



## Anwendungshandbuch

### Inhalt

1	Vorwort .....	2
2	Systemvoraussetzungen .....	3
3	Offene Punkte / Feedback .....	3
4	Installation.....	4
4.1	Installation von SIP-Val.....	4
4.2	Installation von Drittapplikationen .....	4
5	Konfiguration des SIP-Val.....	7
5.1	Bestandteile der Konfigurationsdatei "sipvalidator.conf.xml" .....	7
5.2	"DROID_SignatureFile_V59-kc.xml" und "SignatureFile.xsd" .....	10
5.3	"report.xml" .....	10
6	cmd_Script-Example.....	11
6.1	"_SIP-Val_Start.cmd" .....	11
6.2	"_SIP-Val_Start_+3c_+3d.cmd" .....	11
6.3	"manuelle_Eingabe_SIP-Val.cmd" .....	11
7	Validierung starten.....	12
7.1	Validierung starten mittels Script .....	12
7.2	Validierung manuell starten .....	13
8	Validierung auswerten .....	15
9	Beschreibung der Validierungsmodule und -schritte .....	17
9.1	Modul 1: Paket- und XML-Konsistenz.....	17
9.2	Modul 2: Datei-Konsistenz .....	19
9.3	Modul 3: Dateiformat- und Datums-Konsistenz .....	19
10	Urheberrecht.....	20

## 1 Vorwort

SIP-Val ist eine java-basierte Konsolenanwendung zur Validierung von Aufbau und Inhalt eines sogenannten Submission Information Package (SIP) zur Ablieferung von digitalen Informationen. Diese Anwendung steht unter der GPL3+ Lizenz und wird durch die KOST der Öffentlichkeit quelloffen zur Verfügung gestellt. SIP-Val erfüllt die im Folgenden beschriebenen Anforderungen.

### Funktionale Anforderungen:

SIP-Val liest ein SIP und validiert die folgenden Punkte aus der SIP Spezifikation des BAR von September 2009 in der Version 1.0<sup>1</sup> sowie aus der SIP Spezifikation, eCH-0160: Archivische Ablieferungsschnittstelle, von November 2012 in der Version 1.0<sup>2</sup>. Letzteres entspricht auch der SIP Spezifikation des BAR von März 2012 in der Version 4.0<sup>3</sup>:

Validierungsschritt	Bezeichnung (Stepname)
1a (Abbruch wenn Fehler)	Lesbarkeit
1b (Abbruch wenn Fehler)	primäre Verzeichnisstruktur
1c (Abbruch wenn Fehler)	Verzeichnis- und Dateinamen
1d (Abbruch wenn Fehler)	Schemavalidierung metadata.xml
1e	SIP-Typ ermitteln
1f	Primärdateien im Verzeichnis
2a	Fehlende Primärdateien
2b	Prüfsummen-Validierung
2c	Zusätzliche Primärdateien
2d	Verzeichnung GEVER-Primärdateien
3a	Formaterkennung
3b	Zusätzliche Formate
3c (Optional)	Formatvalidierung
3d (Optional)	Zeitraum-Validierung

Die Resultate (inklusive Meldungen zu Inkonsistenzen oder Fehler) werden pro Schritt ausgegeben und in eine Validierungs-Logdatei geschrieben.

Das Ergebnis der Gesamtvalidierung (korrektes/fehlerhaftes SIP) wird ebenfalls ausgegeben und im *exit*-Status des Programms sichtbar, so dass die Validierung in eine automatisierte Verarbeitungskette eingebunden werden kann. Der *exit*-Status kann die folgenden Werte annehmen:

- 0 alles OK
- 1 fehlerhafte Angaben
- 2 Validierung nicht bestanden

Die einzelnen Validierungsschritte / Prüfungen werden nacheinander ausgeführt. Wo möglich, wird die Validierung auch bei Fehlern weiter fortgesetzt, um die Anzahl von Korrekturzyklen zu reduzieren. Nur die Prüfungen 1a) bis 1d) im Modul 1 „Paket- und XML-Konsistenz“ sind Fehler, die direkt zum Programmabbruch führen.

---

<sup>1</sup> Die Spezifikation kann von der Seite "Ablieferungsschnittstelle" der BAR-Website heruntergeladen werden: <http://www.bar.admin.ch/themen/00876/00877/>.

<sup>2</sup> Die Spezifikation kann von der eCH Website heruntergeladen werden: <http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0160>.

<sup>3</sup> Die Spezifikation kann von der Seite "Ablieferungsschnittstelle" der BAR-Website heruntergeladen werden: <http://www.bar.admin.ch/themen/00876/00877/01561/>.

### Nichtfunktionale Anforderungen:

Für besondere Aufgaben (ZIP-Entpacken, XML-Validierung, Format-Erkennung) werden externe Programme oder entsprechende Java-Frameworks eingesetzt.

Die Anwendung ist modular aufgebaut, damit ohne viel Aufwand ein oder mehrere weitere Validierungsmodule eingebaut werden können.

Zur Schemavalidierung wird das im SIP hinterlegte Schema verwendet (dynamische Schemavalidierung).

Die Log-/Programmausgabe erlaubt die Verwendung des Tools in einer Prozesskette, d.h. das Ergebnis der einzelnen Validierung kann einfach ermittelt werden.

Die Konsolenausgabe begrenzt sich auf die Bezeichnung der Validierungsschritte (inklusive zugehöriges Testmodul), die Ergebnisse "valid" oder "invalid", das Gesamtergebnis sowie die Pfade zum SIP und zur generierten Log-Datei. Alle zusätzlichen Informationen werden in der Log-Datei aufgeführt.

## **2 Systemvoraussetzungen**

Microsoft Windows 98, ME, NT, 2000, XP, VISTA oder Windows 7

Mindestens 128 MB RAM (512 MB oder mehr empfohlen)

Mindestens 20 GB Festplattenspeicher

Java Runtime Environment (JRE) Version 6 oder neuer<sup>4</sup> (Download: <http://www.java.com>)

## **3 Offene Punkte / Feedback**

Die offenen Punkte, von Bugs respektive Fehler bis hin zu Ergänzungswünschen und Fragen sind auf der Entwicklungsplattform GitHub unter Issues ersichtlich:

<https://github.com/KOST-CECO/SIP-Val/issues>

Diese Liste kann und soll durch jedermann erweitert werden und wird durch das Entwicklerteam bearbeitet.

---

<sup>4</sup> Die aktuell installierte Version lässt sich überprüfen mit der Eingabe von `java -version` im cmd-Fenster.

## 4 Installation

### 4.1 Installation von SIP-Val

1	SIP-Val-Verzeichnis erstellen.
2	SIP-Val herunterladen und in das SIP-Val-Verzeichnis entpacken.

The first screenshot shows a Windows Explorer window with the address bar set to 'Computer > System (C:) > Software > KOST-Val'. The file list shows a new folder named 'SIP-Val' of type 'Dateiordner'. The second screenshot shows the same path, but now the 'SIP-Val' folder is expanded, revealing its contents: 'cmd\_Script-Example', 'configuration', 'doc', 'license', 'fr\_sipvalidator.jar', and 'sipvalidator.jar'. A green box highlights the 'SIP-Val.zip' file on the left, with an arrow pointing to the extracted files in the Explorer window.

### 4.2 Installation von Drittapplikationen

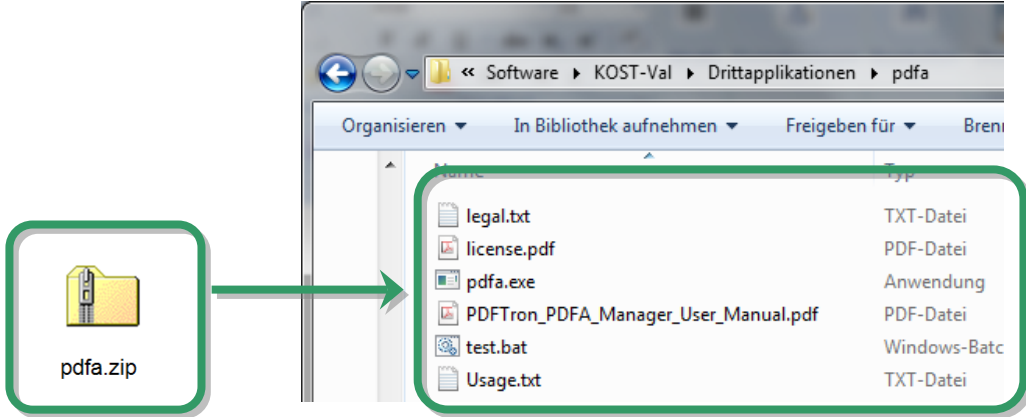
#### 4.2.1 Installation von SIARD-Val

1	Command-line-Variante von SIARD-Val (ZIP-File) unter <a href="http://kost-ceco.ch/cms/index.php?siard_val_de">http://kost-ceco.ch/cms/index.php?siard_val_de</a> herunterladen.
2	SIARD-Val in ein beliebiges Verzeichnis entpacken oder in den dafür vorgesehenen Ordner SIARD-Val.

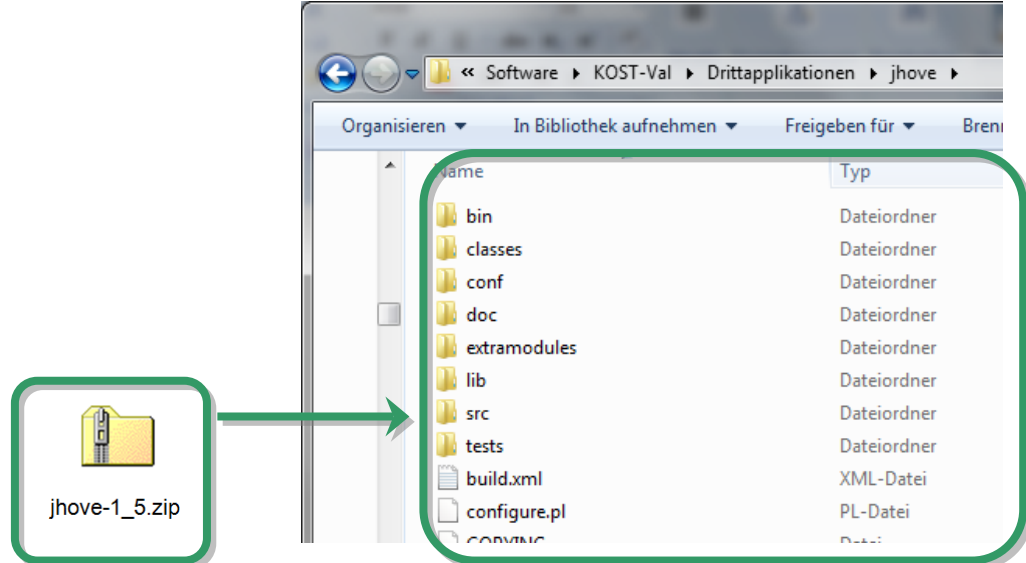
The screenshot shows a Windows Explorer window with the address bar set to 'Computer > System (C:) > Software > KOST-Val > SIARD-Val'. The file list shows the contents of the SIARD-Val folder: 'cmd\_Script-Example', 'configuration', 'doc', 'license', 'fr\_siard-val.jar', and 'siard-val.jar'. A green box highlights the 'SIARD-Val.zip' file on the left, with an arrow pointing to the extracted files in the Explorer window. The file types are listed as 'Dateiordner' for folders and 'Executable Jar File' for the .jar files.

**Hinweis:**  
Sollte ein anderes Verzeichnis gewählt werden, muss entsprechend die Konfigurationsdatei "sipvalidator.conf.xml" angepasst werden. Siehe Kapitel 5.

#### 4.2.2 Installation von PDFTRON (PDF/A Manager Trial Version)

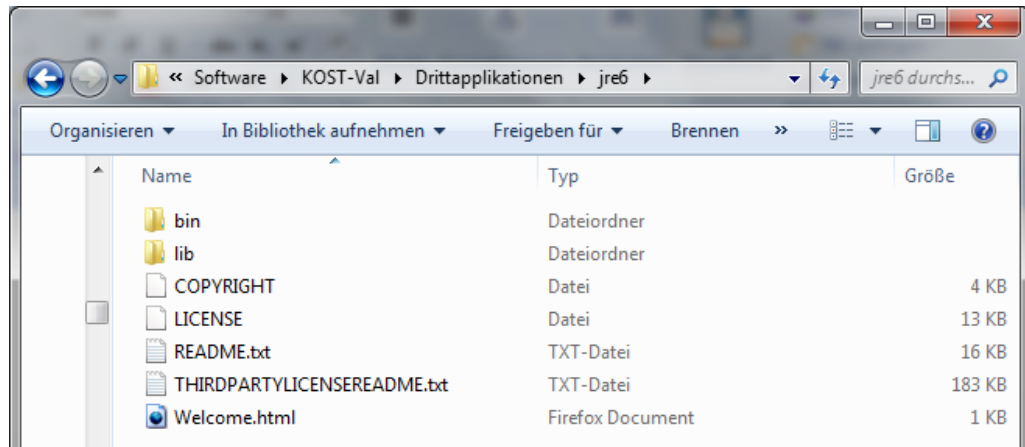
- 1 Command-line-Variante des PDF/A Managers (ZIP-File) unter <http://www.pdftron.com/downloads.html> herunterladen.
  - 2 Den PDF/A Manager in ein beliebiges Verzeichnis entpacken oder in den dafür vorgesehenen Ordner Drittapplikationen\pdfa.
- 
- Hinweis:**  
Sollte ein anderes Verzeichnis gewählt werden, muss entsprechend die Konfigurationsdatei "sipvalidator.conf.xml" angepasst werden. Siehe Kapitel 5.

#### 4.2.3 Installation von JHOVE (Version 1.5)

- 1 Die Applikation JHOVE (zip-File) unter <http://sourceforge.net/projects/jhove/> herunterladen.
  - 2 JHOVE in ein beliebiges Verzeichnis entpacken oder in den dafür vorgesehenen Ordner Drittapplikationen\jhove.
- 
- Hinweis:**  
Sollte ein anderes Verzeichnis gewählt werden, muss entsprechend die Konfigurationsdatei "sipvalidator.conf.xml" angepasst werden. Siehe Kapitel 5.

#### 4.2.4 Installation von Java (Version 6 oder neuer)

- 1 Sollte Java Runtime Environment (JRE) nicht bereits in Version 6 oder neuer<sup>5</sup> installiert sein, muss jre6 unter <http://www.java.com/> heruntergeladen und unter Drittapplikationen\jre6 abgelegt werden.



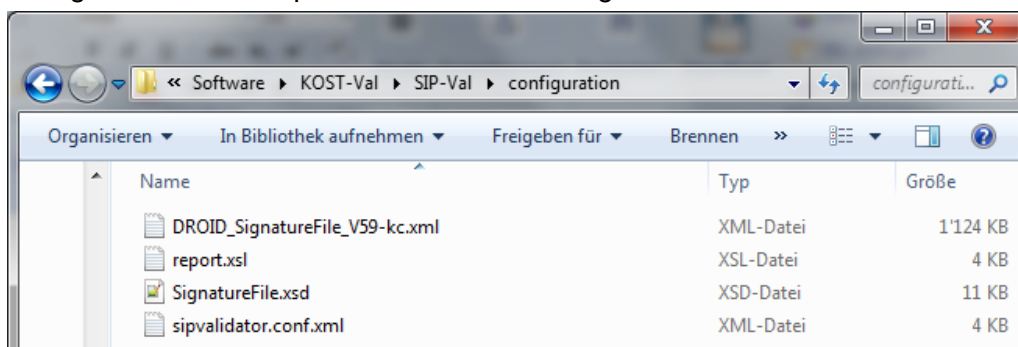
**Hinweis:**

Ist jre6 oder neuer bereits installiert, kann darauf verzichtet werden diese Dateien erneut abzulegen. Beachten Sie, dass Sie die cmd-Scripts wie in Kapitel 6 beschrieben anpassen müssen.

<sup>5</sup> Die aktuell installierte Version lässt sich überprüfen mit der Eingabe von `java -version` im cmd-Fenster.

## 5 Konfiguration des SIP-Val

Sämtliche Konfigurationen des SIP-Val können im Unterordner "configuration" in der Konfigurationsdatei "sipvalidator.conf.xml" vorgenommen werden.



Im Ordner "configuration" sind auch die Dateien "DROID\_SignatureFile\_V59-kc.xml", "SignatureFile.xsd" und "report.xml" abgelegt. Diese werden im Kapitel 5.2 beschrieben.

### 5.1 Bestandteile der Konfigurationsdatei "sipvalidator.conf.xml"

xml-Tag	Beschreibung
<allowedlengthofpaths>	Erlaubte maximale Anzahl Zeichen in Pfadlängen
<allowedlengthoffiles>	Erlaubte maximale Anzahl Zeichen in Dateinamen
<allowsipname>	Vorgaben zum Aufbau des SIP-Namens
<pathtopdftronexe>	Pfadangabe zu PDFTRON (siehe auch Kapitel 4.2, Installation von Drittapplikationen)
<pathtosiardvaljar>	Pfadangabe zu SIARD-Val (siehe auch Kapitel 4.2, Installation von Drittapplikationen)
<patthojhovejar>	Pfadangabe zu JHOVE (siehe auch Kapitel 4.2, Installation von Drittapplikationen)
<patthojhoveconfig>	Pfadangabe zur Konfigurationsdatei von JHOVE (siehe auch Kapitel 4.2, Installation von Drittapplikationen)
<pathtoworkdir>	Pfadangabe zum temporären Arbeitsverzeichnis von SIP-Val
<pathtodroidssignature>	Pfadangabe zur verwendeten DROID-Signaturdatei
<allowedversionbar1>	Angabe ob die BAR-SIP Version 1 erlaubt ist
<allowedversionbar4ech1>	Angabe ob die BAR-SIP Version 4 respektive eCH-0160 Version 1 erlaubt ist
<allowedformats>	Auflistung der erlaubten Dateiformate, bestehend aus Unterelementen mit Detailinformationen
<validatedformats>	Auflistung der zu validierenden Dateiformate, bestehend aus Unterelementen mit Detailinformationen. Für den optionalen Validierungsschritt 3c kann im Untertag <validator> die zu verwendende Drittapplikation auf die einzelnen Formate zugewiesen werden (JHOVE, PDFTRON oder SIARDVAL).

#### 5.1.1 <allowedlengthofpaths>

Erlaubte maximale Anzahl Zeichen in Pfadlängen. Der Initialwert ist **179** und entspricht der Einschränkung im BAR-SIP: weniger als 180 Zeichen.

### 5.1.2 <allowedlengthoffiles>

Erlaubte maximale Anzahl Zeichen in Dateinamen. Der Initialwert ist **40** und entspricht der Einschränkung im BAR-SIP (Version 1): maximal 40 Zeichen.  
Bei der Validierung der SIP-Version 4 gibt es keine Einschränkungen, entsprechend ist dieses Feld nur für die Version 1 relevant.

### 5.1.3 <allowsipname>

Vorgaben zum Aufbau des SIP-Namens. Der Initialwert (Regex-Ausdruck) ist **SIP\_[1-2][0-9]{3}[0-1][0-9][0-3][0-9]\_\\w{3}** und entspricht der Einschränkung im BAR-SIP: SIP\_[Ablieferungsdatum]\_[Name der Abliefernden Stelle](\_[Referenz]), wobei das Datum in der Form JJJJMMTT angegeben werden muss und die Referenz optional verwendet werden kann.

### 5.1.4 <pathtopdftronexe>

Pfadangabe zu PDFTRON. Der Initialwert ist **Drittapplikationen\pdfa\pdfa.exe**. Dabei kann der absolute Pfad angegeben werden oder wie im Initialwert der relative Pfad (aus Sicht von Start.cmd respektive Start\_+3c\_+3d.cmd).

### 5.1.5 <pathtosiardvaljar>

Pfadangabe zu SIARD-Val. Der Initialwert ist **SIARD-Val\siard-val.jar**. Dabei kann der absolute Pfad angegeben werden oder wie im Initialwert der relative Pfad (aus Sicht von Start.cmd respektive Start\_+3c\_+3d.cmd).

### 5.1.6 <pathtojhovejar>

Pfadangabe zu JHOVE. Der Initialwert ist **Drittapplikationen\jhove\bin\JhoveApp.jar**. Dabei kann der absolute Pfad angegeben werden oder wie im Initialwert der relative Pfad (aus Sicht von Start.cmd respektive Start\_+3c\_+3d.cmd).

### 5.1.7 <pathtojhoveconfig>

Pfadangabe zur Konfigurationsdatei von JHOVE. Der Initialwert ist **Drittapplikationen\jhove\conf\jhove.conf**. Dabei kann der absolute Pfad angegeben werden oder wie im Initialwert der relative Pfad (aus Sicht von Start.cmd respektive Start\_+3c\_+3d.cmd).

### 5.1.8 <pathtoworkdir>

Pfadangabe zum temporären Arbeitsverzeichnis von SIP-Val. Der Initialwert ist **work\_SIP-Val**. Dabei kann der absolute Pfad angegeben werden oder wie im Initialwert der relationale Pfad (aus Sicht von Start.cmd respektive Start\_+3c\_+3d.cmd). Dieser Ordner muss nicht angelegt werden.

### 5.1.9 <pathtodroidssignature>

Pfadangabe zur verwendeten DROID-Signaturdatei im Verzeichnis. Der Initialwert ist **SIP-Val\configuration\DROID\_SignatureFile\_V59-kc.xml**. Weitere Informationen sind im Kapitel 5.2 ersichtlich.

### 5.1.10 <allowedversion1>

Angabe ob die BAR-SIP-Version 1 erlaubt (1) ist oder nicht (0). Der Initialwert ist **1**.

### 5.1.11 <allowedversion4>

Angabe ob die BAR-SIP-Version 4 respektive eCH-0160 Version 1 erlaubt (1) ist oder nicht (0). Der Initialwert ist **1**.

### 5.1.12 <allowedformats>

Auflistung der erlaubten Dateiformate bestehend aus Unterelementen mit Detailinformationen. Die Initialwerte sind aus der nachfolgenden Tabelle ersichtlich. Die Liste ist bewusst ausführlich gehalten, so dass die nicht erlaubten Formate einfach entfernt werden können und möglichst wenig nachgetragen werden muss. Angaben betreffend der Extension und der PUID sind auf der Website von PRONOM ersichtlich (<http://nationalarchives.gov.uk/PRONOM/Default.aspx>).

<puid> </puid>	<extension> </extension>	<comment> </comment>
x-fmt/111	txt	Plain Text file
fmt/353	tif, tiff	keine Version Unterscheidung
x-fmt/392	jp2, jpg2, j2c	JPEG2000
fmt/95	pdf	Erkennung der Haupteigenschaften von PDF/A-1a
fmt/354	pdf	Erkennung der Haupteigenschaften von PDF/A-1b
fmt/141	wav, wave	Waveform Audio (PCMWAVEFORMAT)
fmt/6	wav, wave	Waveform Audio
x-fmt/18	csv	Comma Separated Values
fmt/161	siard	Erkennung der SIARD-Haupteigenschaften v1

### 5.1.13 <validatedformats>

Auflistung der zu validierenden Dateiformate bestehend aus Unterelementen mit Detailinformationen. Die Initialwerte sind aus der nachfolgenden Tabelle ersichtlich. Die Liste ist bewusst ausführlich gehalten, so dass die nicht erlaubten Formate einfach entfernt werden können und möglichst wenig nachgetragen werden muss. Angaben betreffend der Extension und der PUID sind auf der Website von PRONOM ersichtlich (<http://nationalarchives.gov.uk/PRONOM/Default.aspx>). Für den optionalen Validierungsschritt 3c kann im Untertag <validator> die zu verwendende Drittapplikation auf die einzelnen Formate zugewiesen werden (entweder JHOVE oder PDFTRON).

<pronomuniqueid> </pronomuniqueid>	<validator> </validator>	<description> </description>
x-fmt/111	-	Plain Text file
fmt/95	PDFTRON	PDF/A-1a
fmt/354	PDFTRON	PDF/A-1b
x-fmt/18	-	Comma Separated Values
fmt/161	SIARDVAL	SIARD
fmt/353	JHOVE	TIFF
x-fmt/392	JHOVE	JPEG2000
fmt/141	JHOVE	Waveform Audio (PCMWAVEFORMAT)
fmt/6	JHOVE	Waveform Audio

### 5.2 "DROID\_SignatureFile\_V59-kc.xml" und "SignatureFile.xsd"

Im Ordner "configuration" sind die Dateien "DROID\_SignatureFile\_V59-kc.xml" und "SignatureFile.xsd" abgelegt. "SignatureFile.xsd" ist das XML Schema zu "DROID\_SignatureFile\_V59-kc.xml" und definiert dessen Struktur.

"DROID\_SignatureFile\_V59-kc.xml" wird für die Formaterkennung benötigt. Es enthält der Stand von DROID per 21.03.2012 und wurde zudem durch die KOST im Bereich der PDF/A-Erkennung<sup>6</sup> angepasst.

Um dieses SignatureFile regelmässig aufdatieren zu können, muss DROID separat installiert und das Update nicht durch den IT-Anbieter unterdrückt werden. Als Alternative stellt die KOST ca. halbjährlich das aktuelle DROID SignatureFile mit ihrer Anpassung im Bereich der PDF/A-Erkennung zum Download zur Verfügung.

### 5.3 "report.xml"

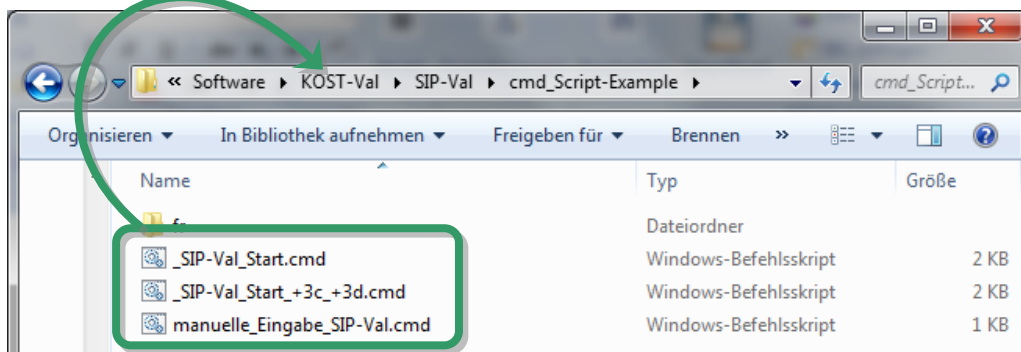
Im Ordner "configuration" ist als Kopiervorlage die Datei "report.xml" abgelegt. Diese Datei wird benötigt, damit der PDFTron Log-Report im Internet Explorer lesbar dargestellt werden kann.

---

<sup>6</sup> In der PRONOM-Datenbank ist die Rangfolge der Formate bei der PDF-Erkennung aus unserer Sicht nicht richtig eingetragen. Zudem muss auch die Interne Signatur angepasst werden. Siehe auch [KorrekturenV52.pdf](#).

## 6 cmd\_Script-Example

Im Unterordner "cmd\_Script-Example" sind Beispiele von Command Scripts abgelegt, die zur Verwendung in den Ordner hereinkopiert werden können, in welchem sich der Ordner SIP-Val befindet. In unserem Beispiel ist dies der Ordner "KOST-Val".



Die einzelnen Scripts werden nachfolgend kurz beschrieben.

### 6.1 "\_SIP-Val\_Start.cmd"

Dieses Script validiert alle vorhandenen SIPs, welche sich im Ordner "KOST-Val\SIPS" befinden, unabhängig davon, ob es sich bei den SIPs um Ordner oder ZIP-Dateien handelt. Die Log-Dateien werden in den Ordner "KOST-Val\logs" geschrieben. Dieses Script verwendet zudem `java.exe` aus dem Ordner "Drittapplikationen". Falls `jre6` auf dem Computer bereits standardmässig installiert ist, kann "`Drittapplikationen\jre6\bin\java.exe`" durch "`java`" ersetzt werden.

### 6.2 "\_SIP-Val\_Start\_+3c\_+3d.cmd"

Dieses Script unterscheidet sich von "\_Start.cmd" durch den zusätzlichen Aufruf der optionalen Validierungsschritte Formatvalidierung (3c) und Zeitraumvalidierung (3d) beim Befehlsaufruf. Dieses Einschalten der optionalen Validierungsschritte ist auch in der Zusammenfassung ersichtlich. Zudem wird am Schluss die Datei "report.xml" in den Ordner "logs" hereinkopiert.

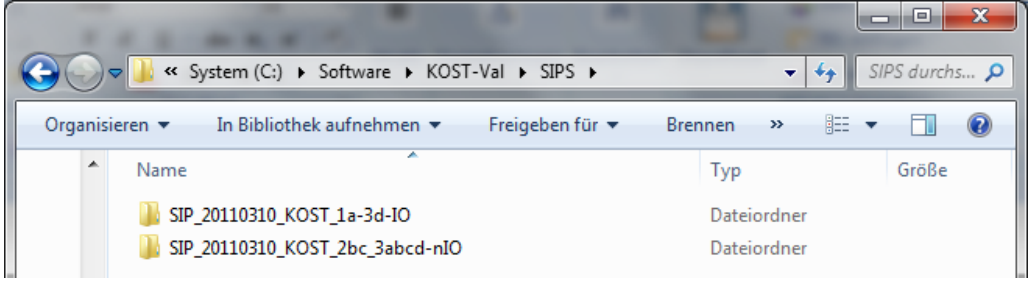
### 6.3 "manuelle\_Eingabe\_SIP-Val.cmd"

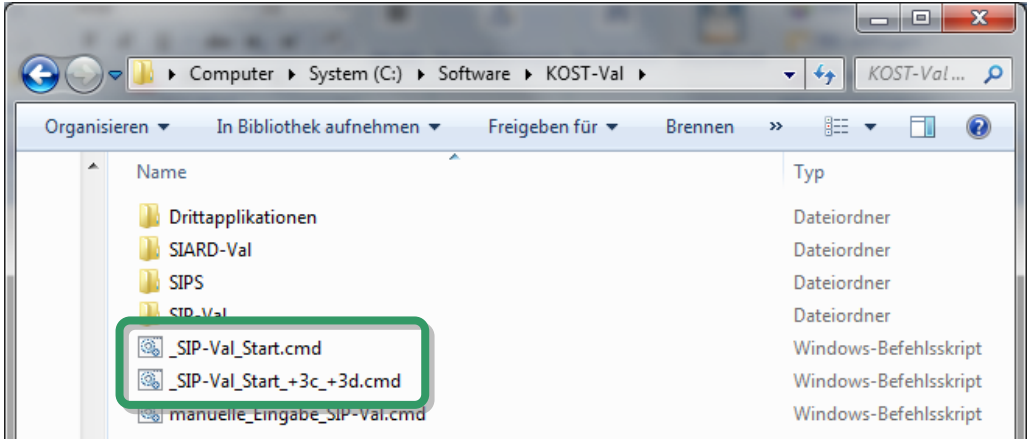
Dieses Script setzt die Java-Settings für das aktuelle cmd-Fenster auf `Drittapplikationen\jre6` und zeigt ein Beispiel der manuellen Eingabe an. Letzteres kann als Kopier-vorlage dienen.


## 7 Validierung starten

### 7.1 Validierung starten mittels Script

- 1 Die zu validierenden SIPs in den Ordner "KOST-Val\SIPS" kopieren.

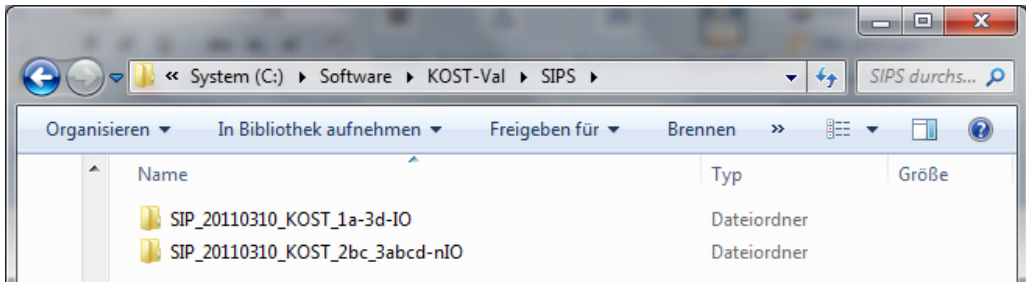
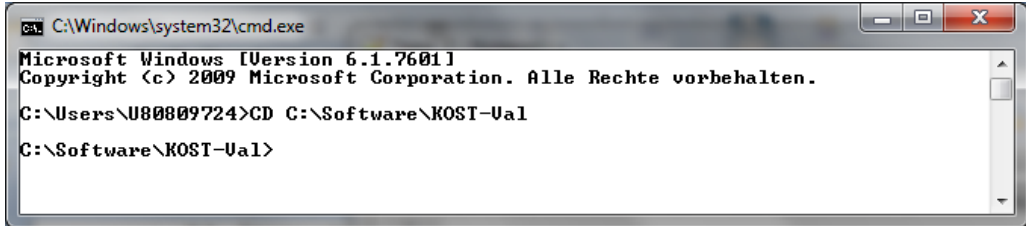
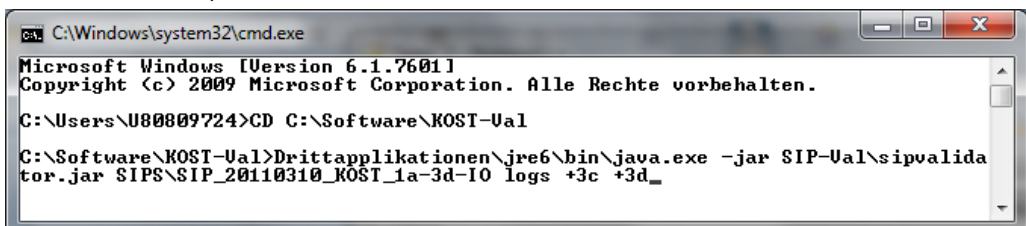

- 2 Starten des Scripts mittels Doppelklick auf "\_SIP-Val\_Start.cmd" oder "\_SIP-Val\_Start\_+3c\_+3d.cmd" (wenn auch die optionalen Validierungsschritte Formatvalidierung (3c) und Zeitraumvalidierung (3d) durchgeführt werden sollen) im Ordner "KOST-Val".


- 3 Alle SIPs wurden validiert, sobald die Zusammenfassung im cmd-Fenster ersichtlich ist.



Die detaillierten Resultate sind im Ordner "logs" abgelegt.

## 7.2 Validierung manuell starten

1	<p>Das zu validierende SIP bereitstellen.</p> 
2	<p>Eingabeaufforderung öffnen (Start → Ausführen ... → cmd [OK]) und in das gewünschte Arbeitsverzeichnis wechseln (CD C:\Software\KOST-Val)<sup>7</sup>.</p> 
3	<p>SIP-Val-Programmaufruf starten (die einzelnen Eingabebestandteile mit Leerzeichen trennen).</p>  <p><b>A</b> Drittapplikationen\jre6\bin\java.exe -jar oder java -jar <b>B</b> SIP-Va\sipvalidator.jar <b>C</b> SIPS\SIP_20110310_KOST_1a-3d-IO <b>D</b> logs <b>E</b> +3c +3d</p> <p><b>A</b> = Java-Befehl (Drittapplikationen\jre6\bin\java.exe -jar oder java -jar<sup>8</sup>), <b>B</b> = relationaler Pfad und die Datei sipvalidator.jar, <b>C</b> = relationaler Pfad und vollständiger Name des zu validierenden SIP-Files, <b>D</b> = relationaler Pfad und Name des Log-Verzeichnisses, <b>E</b> = Aufruf der optionalen Validierungsschritte.</p> <p><u>Anmerkungen:</u> Wenn ein Eingabebestandteil Leerzeichen enthält, muss dieser in Anführungs- und Schlusszeichen eingegeben werden.</p>

<sup>7</sup> Das Laufwerk wird z.B. mit CD /D gewechselt.

<sup>8</sup> Die Eingabe `java -jar` ist möglich, wenn Java Runtime Environment (JRE) Version 6 die Standardversion ist. Siehe auch Kapitel 3.2.3, Installation von Java (Version 6 oder neuer).

Der SIP-Val kann auch von einem beliebigen Ort aus aufgerufen werden. Dies bedingt jedoch die Eingabe von absoluten Pfaden nicht nur im Programmaufruf selber, sondern auch in der Konfigurationsdatei "sipvalidator.conf.xml".

```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\U80809724>C:\Software\KOST-Ual\Driftapplikationen\jre6\bin\java.exe -jar
C:\Software\KOST-Ual\SIP-Val\sipvalidator.jar C:\Software\KOST-Ual\SIPS\SIP_20
110310_KOST_1a-3d-IO C:\Software\KOST-Ual\logs +3c +3d_
  
```

- A C:\Software\KOST-Va\Driftapplikationen\jre6\bin\java.exe -jar / java -jar
- B C:\Software\KOST-Va\SIP-Val\sipvalidator.jar
- C C:\Software\KOST-Va\SIPS\SIP\_20110310\_KOST\_1a-3d-IO
- D C:\Software\KOST-Va\logs
- E +3c +3d

Eine weitere Möglichkeit ist die manuelle Eingabe mit Hilfe des Scripts "manuelle\_Eingabe\_SIP-Val.cmd". Mit diesem Skript kann SIP-Val etwas einfacher aufgerufen werden:

java.exe -jar SIP-Val\sipvalidator.jar SIPS\mein\_SIP logs [+3c] [+3d]

```

C:\Windows\system32\cmd.exe

Mit diesem Skript kann SIP-Ual etwas einfacher aufgerufen werden:
java.exe -jar SIP-Ual\sipvalidator.jar SIPS\mein_SIP logs [+3c] [+3d]

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Software\KOST-Ual>
  
```

Dabei kann die 2. Zeile als Kopiervorlage verwendet werden<sup>9</sup>.

4

Das SIP wurden validiert, sobald "Validierung beendet" oder "Validierung abgebrochen" im cmd-Fenster erscheint.

```

C:\Windows\system32\cmd.exe

[3c] = Valid
[3d] = Valid

TOTAL = Valid

jhove-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_1a-3d-IO.zip.jhove-lo
pdftron-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_1a-3d-IO.zip.pdftr
SIARD-Ual-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_1a-3d-IO.zip.siar

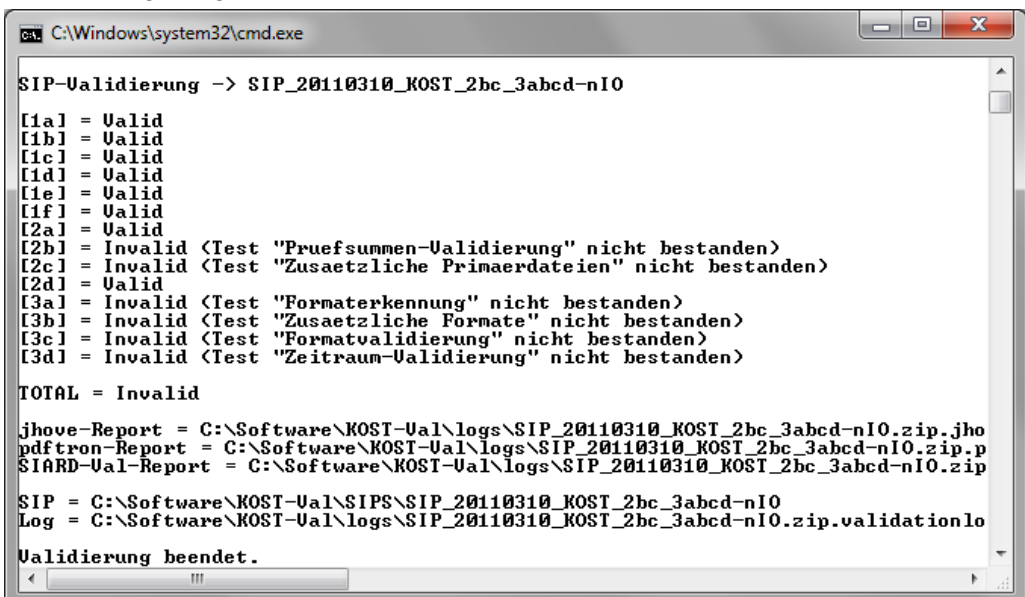
SIP = C:\Software\KOST-Ual\SIPS\SIP_20110310_KOST_1a-3d-IO
Log = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_1a-3d-IO.zip.validationlog.log

Validierung beendet.
  
```

Die detaillierten Resultate sind im Ordner "logs" abgelegt.

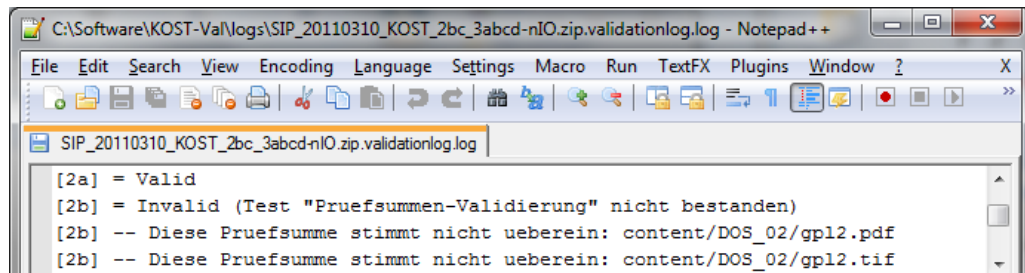
<sup>9</sup> Zum Kopieren den Text der zweiten Zeile mit der Maus markieren, durch Klicken der rechten Maustaste in den Zwischenspeicher legen und wiederum mit der rechten Maustaste an der Eingabeaufforderung einfügen. Anschliessend muss der Befehl noch angepasst werden, indem die optionalen Validierungsschritte oder die eckigen Klammern entfernt werden. Mit der Pfeiltaste zu "mein\_SIP" vorrücken, löschen und gewünschtes SIP eingeben oder hereinkopieren, mit Eingabetaste starten.

## 8 Validierung auswerten

1	Zusammenstellung der Validierungsschritte:	
	<b>Validierungsschritt</b>	<b>Bezeichnung (Stepname)</b>
	1a (Abbruch wenn nIO)	Lesbarkeit
	1b (Abbruch wenn nIO)	primäre Verzeichnisstruktur
	1c (Abbruch wenn nIO)	Verzeichnis- und Dateinamen
	1d (Abbruch wenn nIO)	Schemavalidierung metadata.xml
	1e	SIP-Typ ermitteln
	1f	Primärdateien im Verzeichnis
	2a	Fehlende Primärdateien
	2b	Prüfsummen-Validierung
	2c	Zusätzliche Primärdateien
	2d	Verzeichnung GEVER-Primärdateien
	3a	Formaterkennung
	3b	Zusätzliche Formate
	3c (Optional)	Formatvalidierung
	3d (Optional)	Zeitraum-Validierung
2	<p>Nach dem Start von SIP-Val werden in der Konsolenausgabe die Ergebnisse zu den einzelnen Validierungsschritten ausgegeben. Wenn der Validierungsschritt ohne Fehler abgeschlossen wurde, wird "Valid" ausgegeben, z.B. [2a] = Valid.</p> <p>Sollte ein Validierungsschritt fehlerhaft sein, wird "Invalid" und die Bezeichnung des Validierungsschrittes ausgegeben, z.B. [2b] = Invalid (Test "Pruefsummen-Validierung" nicht bestanden).</p> <p>Am Schluss werden das Gesamtergebnis sowie die Pfade zu den einzelnen Dateien angezeigt.</p>	
	 <pre> C:\Windows\system32\cmd.exe  SIP-Validierung -&gt; SIP_20110310_KOST_2bc_3abcd-nIO  [1a] = Valid [1b] = Valid [1c] = Valid [1d] = Valid [1e] = Valid [1f] = Valid [2a] = Valid [2b] = Invalid &lt;Test "Pruefsummen-Validierung" nicht bestanden&gt; [2c] = Invalid &lt;Test "Zusaetzliche Primaerdateien" nicht bestanden&gt; [2d] = Valid [3a] = Invalid &lt;Test "Formaterkennung" nicht bestanden&gt; [3b] = Invalid &lt;Test "Zusaetzliche Formate" nicht bestanden&gt; [3c] = Invalid &lt;Test "Formatvalidierung" nicht bestanden&gt; [3d] = Invalid &lt;Test "Zeitraum-Validierung" nicht bestanden&gt;  TOTAL = Invalid  jhove-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_2bc_3abcd-nIO.zip.jho pdftron-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_2bc_3abcd-nIO.zip.p SIARD-Val-Report = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_2bc_3abcd-nIO.zip  SIP = C:\Software\KOST-Ual\SIPS\SIP_20110310_KOST_2bc_3abcd-nIO Log = C:\Software\KOST-Ual\logs\SIP_20110310_KOST_2bc_3abcd-nIO.zip.validationlo  Validierung beendet. </pre>	

3

Die Log-Ausgabe führt zusätzliche Detailinformationen über die einzelnen Validierungsschritte auf, insbesondere über die invaliden. Ansonsten ist sie gleich aufgebaut wie die Konsolenausgabe.

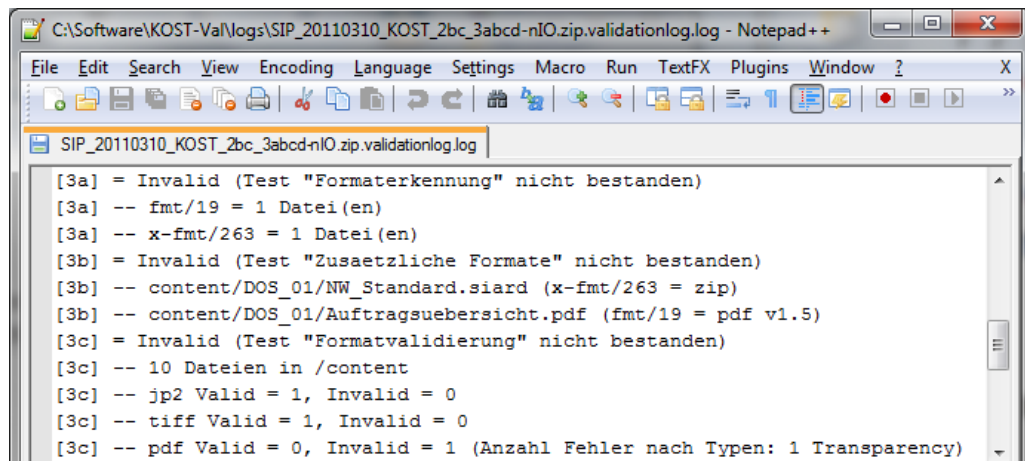


```

C:\Software\KOST-ValNogs\SIP_20110310_KOST_2bc_3abcd-nIO.zip.validationlog.log - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X
SIP_20110310_KOST_2bc_3abcd-nIO.zip.validationlog.log
[2a] = Valid
[2b] = Invalid (Test "Pruefsummen-Validierung" nicht bestanden)
[2b] -- Diese Pruefsumme stimmt nicht ueberein: content/DOS_02/gpl2.pdf
[2b] -- Diese Pruefsumme stimmt nicht ueberein: content/DOS_02/gpl2.tif

```

Die Auswertungen der optional eingesetzten Drittapplikationen werden durch SIP-Val in das jeweilige Verzeichnis abgelegt (siehe auch Kapitel 5). Diese sind insbesondere für den Spezialisten gedacht, da sie technische Angaben über die Formatvalidierung enthalten. Aus dem gleichen Grund wurden bei der Log-Ausgabe von Validierungsschritt 3c wenn möglich verständlichere Informationen zusammengefasst.



```

C:\Software\KOST-ValNogs\SIP_20110310_KOST_2bc_3abcd-nIO.zip.validationlog.log - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X
SIP_20110310_KOST_2bc_3abcd-nIO.zip.validationlog.log
[3a] = Invalid (Test "Formaterkennung" nicht bestanden)
[3a] -- fmt/19 = 1 Datei(en)
[3a] -- x-fmt/263 = 1 Datei(en)
[3b] = Invalid (Test "Zusaetzliche Formate" nicht bestanden)
[3b] -- content/DOS_01/NW_Standard.siard (x-fmt/263 = zip)
[3b] -- content/DOS_01/Auftragsuebersicht.pdf (fmt/19 = pdf v1.5)
[3c] = Invalid (Test "Formatvalidierung" nicht bestanden)
[3c] -- 10 Dateien in /content
[3c] -- jp2 Valid = 1, Invalid = 0
[3c] -- tiff Valid = 1, Invalid = 0
[3c] -- pdf Valid = 0, Invalid = 1 (Anzahl Fehler nach Typen: 1 Transparency)

```

Das vorliegende SIP enthält 10 Primärdateien ([3c] -- 10 Dateien in /content). Davon erfüllen 2 nicht die Formaterkennung ([3a] -- fmt/19 = 1 Datei(en) und [3a] -- x-fmt/263 = 1 Datei(en)). Im Schritt 3b werden diese angegeben: [3b] -- content/DOS\_01/NW\_Standard.siard (x-fmt/263 = zip) (diese Datei ist eigentlich ein ZIP) und [3b] -- content/DOS\_01/Auftragsuebersicht.pdf (fmt/19 = pdf v1.5) (ist kein PDF/A sondern nur ein PDF 1.5). Dies bedeutet, dass 8 Dateien als archivtaugliche Formate erkannt wurden. Im Schritt 3c werden diese 8 Dateien wenn möglich mit einem Validator zusätzlich überprüft. Im Beispiel ist dies ein JPEG2000 (valid), 2 TIF (je einmal valid und invalid), ein WAV (invalid) und ein PDF/A (invalid). Die restlichen 3 Dateien konnten nicht validiert werden (CSV und TXT-Dateien). Bei den invaliden SIARD- und PDF/A-Dateien wird zudem die Anzahl der Fehler pro Hauptgruppe ausgegeben (Invalid = 1 (Anzahl Fehler nach Typen: 1 Transparency))<sup>10</sup>. Diese Angaben sollen dazu dienen, dass der Archivar selbständig entscheiden kann, wie er mit dem vorliegenden SIP und dessen Ergebnissen weiterfahren soll.

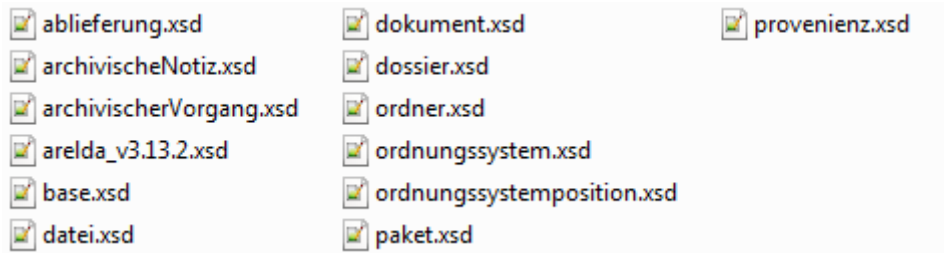
<sup>10</sup> Siehe auch die Studien der KOST zu PDF/A, [http://kost-ceco.ch/cms/index.php?studies\\_de](http://kost-ceco.ch/cms/index.php?studies_de).

## 9 Beschreibung der Validierungsmodule und -schritte

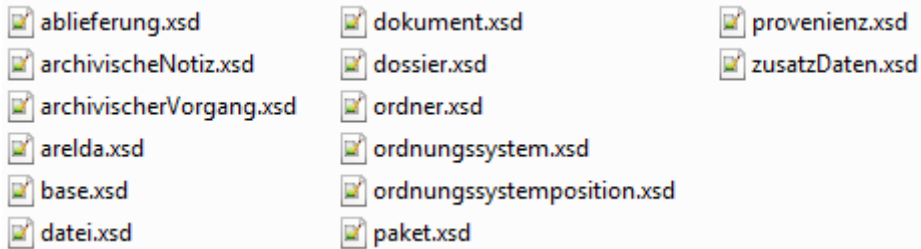
Nachfolgend werden die einzelnen Validierungsschritte detaillierter beschrieben; insbesondere welche Kriterien genau überprüft werden und wie diese auszusehen haben. Diese Kriterien sind einen Auszug der Muss-Kriterien aus der SIP-Spezifikation v1.

### 9.1 Modul 1: Paket- und XML-Konsistenz

- a) Kann das SIP fehlerfrei geöffnet werden? Folgende Container müssen ausgelesen werden können:
- Ordner (keine zusätzliche Endung)
  - ZIP
  - ZIP64
- b) Besteht eine korrekte primäre Verzeichnisstruktur? Das SIP ist valid, wenn eine der folgenden drei Strukturvarianten (definiert durch jeweils zwei Pfade) zutrifft:
- [SIP-Name]/header/metadata.xml und [SIP-Name]/content
  - [SIP-Name].zip/header/metadata.xml und [SIP-Name].zip/content
  - [SIP-Name].zip/[SIP-Name]/header/metadata.xml und [SIP-Name].zip/[SIP-Name]/content
- Anmerkung: [SIP-Name] muss innerhalb einer Verzeichnisstruktur immer identisch sein (betrifft Variante 3).
- c) Entsprechen die Verzeichnis- und Dateinamen den Einschränkungen in der Spezifikation?  
Kriterien:
- Zeichensatz: Die Vorgaben in Bezug auf den zu verwendenden Zeichensatz zur Benennung von Ordnern und Dateien müssen vollumfänglich eingehalten werden. Der erlaubte Zeichensatz zur Benennung von Ordnern und Dateien ist ein Subset von US-ASCII (die Zeichencodes der erlaubten Zeichen sind in den Zeichensätzen US-ASCII, ISO-8859 und UTF-8 identisch). Die folgenden Zeichen sind erlaubt:
    - Buchstaben: A-Z a-z
    - Ziffern: 0-9
    - Andere: ! # \$ % ( ) + , - \_ . = @ [ ] { } ~ SPACE
  - Name des Toplevel-Ordners: Der Name des Toplevel-Ordners beginnt bei einem SIP immer mit der Zeichenfolge SIP\_ gefolgt von bestimmten identifizierenden Angaben. Der Name des Toplevel-Ordners ist immer gemäss dem folgenden Muster aufgebaut:  
SIP\_[Ablieferungsdatum]\_[Name der Abliefernden Stelle]([Referenz]).  
Die Einschränkung der Benennung ist in der Konfigurations-Datei festgehalten, siehe auch 5.1.3 <allowsipname>.
  - Namen der Secondlevel-Ordner: Der Toplevel-Ordner enthält immer genau zwei Ordner. Diese Ordner haben die Namen header/ und content/. Es ist nicht erlaubt, andere Ordner oder Dateien direkt im Toplevel-Ordner zu platzieren.
  - Header-Ordner: Der vordefinierte Ordner im Ordner header/ eines SIP für die Ablieferung heisst: xsd/. Es sind keine weiteren Ordner im Ordner header/ erlaubt. Die vordefinierte Datei im Ordner header/ heisst immer metadata.xml. Diese Datei ist die einzige Datei, die im Ordner header/ erlaubt ist.
  - Namen der xsd-Dateien: Folgende Dateien müssen für die BAR-SIP Version 1 unter header/xsd/ mit der entsprechenden Benennung enthalten sein:



Für die BAR-SIP Version 4 respektive eCH-0160 Version 1 sind es nachfolgende Dateien:



Es sind keine weiteren Dateien erlaubt.

- Je nachdem welche arelda-Datei im SIP vorliegt handelt es sich um die SIP-Version 1 oder 4. Diese wird mit den erlaubten Versionen gemäss der Konfigurationsdatei verglichen. Unerlaubte Versionen werden ausgegeben.
  - Länge der Pfade: Die Pfadlänge zu jeder Datei und zu jedem Ordner innerhalb des Informationspaketes muss weniger als 180 Zeichen betragen. Der Pfad beinhaltet dabei immer auch den Toplevel-Ordner<sup>11</sup>. Auch die Slashes (/) müssen mitgezählt werden. Die Anzahl erlaubter Zeichen ist in der Konfigurations-Datei festgehalten.
  - Länge der Ordner oder Dateien (nur für die Version 1): Der Namen eines einzelnen Ordners oder einer einzelnen Datei darf maximal 40 Zeichen lang sein. Dabei muss bei einer Datei auch die Dateiendung (File Extension) sowie der Punkt zwischen Bezeichnung und Dateiendung mitgezählt werden<sup>12</sup>. Die Anzahl erlaubter Zeichen ist in der Konfigurations-Datei festgehalten.
- d) Entspricht metadata.xml den Schemadateien in /header/xsd? Allfällige XML-Fehler werden angezeigt<sup>13</sup>.
- e) Der SIP Typ wird ermittelt und angezeigt (metadata.xml)//ablieferungstyp: GEVER oder FILE
- f) Es wird angezeigt, falls keine Primärdateien im Verzeichnis /content vorhanden sind (korrektes GEVER-SIP nur zur Archivierung einer Ordnerstruktur, aber fehlerhaftes FILE-SIP).

<sup>11</sup> Beim Toplevel-Ordner wird immer die Anzahl der Zeichen mit der Extension .zip mit eingerechnet, auch wenn das SIP als Ordnerstruktur abgegeben wurde. Dies bedeutet teilweise zusätzliche 4 Zeichen in der Pfadlänge.

<sup>12</sup> Beim Toplevel-Ordner wird immer die Anzahl der Zeichen mit der Extension .zip mit eingerechnet, auch wenn das SIP als Ordnerstruktur abgegeben wurde. Dies bedeutet teilweise zusätzliche 4 Zeichen in der Datei- respektive Ordnerlänge.

<sup>13</sup> Das Schema arelda\_v3.13.2.xsd hat im Bereich von Pruefalgorithmus und Pruefsumme (Zeile 924 und 925) einen Fehler. Diese Elemente müssen vorkommen und sollten entsprechend den Eintrag minOccurs="1" enthalten. Dieser Fehler wird durch das BAR behoben werden. Bis dahin beanstandet SIP-Val nicht, wenn diese Elemente fehlen, obwohl sie gemäss der SIP-Spezifikation v1 vorhanden sein sollten.

## 9.2 Modul 2: Datei-Konsistenz

### 9.2.1 Die obligatorischen Validierungsschritte

- Überprüfen, ob alle in (metadata.xml) /paket/inhaltsverzeichnis referenzierten Dateien vorhanden sind. Fehlende Dateien auflisten.
- Dateien mit Prüfsumme in (metadata.xml) //pruefsumme validieren. Die folgenden Prüfalgorithmen sind erlaubt: MD5, SHA-1, SHA-256 sowie SHA-512.
- Im SIP vorhandene, aber nicht in (metadata.xml)/paket/inhaltsverzeichnis verzeichnete Dateien und Ordner auflisten (ausser Datei metadata.xml)
- Bei GEVER-SIP prüfen, ob alle in (metadata.xml) /paket/inhaltsverzeichnis/content referenzierten Dateien auch in (metadata.xml)/paket/ablieferung/ordnungssystem verzeichnet sind. Allfällige Inkonsistenzen auflisten. ( //dokument[@id] => //datei[@id] ).

## 9.3 Modul 3: Dateiformat- und Datums-Konsistenz

### 9.3.1 Die obligatorischen Validierungsschritte

- Mit DROID (und der File format registry PRONOM) eine Formaterkennung durchführen und die Formate mit der Liste erlaubter Dateiformate vergleichen. Diese Liste ist in der Konfigurations-Datei festgehalten, siehe auch 5.1.12 <allowedformats>.
- Alle nicht spezifizierten Dateien mit entsprechenden Formatangaben auflisten.

### 9.3.2 Die optionalen Validierungsschritte

- (*einschaltbar*) Formatvalidierung, mit JHOVE sowie mit zusätzlichen externen Validatoren (z.B. SIARD: SIARD-Val [KOST] und PDF/A-1b: PDF/A Manager [PDFTRON]) aller nach Dateiformat ausgewählten Dateien in /content (konfigurierbare Liste von den zu validierenden Dateiformaten, siehe auch 5.1.13 <validatedformats>).
- (*einschaltbar*) Zeitraum-Validierung: Überprüfung der Zeitangaben in (metadata.xml)/ablieferung.

Es werden die folgenden hierarchisch aufgeführten Elemente überprüft:

Element	Zeitangabe
Ablieferung	Entstehungszeitraum (von - bis)
Dossier <sup>14</sup>	Entstehungszeitraum (von - bis)
Dokument	Entstehungszeitraum (von - bis)

Verletzt ein Element mindestens eines der nachfolgenden Kriterien, wird es (und der Zeitraums-Validierungsschritt) als invalid eingestuft:

- Der Zeitwert „Bis“ muss grösser oder gleich als der Zeitwert „Von“ sein, und keiner der beiden Zeitwerte darf in der Zukunft liegen.
- Die Zeitangaben eines Elements müssen innerhalb des Entstehungszeitraums des übergeordneten Elements liegen.

<sup>14</sup> Einzig beim Dossier muss der Entstehungszeitraum angegeben werden. Alle übrigen Element-Typen sind auch valid, wenn sie keinen Entstehungszeitraum aufweisen.

## 10 Urheberrecht

SIP-Val ist eine Entwicklung der KOST. Alle Rechte liegen bei der KOST. SIP-Val wurde im 2011 durch die KOST unter der GNU General Public License v3+ veröffentlicht.

<b>Notice:</b>	This product includes software developed by the Apache Software Foundation ( <a href="http://www.apache.org/">http://www.apache.org/</a> ).
----------------	---

SIP-Val stützt sich auf folgende unveränderte Komponenten anderer Hersteller, welche direkt im Quellcode von SIP-Val eingebunden sind:

Drittprogramm / -Komponente	Version	Lizenz
<b>Apache Commons</b> <a href="http://commons.apache.org/">http://commons.apache.org/</a> - commons-collections-3.2.1.jar - commons-configuration-1.6.jar - commons-digester-1.8.jar - commons-lang-2.4.jar - commons-logging-1.1.1.jar	3.2.1 1.6 1.8 2.4 1.1.1	Apache License 2.0
<b>Apache log4j</b> <a href="http://logging.apache.org/log4j/">http://logging.apache.org/log4j/</a>	1.2.12	Apache License 2.0
<b>Apache Xalan-Java</b> <a href="http://xml.apache.org/xalan-j/">http://xml.apache.org/xalan-j/</a>	2.7.0	Apache License 2.0
<b>Apache Xerces</b> <a href="http://xerces.apache.org/">http://xerces.apache.org/</a>	2.7.1	Apache License 2.0
<b>DROID</b> <a href="http://sourceforge.net/apps/mediawiki/droid/">http://sourceforge.net/apps/mediawiki/droid/</a>	5.0.3	3c BSD-Lizenz
<b>Junit 4.4</b> <a href="http://www.junit.org/">http://www.junit.org/</a>	4.4	CPL v1.0
<b>Spring Framework API</b> <a href="http://static.springsource.org/spring/docs/3.0.x/api/">http://static.springsource.org/spring/docs/3.0.x/api/</a>	3.0.0	Apache License 2.0
<b>zip64</b> <a href="http://sourceforge.net/projects/zip64file/">http://sourceforge.net/projects/zip64file/</a>	1.02	GPL v2+ Lizenz

SIP-Val stützt sich auf folgende unveränderte Bibliothek anderer Hersteller, welche separat durch den Anwender installiert werden muss, damit SIP-Val funktioniert:

Drittprogramm / -Komponente	Version	Lizenz
<b>JHOVE</b> <a href="http://hul.harvard.edu/jhove/">http://hul.harvard.edu/jhove/</a>	1.5	LGPL v2.1 Lizenz

Die Benutzer von SIP-Val sind gehalten, die Lizenzbestimmungen all dieser Komponenten zu befolgen, welche im Verzeichnis SIP-Val\license vorliegen.

### Anmerkung

Für die Durchführung einer PDF/A-Validierung (optionaler Validierungsschritt 3c) wurde zu Demonstrationszwecken die indirekte Verwendung (via Konsolenaufruf) der Testversion der kommerziellen Software PDF/A Manager von PDFTRON implementiert sowie für die SIARD-Validierung das KOST Tool SIARD-Val. Diese Softwares sind nicht Bestandteil von SIP-Val und müssen zur allfälligen Verwendung separat beschafft und installiert werden. Weitere Informationen dazu siehe Kapitel 4.2.1.