Preservation-Planning
TIFF-Analyse – rollende Planung

Inhalt

1 Planung

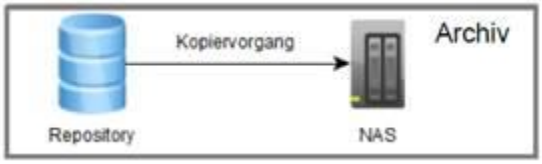
Wir haben folgendes Vorgehen mit dem *Digital Humanities Lab Basel* bezüglich Analyse von TIFF Dateien abgeklärt:

1. Jhove Analyse ermittelt grundsätzlich korrekter Aufbau der Datei, und Information über Alignement, Order und Typisierung der Tags und generiert das Jhove LOG
2. Ein Programm vom *Digital Humanities Lab (DHLAB)* liest sequenziell alle Tag, Tagnummern Tag Typen und Taginhalt und schreibt die Werte pro File in eine Datenbank (SQLite oder mySQL)
3. Mit EXIFTool werden allenfalls vorhanden EXIF und IPCT Metadaten ausgelesen und ebenfalls in die Datenbank geschrieben.
4. Als letztes wird optional ein *Thumbnail* generiert und in eine Datei geschrieben, damit kann die Integrität der Bitmap validiert und Bitmap Fehler einfach erkannt werden (z.B. alles Schwarz). Offen ist dabei, ob wir das Thumbnail LOG mitnehmen können oder ob die Auswertung nur im Archiv erfolgen kann.

Zur externen Auswertung wird das Jhove LOG und die TIFF-TAG/Metadaten Datenbank benötigt, das Thumbnail LOG muss allenfalls im Archiv ausgewertet werden. Das Thumbnail LOG wird nur genötigt, wenn im Jhove LOG oder in der TIFF-TAG/Metadaten Datenbank Anomalien festgestellt werden um.

2 Vorgehen in einzelnen Schritten

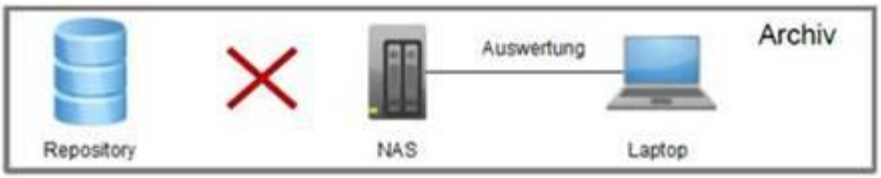
- Das Programm vom DHLAB muss vom Team von Herr Rosenthaler noch angepasst werden.
- Parallel dazu werden wir hier in der Geschäftsstelle eine Testinstanz mit unserer Hardware und eigenen Beispieldateien aufbauen und dort alle Komponenten testen können.
- In dieser Phase zeigen wir auch gerne den Beteiligten wie alles funktionieren soll. Wir hoffen, dass wir Ende Februar mit allen Vorbereitungen soweit sind.
- Sobald die Frage der Verfügbarkeit eines Raids in den beteiligten Archiven geklärt ist, können wir aber schon mit dem Kopieren der Dateien beginnen.
  Dafür muss auf einem archivinternen Rechner mit Zugang zum Repository und dem NAS ein Kopierprogramm im Hintergrund laufen können (z.B. robocopy vom Microsoft)



3 Angaben von den Archiven

Von Seiten der Archive benötigen wir nun noch folgende Angaben:

- Wer ist unser Ansprechpartner in diesem Projekt (BAR Marcel, StABS Markus, StASG ?)
- Grobes Mengengerüst: Anzahl Dateien / Speicherplatz
- Soll das NAS (10 TByte Raid 1) von KOST/DHLAB zur Verfügung gestellt werden oder kann das Archiv ein NAS für 4 Mnt. ausleihen?
- Soll der Rechner für die Auswertung von KOST/DHLAB zur Verfügung gestellt werden oder kann das Archiv ein Computer für 4 Mnt. ausleihen?
- Müssen die Speicherplatten im Raid nach der Auswertung im Archiv bleiben, wenn das NAS von KOST/DHLAB stammt oder reicht es die Daten mit einem sog. Shredder Programm zu überschreiben?



4 Entwicklung

Die einzelnen Aufgaben (Sitzung vom 15. März)

0) Loop durch alle Verzeichnisse auf dem RAID Server mit Kopieren jeweils einer TIFF Datei vom RAID auf den Auswertungsrechen (mit Restartfähigkeit) KOST
1) Ein Hash Wert (z.B. md5) wird für die Datei berechnet um Doubletten zu erkennen. Der Hash Wert dient als Schlüssel in der DB KOST
2) Die Jhove Analyse ermittelt grundsätzlich korrekter Aufbau der Datei und Information über Alignement, Order und Typisierung der Tags und schreibt das Jhove LOG. Der LOG-Offset wird zur schnelleren Analyse in die DB geschrieben. KOST
3) Ein Programm vom Digital Humanities Lab (DHLAB) liest sequenziell alle Tag, Tagnummern Tag Typen und Taginhalt und schreibt die Werte in die DB
4) Mit EXIFTool werden allenfalls vorhanden EXIF und IPCT Metadaten ausgelesen und ebenfalls in die Datenbank geschrieben. KOST
5) Als letztes wird optional ein Thumbnail generiert und in eine lokale Datei geschrieben, damit kann die Integrität der Bitmap validiert und Bitmap Fehler einfach erkannt werden (z.B. alles Schwarz). Der Thumbnail Name wird ebenfalls in der DB festgehalten. Offen ist dabei, ob wir das Thumbnail LOG mitnehmen können oder ob die Auswertung nur im Archiv erfolgen kann.

4.1 Infrastruktur und Programmiersprachen

Vorschlag der Geschäftsstelle:

Datenbank: SQlite Vorteil, kein Server, keine Administrationhttps://www.sqlite.org/

Loop-Programm: Golang, Vorteil kompilierte Sprache, auf allen Plattformen verfügbar, API zu SQLite und einfacher als C/C++ https://golang.org/

Arbeitsbereich: GitHub https://github.com/KOST-CECO/TiffAnalyseProject

Vorschlag Makrus:

Linux als Betriebssystem auf den Auswertungsrechnern: *Linux Mint 17.3*

4.2 Analyse Programm

Das Analyse Loop Programm liest alle TIFF Dateien (Korpus) vom NAS und führt mit der jeweils gelesenen Datei mehrere Analyseschritte durch aufrufen externen Programme aus. Der Loop Prozess ist zweiteilig und besteht aus sein Initialisierung der Prozessdatenbank und der eigentlichen Analyse.

4.2.1 Korpus Initialisierung

Ein erster Initialisierungsschritt erstellt die Datenbank und schreibt für jede TIFF Datei einen Eintrag mit dem Pfad und Dateinamen als Schlüssel. Argumente sind "Datenbank" und "Datei Root". Die Initialisierung kann mehrfach aufgerufen werden und fügt so neue Verzeichnispfade zur Datenbank hinzu.

Damit für eine spätere Auswertung ausserhalb der Archive problemlos gearbeitet werden kann, werden Dateinamen und Dateipfad, welche allenfalls Rückschlüsse auf den Inhalt der Dateien erlauben, in einer separaten Tabelle gehalten. Als gemeinsamer Schlüssel für alle spätere Auswertung wird der MD5 Schlüssel der TIFF Datei

verwendet. Weil zum Berechnen des MD5 Schlüssels die ganze Datei gelesen werden muss, wird diese Berechnung erst im Analyseschritt vorgenommen.

4.2.2 Korpus Analyse

Im Analyse Fall werden die Dateieinträge in der Datenbank abgearbeitet und als erstes ein MD5 Schlüssel berechnet. Damit können Doubletten im Dateisystem erkannt werden. Anschliessend werden die Analyse Tools im Kommandozeilen Modus aufgerufen.

Durch die Verwendung einer Datenbank ist jederzeit ein Abbrechen und wieder Starten der Analyse möglich.

- Dem Analyse Tools im Kommandozeilen Modus wird der Dateipfad und der Pfad zur Logdatei übergeben.
- Kann das Analyse Tools kein Logfile im Append Modus öffnen, kann der Log Output vom Loop Programm entweder an die Logdatei angehängt oder in die Datenbank geschrieben werden.
- Der aktuelle Offset der Log Datei wird in der Datenbank gespeichert.
- Der Exit Value des Analyse Tools wird in der Datenbank festgehalten
- Es kann festgelegt werden, ob der System Output des Analyse Tools in eine spezielle Output Datei geschrieben werden oder in der Datenbank gespeichert werden soll.
- Eine Logrotation verhindert allzu grosse Logdateien. Nach "n" definierten Loop Schritten wird eine neue Logdatei begonnen.

4.2.3 Datenmodell

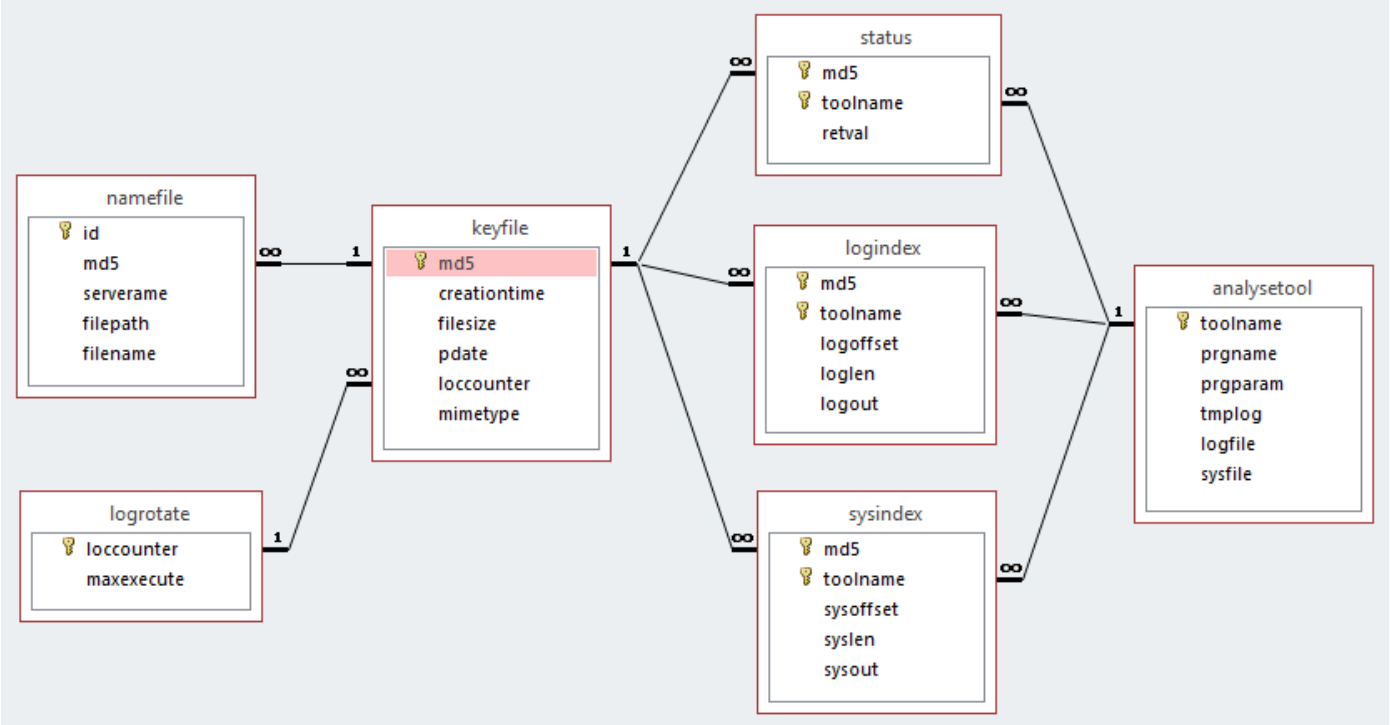
Die Tabellen *keyfile* und *namefile* enthalten die den primären Verzeichnisscan, also die Namen aller Dateien mit Dateigrösse und *Creation Time* sowie sie aus dem Lesen der Verzeichnisstrukturen erstellt werden können. Der MD5 Schlüssel wird erst beim eigentlichen Analysedurchgang angelegt, weil dafür die Dateien ebenfalls vollständig gelesen werden müssen.

Zum Ausführen der Analysemodule werden die notwendigen Informationen aus der Tabelle *analysetool* ausgelesen, d.i Programmname und Pfad, Logdatei, Datei bzw. BLOB für den System Output.

Die Tabelle *status* hält den Exit Status des Analyseprogramms fest.

Die Tabellen *logindex* und *outindex* speichern den Offset in die jeweilige Log oder Output Datei für die eben analysierte Datei.

Auf GitHub https://github.com/KOST-CECO/TiffAnalyseProject/tree/master/createDB sind die notwendigen SQL-loader scripts für die Datenbank abgelegt (sowohl für mySQL wie SQLite3).



tablename	name	type	description
analysetool	toolname	CHARACTER VARYING(30)	Name des registrierten Analyseprograms in Kurzform
	prgname	CHARACTER VARYING(255)	Pfad und Dateiname zum Analyseprograms
	prgparam	CHARACTER VARYING(255)	Parameter des Analyseprogramms mit Wildcards %file% und %log%
	tmplog	CHARACTER VARYING(255)	Temporäre Logdatei: ersetzt Wildcards %log% beim Ausführen des Analyseprogramms, Fehlen meint keine Log Datei schreiben
	logfile	CHARACTER VARYING(255)	Pfad und Dateiname der mit diesem Analyseprogramms verbunden Logdatei: Ist kein Logfile definiert wird in LOB "logout" gespeichert
	sysfile	CHARACTER VARYING(255)	Pfad und Dateiname der mit diesem Analyseprogramms verbunden Ausgabedatei: Ist kein Sysfile definiert wird in LOB "sysout" gespeichert

keyfile	md5	CHARACTER VARYING(32)	MD5 Hashwert und Referenz zum „namefile“
	creationtime	TIMESTAMP(0)	Entstehungszeitpunkt der Datei laut Dateisystem
	filesize	LONG	Dateigrösse in Byte
	pdate	TIMESTAMP(0)	Zeitpunkt und Flag für den Abschluss der gesamten Analyse
	loccounter	INTEGER	Zähler für "logfile" bzw. "sysfile" beginnend mit Eins
	mimetype	INTEGER	Internet Media Type, auch MIME-Type aufgrund der Magic Number

logindex	md5	CHARACTER VARYING(255)	MD5 Schlüssel der TIFF Datei
	toolname	CHARACTER VARYING(255)	Kurzname des Tools
	logoffset	INTEGER	Offset in die Ausgabedatei analysetool.logfile
	loglen	INTEGER	Länge des Logausgabe
	logout	CHARACTER LARGE OBJECT	vollständige LOG Ausgabe des Analysetools

logrotate	loccounter	INTEGER	Zähler für "logfile" bzw. "sysfile" beginnend mit Eins
	maxexecute	INTEGER	Maximal Verarbeitungsschritte pro "logfile" bzw. "sysfile"

namefile	id	INTEGER	Referenz zu "keyfile"
	md5	CHARACTER VARYING(32)	MD5 Hashwert
	serverame	CHARACTER VARYING(255)	Name des NAS Servers oder des zugeordneten Laufwerkbuchstabens
	filepath	CHARACTER VARYING(255)	Dateipfad
	filename	CHARACTER VARYING(255)	Dateiname mit Dateiextension

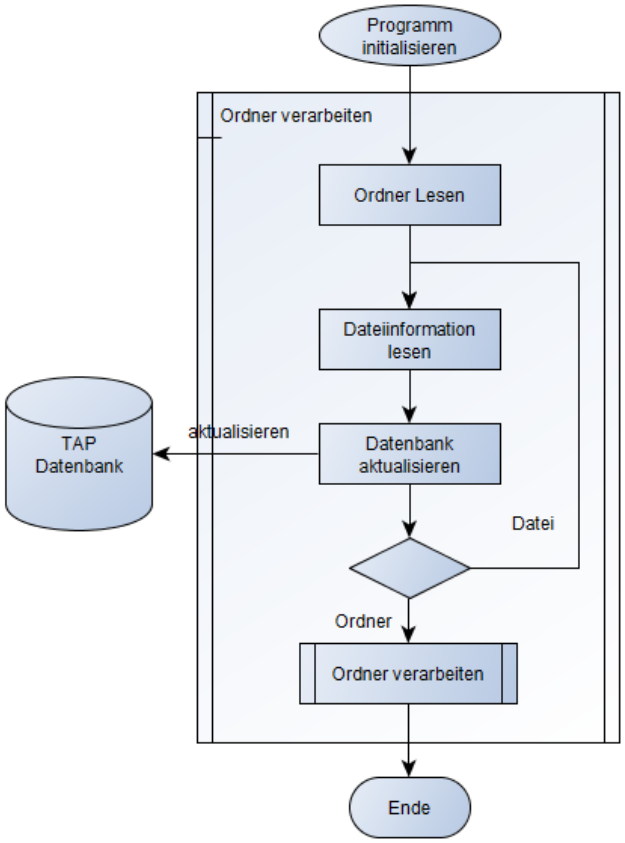
status	md5	CHARACTER VARYING(255)	MD5 Schlüssel der TIFF Datei

	toolname	CHARACTER VARYING(255)	Name des registrierten Analyseprograms in Kurzform
	retval	CHARACTER VARYING(255)	Rückgabe Wert des Tools (Exit Status 0 = erfolgreicher Abschluss) http://www.hiteksoftware.com/knowledge/articles/049.htm

sysindex	md5	CHARACTER VARYING(255)	MD5 Schlüssel der TIFF Datei
	toolname	CHARACTER VARYING(255)	Kurzname des Tools
	sysoffset	INTEGER	Offset in die Ausgabedatei analysetool.sysfile
	sylen	INTEGER	Länge des Konsolenausgabe
	sysout	CHARACTER LARGE OBJECT	vollständige SystemOut Ausgabe des Analysetools: stderr & stdout

4.2.4 Skript Programmierung

Wie schon oben beschrieben gliedert sich die Script Programmierung in ein "Initial Loop" Programm, das sämtliche TIFF Dateien liest und einen Verarbeitungsindex anlegt. Das „Initial Loop“ Programm soll mit unterschiedlichen und mehreren Foldern als Start Parameter umgehen können.



„Initial Loop“ liest sämtliche TIFF Dateien

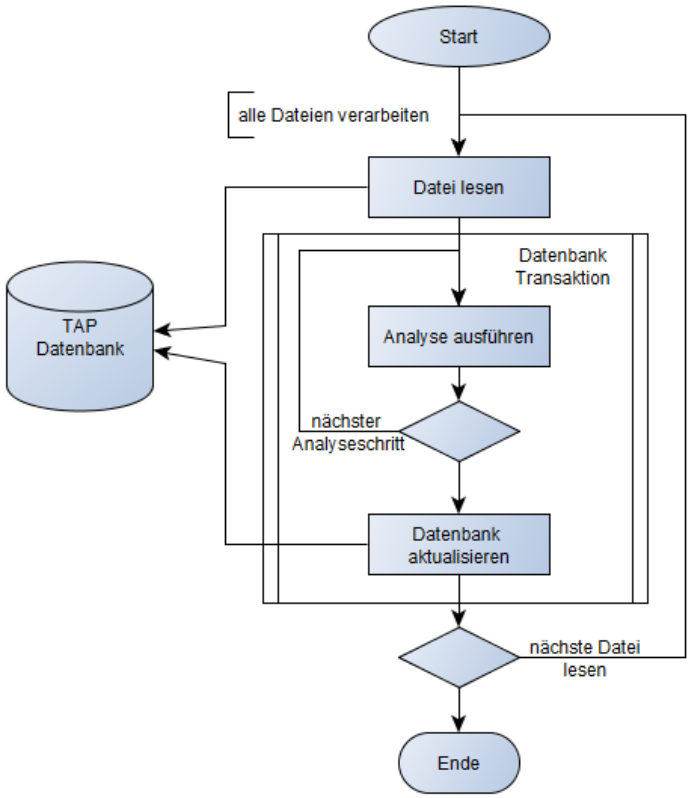
Der Teil „Process Loop“ liest alle Dateien aus dem Verarbeitungsindex, berechnet einen md5 Schlüssel, erkennt via md5 Doubletten und führt dann jeweils alle in der Tabelle *analysetools* beschriebenen Analyseschritte/Programme aus.

Je nach Konfiguration in der Tabelle *analysetools* wird die Programmausgabe in eine Tabelle (LOB) oder eine Log Datei geschrieben. Wobei umfangreichere Analyseergebnisse mit Vorteil in Log Dateien geschrieben werden. Nach eine bestimmten Anzahl Analyseschritten wird eine neue Logdatei initialisiert.

Unterschieden wird zwischen Bildschirmausgabe und Logdatei der Analyseprogramme. Die Bildschirmausgabe umfasst die Konsolenausgabe *stdout* und

errout. Die Logdatei ist eine explizit vom Analyseprogramm definierte *log* oder *output* Datei. Diese Datei wird vor dem Start des Analyseprogramms gelöscht und anschliessend an die entsprechend definiert Log Datei angehängt.

Der Exit Status des Analyseprogramms wird für jeden Programmlauf als Text in *status.retval* festgehalten.



„Process Loop“ führt alle Analyseschritte pro TIFF Datei aus

4.2.5 Skript installieren

Folgende Schritte müssen ausgeführt werden:

1. Script Package in einem Ordner mit entsprechenden Rechten (create/execute) installieren.
2. In Script *load_ToolList.sql* die entsprechenden Analyseprogramme und Log Dateien eintragen (siehe Beispiele)
3. Mit Script create_TAP.bat eine neue Datenbank anlegen
   `usage: create_TAP.bat path/dbname.db`
4. Mit inloop.exe alle Dateien einlesen und in die Datenbank schreiben. inloop.exe kann mit mehreren Startordnern mehrfach aufgerufen werden
   `usage: iniloop.exe folder database`
5. Mit runloop.exe die Verarbeitung starten
   `usage: runloop.exe [options] database`

5 Analysemodule

Folgende Analysemodule sind im Augenblick vorgesehen, weiter können nach Bedarf dazu kommen:

- [MD5 (integriert in den Loop)]
  Das Berechnen des MD5 Schlüssels gehört nicht zu den Analysemodulen, wird aber vor der Analyse durchgeführt.
- Simple Formaterkennung mit *files*
  http://gnuwin32.sourceforge.net/packages/file.htm
  Mit der Formaterkennung wird werden falsch gelabelte Dateien erkannt.
- Validierung mit JHOVE http://jhove.openpreservation.org/
  Die JHOVE Validierung ermittelt die grundlegende Struktur der TIFF Datei. Wichtig sind hier *Status* und *InfoMessage*.
- DPF-Manager http://www.preforma-project.eu/dpf-manager.html
  Alternative zu JOHVE aus dem Performa Projekt.
- checkit_tiff: *a conformance checker for baseline TIFFs* der Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden
  https://github.com/SLUB-digitalpreservation/checkit_tiff
- TIFF Tag Extraktion vom DHLAB (= vollständige Tagextraktion)
  Das C++ Programm extrahiert alle TIFF Tags in eine CSV Tabelle (TIFF Tag, Datentyp und Wert)
- EXIF Extraktion mit ExifTool http://owl.phy.queensu.ca/~phil/exiftool/
  Eingebettete EXIF und XMP Metadaten werden extrahiert.
- Für jede Datei wird ein sehr kleines *Thumbnail* generiert und in eine gemeinsame Log Datei geschrieben. In diesem Schritt wird erst die Payload oder Bitmap der TIFF Datei untersucht. Eine korrekte Konvertierung belegt die korrekte Implementierung von Komprimierung und Farbraum.

Folgendes soll gelten:
Die Reihenfolge der Module ist beliebig, ausser das MD5 als erstes ermittelt werden muss. Alle Schritte werden ausgeführt, auch wenn bei einem Modul ein Fehler auftritt.

6 Auswertung

Die Analysemodule werden ohne weiter Auswertung des Log- oder Systemausgabe ausgeführt. Der einzige Hinweis über das erfolgreiche Ausführen ist der Rückgabewert oder Exit Status (=retval) in der Statustabelle (=status).

Die Auswertung erfolgt vollständig offline, entweder im Archiv oder ausgelagert. Der Vorteil dieses Vorgehens ist der, dass verschiedene Auswertungen möglich sind oder dass die Auswertungsmethode auch wären der Arbeit noch verändert werden kann, etwas was in der Analysephase nicht möglich ist.

7 Testdaten

Öffentlich zugängliche Testdaten:

https://github.com/EasyinnovaSL/DPFManager/tree/develop/src/test/resources

https://github.com/openpreserve/jhove/tree/junit_tests/examples/tiff

http://sipi.usc.edu/database/database.php?volume=misc&image=7#top