisée. Le statut *exit* peut prendre les valeurs suivantes :

- 0 tout est OK, valide
- 1 erreur dans l'appel du programme
- 2 échec de la validation, invalide

9 Description des étapes de validation

9.1 TIFF

Chaque étape de validation TIFF est décrite ci-après de façon détaillée.

A détection

Le fichier TIFF est-il reconnu comme tel ?

B Jhove

La validation Jhove est-elle réussie ?

C compression

La compression utilisée est-elle autorisée selon le fichier de configuration ?

D espace colorimétrique

L'espace colorimétrique utilisé est-il autorisé selon le fichier de configuration ?
N'existe il pas du « Planar Configuration »?

E BitsPerSample

Les BitsPerSample utilisés sont-ils autorisés selon le fichier de configuration ?

F Multipage

L'utilisation de la propriété « multipage » est-elle autorisée selon le fichier de configuration ?

G tuiles

L'utilisation de la propriété « tuiles » est-elle autorisée selon le fichier de configuration ?

H grandeur

La taille du fichier est-elle autorisée selon le fichier de configuration ?

9.2 SIARD

Ce chapitre décrit plus en détail les étapes de validation. Ces critères sont un extrait des critères obligatoires de la spécification SIARD.

A lisibilité

Le fichier SIARD peut-il être lu ?

B validation de la structure des données

La structure primaire des données est-elle correcte ?

C validation du header

Le dossier header est-il valide ?

D validation de la structure

La structure de metadata.xml (folder) correspond-elle à la structure des fichiers de content ?

E validation des colonnes

Les données du fichier metadata.xml ont-elles été transférées correctement dans le fichier tableZ.xsd?

F validation des lignes

Les données du fichier metadata.xml ont-elles été transférées correctement dans le fichier tableZ.xsd?

G validation des tables

Les noms de colonnes sont-ils uniques dans la table ?

H validation du contenu

Les fichiers XML dans le contenu sont-ils valides par rapport à leur définition de schéma (XSD) ?

I détection du format SIARD

Le fichier SIARD est-il détecté comme SIARD ?

J données primaires supplémentaires

Tous les fichiers ou dossiers contenu dans le dossier « content » sont-ils décrits dans metadata.xml? Furent xyz et abc rempli et non laissé sur (...)?

W avertissement

Furent <dataOwner> et <dataOriginTimespan> rempli et non laissé sur (...)?

9.3 PDF/A

Les messages du rapport initial de callas et les messages d'erreur détaillés de PDF Tools sont attribués aux modules basés sur des mots. Les messages d'erreur (« Message ») sont pris en charge 1:1. Lors de la validation avec PDF Tools, les catégories sont attribuées aux modules, traduites, et le texte est affiché.

A général

La validation des conditions générales telles que fichier PDF/A non corrompu est-elle réussie ?

B structure

La validation de la structure (par exemple BOF et EOF) est-elle réussie ?

C graphique

La validation graphique (par exemple, espace de couleur valide) est-elle réussie ?

Le nombre de composants dans l'entrée N du PDF/A OutputIntent correspond-il au profil ICC ?¹²

D polices

Les polices sont-elles intégrées correctement dans le fichier PDF/A (par exemple, les caractères utilisés sont-ils tous intégrés) ?

E transparence

Les exigences en matière de transparence sont-elles remplies pour le fichier PDF/A ?

F annotations

Le fichier PDF/A ne contient-il que des annotations licites (p.ex. pas d'annotation 3D) ?

G actions & interactions

Le fichier PDF/A ne contient-il que des actions licites (p.ex. pas de JavaScript) ?

Le fichier PDF/A ne contient-il que des interactions licites ?

H métadonnées

Les métadonnées sont-elles correctement incluses dans le fichier PDF/A ?

I disponibilité

Les exigences à l'accessibilité (niveau de conformité A) sont-elles remplies ?

J contrôle JBIG2 (configurable)

Le fichier PDF/A contient-il la compression JBIG2 ?

K validation des polices (configurable)

Les polices contenues sont-elles consultables et extractibles ?

¹² A cet effet, on effectue un test supplémentaire avec « pdfaPilot » de callas, qui émet un avertissement ou une erreur en fonction de la configuration (voir 6.1). Le « 3-Heights™ PDF/A Validator » de PDF Tools, effectuée cette vérification en standard, mais elle est critiquée dans la iCategory_1, qui est émise en raison de la composition sous « A) validation générale ».

9.1 JP2

Les tests échoués de Jpylyzer sont attribués aux modules. Des messages d'erreur correspondant à ces derniers sont émis. Ce chapitre décrit plus en détail les étapes de validation de JP2.

A Détection et Jpylyzer

Le fichier JP2 est-il reconnu comme tel ? La validation avec Jpylyzer a-t-elle été réussie ?

B Validation des métadonnées

Les métadonnées ont-elles été saisies correctement ?

C Validation de l'image

L'image est-elle construite correctement ?

D Validation étendue

Les tests Jpylyzer additionnels ont-ils été réussis ?

9.2 JPEG

Les messages d'erreur de BadPeggy sont attribués aux modules et traduits (original en anglais). Ce chapitre décrit plus en détail les étapes de validation de JPEG.

A Détection et BadPeggy

Le fichier JPEG est-il reconnu comme tel ? La validation avec BadPeggy a-t-elle été réussie ?

B Données corrompues

BadPeggy génère-il un message d'erreur, ce qui indique des données corrompues°?

C Structure de fichier invalide

BadPeggy génère-il un message d'erreur, ce qui indique une structure de fichier non valide°?

D Autres problèmes

BadPeggy génère-il un message d'erreur, qui n'a pas encore été attribué et traduit°?

9.3 SIP

Ce chapitre décrit plus en détail les étapes de validation. Ces critères sont un extrait des critères obligatoires de la spécification SIP.

Module 1 : Consistance paquet et XML

1A lisibilité

Le SIP (ZIP / ZIP64) peut-il être ouvert sans erreur ?

1B validation de la structure des données

Existe-t-il une structure de répertoire primaire correcte ?

1C nommage de la structure et des fichiers

Les noms correspondent-ils aux restrictions de la spécification ?

1D validation du schéma metadata.xml

Les métadonnées correspondent-elles aux fichiers schéma dans header/xsd ?

1E déterminer le type du SIP

Le type SIP est transmis et affiché : GEVER ou FILE.

1F existence des données primaires

Existe-t-il des fichiers primaires dans le répertoire content/ ?¹³

Module 2 : Consistance du fichier

2A données primaires manquantes

Tous les fichiers référencés existent-ils ?

2B données primaires supplémentaires

N'existe-t-il pas de fichiers primaires dans le SIP ?

2C validation de la somme de contrôle

Les sommes de contrôle des fichiers sont-elles correctes ?

2D description des données primaires

Tous les fichiers référencés sont-ils aussi enregistrés dans ordnungssystem ?

Module 3 : Consistance du format de fichier et de la date

3A détection des formats

Effectuer une reconnaissance et comparer si les formats sont autorisés.

3B formats supplémentaires

Lister tous les fichiers avec un format non spécifié et indiquer le format.

3C validation de formats

Les formats sont-ils valides ?

3D validation de période

Les indications temporelles dans (metadata.xml)/ablieferung sont-elles consistantes ?

¹³ Les fichiers primaires peuvent manquer dans un GEVER-SIP qui est utilisé uniquement pour l'archivage d'une structure de système de classement, mais jamais dans un FILE-SIP.

10 Droits d'auteur

KOST-Val a été développé par le CECO. Le CECO en possède tous les droits. KOST-Val a été publié par le CECO en 2012 sous la licence GNU General Public License v3+.

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

KOST-Val se base sur les composants inchangés suivants d'autres fabricants qui sont intégrés directement dans le code source de KOST-Val :

Programme tiers/Composants tiers	Version	Licence
3-Heights™ PDF/A Validator API http://www.pdf-tools.com	6.11.0.7	Voir chapitre 10.1
Apache Commons http://commons.apache.org/ - commons-logging-1.2.jar - commons-io-2.6.jar	1.2 2.6	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
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 utilise les composants inchangés suivants d'autres fabricants qui sont livrés avec KOST-Val :

Programme tiers/Composants tiers	Version	Licence
ExifTool http://www.sno.phy.queensu.ca/~phil/exiftool/	10.15	PERL respective GPL v3.0 License
Jpylyzer http://jpylyzer.openpreservation.org/	2.0.0	LGPL v3.0 License
pdfaPilot CLI https://www.callassoftware.com	9.1.326	Voir chapitre 10.2
GNU sed https://www.gnu.org/software/sed	4.4	GPL v3+ License
XmlLint http://xmlsoft.org/xmllint.html/	20630	MIT License

Les utilisateurs de KOST-Val sont tenus de suivre les dispositions de licence de tous ces composants qui se trouvent dans le répertoire KOST-Val\license.

10.1 Licence « 3-Heights™ PDF/A Validator API »

Pour l'utilisation de la version limitée de « 3-Heights™ PDF/A Validator » de PDF Tools, le CECO a convenu des conditions d'accord individuel suivantes aux Conditions Générales de licence avec 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.

Les points suivants sont déterminants pour l'utilisateur :

- La licence comprend l'utilisation du logiciel (KOST-Val) par des institutions patrimoniales composées des archives ou des bibliothèques, leurs fournisseurs et le CECO lui-même.
- La clé de licence OEM qui est fermement liée à KOST-Val ne peut pas être utilisée hors de l'application (KOST-Val).
- La licence est illimitée dans le temps, mais le débit par installation est limité (72'000 pages par an).
- Le support de niveau 1 aux usagers est effectué par le CECO. Le CECO transmet à PDF Tools AG les cas relevant du support de niveau 2.
- Si l'utilisateur a d'autres besoins, par exemple un débit plus élevé, intégration dans d'autres applications, etc., il doit acheter le logiciel « 3-Heights™ PDF/A Validator API » directement auprès de PDF Tools AG.

Les utilisateurs de KOST-Val sont tenus de respecter les présentes dispositions de la licence.

10.2 Licence « pdfaPilot CLI »

Pour l'utilisation de la version limitée de « pdfaPilot CLI » de callas, le CECO a convenu des conditions d'accord individuel suivantes aux Conditions Générales de licence avec 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.

Les points suivants sont déterminants pour l'utilisateur :

- La licence comprend la distribution de KOST-Val aux « utilisateurs » (institutions patrimoniales, archives, bibliothèques, et leurs fournisseurs).
- La licence est illimitée dans le temps, mais le débit par installation est limité à 72'000 pages par an.
- Le CECO peut publier leur propre code de KOST-Val open source mais pdfaPilot CLI est exclue.
- Le support de niveau 1 aux usagers est effectué par le CECO. Le CECO transmet à callas software les cas relevant du support de niveau 2.

Les utilisateurs de KOST-Val sont tenus de respecter les présentes dispositions de la licence.

11 Annexe

11.1 Structure du programme

KOST-Val a été développé selon les exigences suivantes :

Exigences fonctionnelles :

Validation TIFF : KOST-Val lit un fichier TIFF et valide avec l'aide de Jhove les points suivants :

Étape de validation	Nom de l'étape
A (abandon en cas d'erreur)	détection
B	Jhove
C	compression
D	espace colorimétrique
E	BitsPerSample
F	multipage
G	tuiles
H	grandeur

Validation SIARD : KOST-Val lit un fichier SIARD et valide les points suivants :

Étape de validation	Nom de l'étape
A (abandon en cas d'erreur)	lisibilité
B	validation de la structure des données
C	validation du header
D	validation de la structure
E	validation des colonnes
F	validation des lignes
G	validation des tableaux
H	validation du contenu
I	détection du format SIARD
J	données primaires supplémentaires
W	avertissement

Validation PDF/A : KOST-Val lit un fichier PDF et valide avec l'aide du 3-Heights™ PDF/A Validator de PDF-Tools ou avec l'aide du pdfaPilot de callas les points suivants :

Étape de validation	Nom de l'étape
A (abandon en cas d'erreur)	générale
B	structure
C	graphique
D	polices
E	transparence
F	annotations & interactions
G	actions
H	métadonnées
I	disponibilité
J (configurable)	contrôle JBIG2
K (configurable)	validation des polices

Validation JP2 : KOST-Val lit un fichier JP2 et valide avec l'aide de Jpylyzer les points suivants :

Étape de validation	Nom de l'étape
A (abandon possible en cas d'erreur)	détection et Jpylyzer
B	validation des métadonnées
C	validation de l'image
D	validation étendue

Validation JPEG : KOST-Val lit un fichier JPEG et valide avec l'aide de BadPeggy les points suivants :

Étape de validation	Nom de l'étape
A (abandon possible en cas d'erreur)	détection et BadPeggy
B	données corrompues
C	structure de fichier invalide
D	autres problèmes

Validation SIP : KOST-Val lit un SIP et valide les points suivants :

Étape de validation	Nom de l'étape
1a (abandon en cas d'erreur)	lisibilité
1b (abandon en cas d'erreur)	validation de la structure des données
1c (abandon en cas d'erreur)	nommage de la structure et des fichiers
1d (abandon en cas d'erreur)	validation du schéma metadata.xml
1e	déterminer le type du SIP
1f	existence des données primaires
2a	données primaires manquantes
2b	données primaires supplémentaires
2c	validation de la somme de contrôle
2d	description des données primaires
3a	détection des formats
3b	formats supplémentaires
3c	validation de format
3d	validation de période

Les résultats (y compris les messages d'incohérences ou d'erreurs) de chaque étape sont écrits dans un fichier de log.

Le résultat de la validation globale (fichier correct/erroné) est aussi délivré et visible dans le statut *exit* du programme, afin que la validation puisse être liée dans la chaîne de traitement automatisée. Le statut *exit* peut prendre les valeurs suivantes :

- 0 tout est OK, valide
- 1 erreur dans l'appel du programme
- 2 échec de la validation, invalide

Les étapes de validation / les vérifications sont exécutées les unes après les autres. Lorsque cela est possible, la validation continue en cas d'erreur afin de réduire le nombre de cycles de correction.

Exigences non fonctionnelles :

Pour des tâches particulières, des programmes externes ou des frameworks Java appropriés sont utilisés.

L'application est constituée de manière modulaire, de sorte que la mise en place d'un ou plusieurs module(s) de validation supplémentaire(s) nécessite moins d'effort.

La sortie du programme et le fichier log permettent une lecture aisée du résultat de chaque validation et par conséquent l'utilisation de l'outil dans une chaîne de processus.

La sortie de la console se limite à la description du module de validation, les résultats valides ou invalides ainsi que le chemin vers le fichier. Toutes les informations additionnelles sont spécifiées dans le fichier log.

11.2 Principe de la validation de format

