

## 06 - Das Dateiformat TIFF

### Was ist TIFF?

TIFF ist ein flexibles Dateiformat für die Speicherung von Bildern. Es besteht im Wesentlichen aus Datenfeldern, sogenannten Tags, die bestimmte Eigenschaften eines Bildes kodieren. Daher hat das Datenformat auch seinen Namen: Tagged Image File Format.

### TIFF-Aufbau

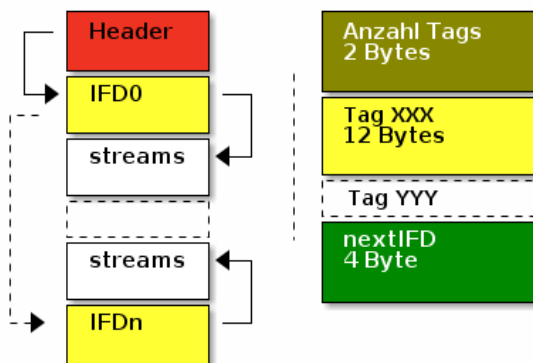


Abb. 1 Struktur einer TIFF-Datei und Aufbau IFD

|                    |                        |                   |                            |
|--------------------|------------------------|-------------------|----------------------------|
| Tag ID<br>(2 Byte) | Field Type<br>(2 Byte) | Count<br>(4 Byte) | Value / Offset<br>(4 Byte) |
|--------------------|------------------------|-------------------|----------------------------|

Abb. 2. Tag (Ausschnitt eines IFDs)

TIFF-Dateien sind sehr einfach aufgebaut. Am Anfang hat jedes TIFF einen konstanten Header. Dieser besteht aus zwei MagicBytes "II" oder "MM", die eine Datei als TIFF identifizieren und aus zwei Bytes, die die Zahl 42 in eine von zwei möglichen Byte-Reihenfolgen kodieren.

Nach dem Header folgt ein Offset (Position in der Datei) auf das erste Image File Directory (IFD). Alle Offsets werden mit vier Bytes kodiert, daraus ergibt sich auch die maximale Dateigröße von 4GB.

Jedes IFD beginnt mit einem zwei Byte großen Eintrag für die Anzahl der darin kodierten Tags, gefolgt von einem Array dieser Tags und einem Offset-Eintrag, der entweder auf das nächste IFD zeigt oder auf den Wert Null, der das Ende anzeigt.

Jedes Tag in dem Array ist 12 Bytes groß. Die ersten zwei Bytes bezeichnen das Tag, die nächsten zwei den Datentyp der Taginformationen. Danach folgen vier Bytes, die die Anzahl der Werte anzeigen, gefolgt von vier Bytes, die entweder die Werte direkt enthalten oder einen Offset aufweisen, an dem die Werte in der Datei stehen.

Durch diese Struktur ist TIFF sehr flexibel. Für eine ausführliche Darstellung des Aufbaus von TIFF-Dateien anhand eines Beispiels sei auf <http://art1pirat.blogspot.de/2013/07/baseline-tiff.html> verwiesen.

### Vorteile von TIFF als LZA-Format

TIFF 6.0 ist seit 1986 stabil. Die offene Spezifikation ist sauber und übersichtlich verfasst.

Mehrere Eigenarten von TIFF machen das Format robust. So müssen alle Offsets an geradzahigen Adressen liegen und die Tags eines IFDs aufsteigend sortiert vorliegen. Auch wenn Bitfehler falsche Offset-Adressen erzeugen würden, wären die Auswirkung lokal begrenzt und mit entsprechendem Aufwand reparierbar (mit einer Ausnahme, s. u.).

Durch die Angabe eines baseline-Profiles wird eine Mindestanforderung definiert, die ein TIFF-Viewer erfüllen muss.

## Vorbehalte gegenüber TIFF

### - strukturelle Vorbehalte

TIFF 6.0 definiert drei verschiedene Tag-Bereiche, der erste ist mit Tags aus der Spezifikation belegt. Der zweite umfasst die Tags 0x8000-0xfde7 und ist als private markiert. Diese Tags hatte Adobe auf Anfrage an Hersteller vergeben. Eine offengelegte Dokumentation der verwendeten Tags ist durch Adobe nicht gegeben. Etliche dieser Tags werden durch Standards und offen publizierte Herstellerangaben näher definiert. Der dritte Bereich von 0xfde8-0xffff ist als reusable deklariert und kann von jedermann frei verwendet werden. Daraus folgt, dass im Prinzip nur die Tags des offen spezifizierten, ersten Bereichs validiert werden können.

Ein ähnliches Problem ergibt sich theoretisch auf für die Art der zugelassenen Kompressionsverfahren. Adobe hat in der TIFF6.0 Spezifikation (und in den Ergänzungen) einige Verfahren offen spezifiziert, Hersteller konnten jedoch andere Verfahren mit eigener Kompressions-ID registrieren lassen.

Ein anderes Problem mit TIFFs ergibt sich aus der Nutzung von Offsets. Treten Bit-Fehler in den Tags auf, die auf eine Offset-Adresse verweisen, so können uU. die Daten dieses Tags nicht mehr korrekt interpretiert werden. Durch die o.g. Merkmale wären einige Ansatzpunkte für eine Reparatur vorhanden.

Doch im Zusammenspiel zwischen der genutzten Kompression und der Art des Verweises auf die komprimierten Datenströme durch die Tags StripByteCount (0x117) und StripOffsets (0x111) ergeben sich zwei Probleme:

- Es gibt keine Möglichkeit für den Dekompressionscode (und damit auch für keinen Validator) festzustellen, wie viele Bytes der gepackte Datenstrom umfasst. Das Tag StripByteCounts ermöglicht es nur, den Zielpuffer für die entpackten Daten zu bestimmen.
- Das Tag StripOffset verweist auf ein Array von Offsetadressen, die ihrerseits auf die gepackten Bilddaten verweisen. Sollte das Tag StripOffset eine falsche Adresse hinterlegt haben, sind die Bilddaten verloren. Wenn keine Kompression verwendet wird, könnte man die Bilddaten durch eine optische Analyse des Datenstroms rekonstruieren.

Als mögliches weiteres Problem können sich Tags erweisen, die widersprüchliche Aussagen treffen. So definieren die EXIF-Tags ihrerseits ein DateTimeOriginal-Tag (0x9003), während TIFF selbst DateTime (0x0132) bereitstellt. Es sollte darauf geachtet werden, dass solche Widersprüche vermieden werden oder eine Lösungsmöglichkeit für diese Konflikte (zum Beispiel durch Prioritäten) existiert.

### - Änderungsanzeichen

Zum Ende des Jahres 2016 kam es zu zwei Vorfällen, die als mögliche Änderungsanzeichen angesehen werden können. Zum einen hat Adobe im Dezember die TIFF6.0 Spezifikation von ihrer Webseite ohne Vorankündigung entfernt. Zum anderen gab es großen Wirbel in der freien Softwarecommunity als die LibTIFF ihre Domain verlor. Dies ist bedeutsam, da nahezu jede Software, die mit TIFF-Dateien umzugehen weiß auf die LibTIFF aufsetzt. Auch wenn die LibTIFF mittlerweile neu unter der Adresse <http://libtiff.maptools.org/> firmiert, saß der Schock tief, da dies seit Bestehen des Projektes bereits der zweite Fall ist (sh. <https://madfileformatscience.garymcgath.com/2016/09/08/libtiff-offline/> ).

## TIFF-Varianten

Für die Langzeitarchivierung ist es empfehlenswert grundsätzlich nur die TIFFs zuzulassen, die dem baseline-Profil entsprechen und allenfalls einige weitere, den zu erwartenden Nutzungsszenarien entsprechenden, Tags zuzulassen.

Sinnvolle zusätzliche Tags sind:

- ICC-Profile (0x8773)
- EXIF-IFD (0x8769), welches auf ein IFD zeigt, dass weitere EXIF-Tags enthält
- XMP-Metadaten (0x02bc), für XMP-Metadaten im RDF-Format
- GeoTIFF-Tags

Daneben gibt es noch TIFF-Subprofile, die durch Standards näher spezifiziert sind, z.B.:

- TIFF/EP, Electronic Photography, ISO 12234-2:2001
- TIFF/IT, Image Technology (veraltet)
- TIFF/Fax <https://www.ietf.org/rfc/rfc2301.txt>
- TI/A <https://www.ti-a.org>
- Adobe DNG [http://www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng\\_spec\\_1.4.0.0.pdf](http://www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng_spec_1.4.0.0.pdf)

## TIFF-ähnliche Datenformate

Da TIFF eine sehr flexible Grundstruktur hat, wird diese auch für andere Datenformate eingesetzt.

Im Folgenden eine kurze Auswahl:

- BigTIFF für Bilder mit Dateigrößen > 4GB
- Panasonic Raw/RW2
- JPEG XR
- Navy Image FileFormat
- Olympus ORF
- Microsoft Document Image.

## TIFF-Validierung

Für die TIFF-Validierung ist es notwendig, das genaue Subprofil zu kennen. Andernfalls läuft man in Gefahr, dass TIFFs validiert werden, die nicht den eigenen Anforderungen entsprechen.

Hier eine Kurzübersicht der gängigen Programme:

- **JHove** Quelle: <http://jhove.openpreservation.org/>
  - validiert eine breite Palette von Dateiformaten
  - gute Unterstützung von TIFF
  - Ergebnis als XML
  - programmiert in Java, quelloffen, plattformunabhängig, GUI- & Kommandozeilenprogramm
- **checkit\_tiff** Quelle: [https://github.com/SLUB-digitalpreservation/checkit\\_tiff](https://github.com/SLUB-digitalpreservation/checkit_tiff)
  - validiert baseline-TIFFs mit Erweiterungen gegen selbstdefinierte Profile,
  - gängige Profile werden mitgeliefert
  - sehr schnell (> 25 Dateien/s), Ergebnis als Textausgabe und via Returncode
  - programmiert in C, quelloffen, plattformunabhängig, Kommandozeilenprogramm
- **DPFManager** Quelle: <http://dpfmanager.org/>
  - validiert baseline TIFF, TIFF/EP und TIFF/IT
  - gute GUI, quelloffen im Rahmen des EU Projektes preforma entwickelt
  - Ergebnis als XML- oder HTML-Report
  - programmiert in Java, plattformunabhängig, GUI- & Kommandozeilenprogramm

## TIFF-Reparatur

Vor jeglichen Reparaturen sollte versucht werden, die Ursache für die Problemfälle zu finden und abzustellen. Reparaturen sind immer ein Eingriff in das Original und sollten daher das letzte Mittel sein. Reparaturversuche gehören in der Regel dokumentiert. Alle Reparaturversuche müssen unter dem Gesichtspunkt der Quellenkritik betrachtet werden.

Im Folgenden wird nur eine Kurzübersicht der Reparaturwerkzeuge angegeben:

- **fixit\_tiff** Quelle: [https://github.com/SLUB-digitalpreservation/fixit\\_tiff](https://github.com/SLUB-digitalpreservation/fixit_tiff)
  - einfache ICC-Farbprofil-Korrektur
  - Korrektur DateTime-Tags
  - Korrektur falscher Reihenfolge TIFF-Tags
  - Löschen von Tags
  - Korrigieren von falschen Datentypen für ausgewählte Tags (zB. exifIFD, richtiff IPTC)
- **tiffset** (Teil der libtiff-Tools)
  - Neusetzen von Tags
- **exiftool** Quelle: <http://www.sno.phy.queensu.ca/~phil/exiftool/>
  - Löschen von Tags
  - Neusetzen von Tags
  - Extrahieren von Tag-Informationen (zB. ICC-Farbprofilen)
  - Hinzufügen von ICC-Farbprofilen
- **tiffcp** (Teil der libtiff-tools)
  - Dekompression
- **ImageMagick** (convert)
  - Colormap umwandeln in RGB

## Weiterführende Information

### Informationen zu TIFF

- Aufbau baseline TIFF: <http://art1pirat.blogspot.de/2013/07/baseline-tiff.html>
- Cheat-Sheet TIFF-Datei, Ange Albertini:
  - <http://i.imgur.com/ZioGqsa.png>
  - <http://i.imgur.com/EtssMll.png>
- TIFF Spezifikationen:
  - TIFF 6.0 <http://web.archive.org/web/20161101052612/http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
  - <http://web.archive.org/web/20161101052612/http://partners.adobe.com/public/developer/en/tiff/TIFFPM6.pdf>
  - <http://web.archive.org/web/20161101052612/http://partners.adobe.com/public/developer/en/tiff/TIFFphotoshop.pdf>

### TIFF nahe bzw. ergänzende Spezifikationen

- GeoTIFF: <http://web.archive.org/web/20160407195617/http://www.remotesensing.org/geotiff/spec/contents.html>
- DNG: [http://www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng\\_spec\\_1.4.0.0.pdf](http://www.images.adobe.com/content/dam/Adobe/en/products/photoshop/pdfs/dng_spec_1.4.0.0.pdf)
- TIFF/Internet Fax <https://www.ietf.org/rfc/rfc2301.txt>
- TI/A Initiative für "archivtaugliches" TIFF <http://www.ti-a.org>
- IPTC: <https://www.iptc.org/std/IIM/4.1/specification/IIMV4.1.pdf>
- ICC: <http://www.color.org/iccprofile.xalter>
- BigTIFF <http://simplesystems.org/libtiff/bigtifftiffdesign.html>

Andreas Romeyke, Mitarbeiter an der Sächsischen Staatsbibliothek, Landes- und Universitätsbibliothek Dresden (SLUB). Er beschäftigt sich mit der Analyse von Dateiformaten und Workflows im Rahmen der digitalen Langzeitarchivierung. Mit seinem Kollegen Jörg Sachse schreibt er regelmäßig zum Thema digitale Langzeitarchivierung auf dem gemeinsamen Blog "Kulturreste" unter <https://kulturreste.blogspot.de> ,  
Kontakt: [romeyke@slub-dresden.de](mailto:romeyke@slub-dresden.de), Webseite: <http://andreas-romeyke.de>

Weitere Kurzartikel aus der Reihe „nestor Thema“ finden Sie auf [www.langzeitarchivierung.de](http://www.langzeitarchivierung.de) -  
der Webseite von **nestor – Kompetenznetzwerk Langzeitarchivierung**.