TIFF Metadata Extraction Errors

The following is a guide to some of the metadata extraction errors you may encounter when ingesting TIFF files. When getting these errors, the first step is to ensure that the TIFF displays. If the TIFF does not display, you will have to replace the TIFF.

For general information about the TIFF file format, please refer to:

Aware Systems TIFF FAQ: https://www.awaresystems.be/imaging/tiff/faq.html

Library of Congress's TIFF sustainability page: https://www.loc.gov/preservation/digital/formats/fdd/fdd000022.shtml

Just Solve the File Format Problem's TIFF page: http://fileformats.archiveteam.org/wiki/TIFF

Issue Risk Rating Definitions

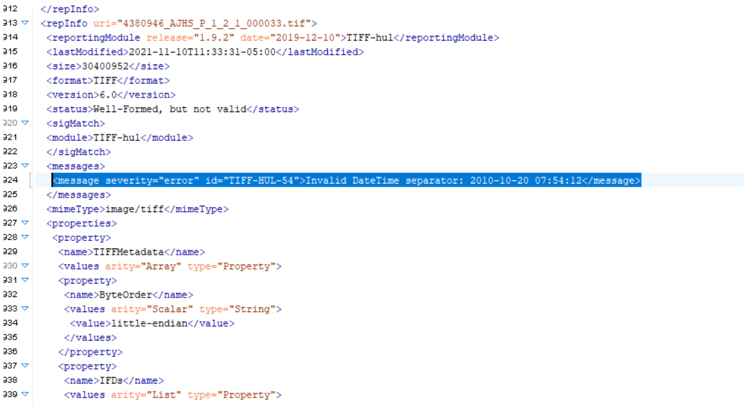
- High –Severe issue, such as data loss or unreadable files
- Middle – Violations of a standard's structure
- Low – Issues that do not violate a standard's structure

Invalid DateTime separator

Description

DateTime refers to the date and time of image creation. In the specification, DateTime format is: `"YYYY:MM:DD HH:MM:SS", with hours on a 24-hour clock, and one space character between the date and the time.`

A TIFF with this issue will have an error message like this in JHOVE’s output:



```
</repInfo>
<repInfo uri="4380946_AJHS_P_1_2_1_000033.tif">
 <reportingModule release="1.9.2" date="2019-12-10">TIFF-hul</reportingModule>
 <lastModified>2021-11-10T11:33:31-05:00</lastModified>
 <size>30400952</size>
 <format>TIFF</format>
 <version>6.0</version>
 <status>Well-Formed, but not valid</status>
 <sigMatch>
 <module>TIFF-hul</module>
 </sigMatch>
 <messages>
  <message severity="error" id="TIFF-HUL-54">Invalid DateTime separator: 2010-10-20 07:54:12</message>
 </messages>
 <mimeType>image/tiff</mimeType>
 <properties>
  <property>
   <name>TIFFMetadata</name>
   <values arity="Array" type="Property">
   <property>
    <name>ByteOrder</name>
    <values arity="Scalar" type="String">
     <value>little-endian</value>
    </values>
   </property>
   <property>
    <name>IFDs</name>
    <values arity="List" type="Property">
```

In this case, the DateTime string is formatted as YYYY-MM-DD HH:MM:SS, using hyphens as the date separator instead of the colons specified in the standard.

TiffDump confirms the invalid DateTime value:

```
4380915_AJHS_P_1_2_1_000002.tif:
Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 29571084 (0x1c3380c) next 0 (0)
SubFileType (254) LONG (4) 1<0>
ImageWidth (256) SHORT (3) 1<3825>
ImageLength (257) SHORT (3) 1<2577>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<1>
Photometric (262) SHORT (3) 1<2>
StripOffsets (273) LONG (4) 235<8 126233 252458 378683 504908
631133 757358 883583 1009808 1136033 1262258 1388483 1514708
1640933 1767158 1893383 2019608 2145833 2272058 2398283 2524508
2650733 2776958 2903183 ...>
Orientation (274) SHORT (3) 1<1>
SamplesPerPixel (277) SHORT (3) 1<3>
RowsPerStrip (278) SHORT (3) 1<11>
StripByteCounts (279) LONG (4) 235<126225 126225 126225 126225
126225 126225 126225 126225 126225 126225 126225 126225 126225
126225 126225 126225 126225 126225 126225 126225 126225 126225
126225 126225 ...>
XResolution (282) RATIONAL (5) 1<300>
YResolution (283) RATIONAL (5) 1<300>
PlanarConfig (284) SHORT (3) 1<1>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
DateTime (306) ASCII (2) 20<2010-10-20 07:51:22\0>
Artist (315) ASCII (2) 22<Zeutschel Omniscan 11\0>
SampleFormat (339) SHORT (3) 3<1 1 1>
```

```
700 (0x2bc) BYTE (1) 4878<0x3c 0x3f 0x78 0x70 0x61 0x63 0x6b 0x65
0x74 0x20 0x62 0x65 0x67 0x69 0x6e 0x3d 0x22 0xef 0xbb 0xbf 0x22
0x20 0x69 0x64 ...>
ICC Profile (34675) UNDEFINED (7) 303628<00 0x4 0xa2 0xc 0x6c
0x63 0x6d 0x73 0x2 00 00 00 0x73 0x63 0x6e 0x72 0x52 0x47 0x42
0x20 0x4c 0x61 0x62 0x20 ...>
```

Issue Risk Rating

Low: While separating the dates with anything other than colons is a violation of the standard, it does not affect the TIFF on a structural level. JHOVE deems TIFFS with this error “Well-Formed, but not valid.” Viewers can still render the TIFFS and tools can still work with the TIFFS and extract metadata.

Impact

High: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, 18,054 TIFFS had this error.

Possible Fix

Fixit_Tiff, a command line tool that fixes some issues in TIFFs, can be used to fix DateTime errors.

The command

```
$ fixit_tiff -i original.tif -o repaired.tif
```

Creates a copy of the TIFF file with a valid DateTime field, with all other image data remaining untouched.

TiffDump confirms that fixit_tiff repaired the DateTime field while copying over all other image data:

```
$ tiffdump 4380915_AJHS_P_1_2_1_000002-repaireddatetime.tif

4380915_AJHS_P_1_2_1_000002-repaireddatetime.tif:
Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 29571084 (0x1c3380c) next 0 (0)
SubFileType (254) LONG (4) 1<0>
ImageWidth (256) SHORT (3) 1<3825>
ImageLength (257) SHORT (3) 1<2577>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<1>
Photometric (262) SHORT (3) 1<2>
StripOffsets (273) LONG (4) 235<8 126233 252458 378683 504908
631133 757358 883583 1009808 1136033 1262258 1388483 1514708
1640933 1767158 1893383 2019608 2145833 2272058 2398283 2524508
2650733 2776958 2903183 ...>
Orientation (274) SHORT (3) 1<1>
SamplesPerPixel (277) SHORT (3) 1<3>
```

```
RowsPerStrip (278) SHORT (3) 1<11>
StripByteCounts (279) LONG (4) 235<126225 126225 126225 126225
126225 126225 126225 126225 126225 126225 126225 126225 126225
126225 126225 126225 126225 126225 126225 126225 126225 126225
126225 126225 ...>
XResolution (282) RATIONAL (5) 1<300>
YResolution (283) RATIONAL (5) 1<300>
PlanarConfig (284) SHORT (3) 1<1>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
DateTime (306) ASCII (2) 20<2010:10:20 07:51:22\0>
Artist (315) ASCII (2) 22<Zeutschel Omniscan 11\0>
SampleFormat (339) SHORT (3) 3<1 1 1>
700 (0x2bc) BYTE (1) 4878<0x3c 0x3f 0x78 0x70 0x61 0x63 0x6b 0x65
0x74 0x20 0x62 0x65 0x67 0x69 0x6e 0x3d 0x22 0xef 0xbb 0xbf 0x22
0x20 0x69 0x64 ...>
ICC Profile (34675) UNDEFINED (7) 303628<00 0x4 0xa2 0xc 0x6c
0x63 0x6d 0x73 0x2 00 00 00 0x73 0x63 0x6e 0x72 0x52 0x47 0x42
0x20 0x4c 0x61 0x62 0x20 ...>
```

Possible Fix's Risks

- Corruption of image data
  - Description: Image data can become corrupted in the copy process of altering the tags.
  - Mitigation: the tool Imagemagick can calculate a hash (like a MD5) for the image data instead of the entire file using the following command:

    ```
        $ identify -quiet -format "%#" image.tif
    ```

    This enables us to compare the image signature of the new and original files to ensure the image data is still the same.

- Updating the wrong images in Rosetta.
  - Description: Updating files in Rosetta is a manual process. Given the scale of this issue in our preservation environment, human error in replacing one file with another's repaired copy is likely.
  - Mitigation: Quality Control checks of updated representations.

- Cost may outweigh the benefit.
  - Description: Repairing and replacing the files may take an inordinate amount of time given the Low Issue Risk Rating
  - Mitigation: Keep a record of all the files currently in the preservation system with this issue and repair them gradually overtime. Check newly created TIFFs for this error and ensure TIFF creation software is creating structurally valid files.

Invalid Strip Offset

Description

TIFF's image data is organized into blocks of data called strips and tiles. Tag 273, stripoffset, points to where each strip's starting byte is in the file. It is like a map for whatever program is reading the file. Without a correct byte offset, the decoding program cannot find, much less read the TIFF's image data.

Mickey Lindlar from TIB wrote about a similar TIFF error here: https://openpreservation.org/blogs/troubles-with-tiff-stripoffsets-inconsistent-with-stripbytecounts/?q=109

Issue Risk Rating

High: In most cases, TIFFs with incorrect strip offsets cannot be rendered properly. They may render partially, or not render at all.

Impact

Low: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, only 17 TIFFs had this error.

Possible Fix

There is not a programmatic solution to fix TIFFs with this error. They will have to be recreated. In some cases, this error may have occurred because of a transfer error. In this case, it may be beneficial to see if the originally created TIFF has the same checksum and/or error as the TIFF that is flagged with this error. If the original TIFF has a different checksum and has no error, a TIFF with this error can be replaced with a re-transfer of the original file instead of being recreated/rescanned.

Premature EOF

Description

EOF refers to end of file. This error occurs when a TIFF has been truncated, possibly in transfer, and all of the TIFF's data is not included in the file.

Issue Risk Rating

High: In most cases, TIFFs with premature EOF errors cannot be rendered properly. They may render partially, or not render at all.

Impact

Low: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, only 40 TIFFs had this error.

Possible Fix

There is not a programmatic solution to fix TIFFs with this error. They will have to be recreated. In some cases, this error may have occurred because of a transfer error. In this instance, it may be beneficial to see if the originally created TIFF has the same checksum and/or error as the TIFF that is flagged with this error. If the original TIFF has a different checksum and has no error, a TIFF with this error can be replaced with a re-transfer of the original file instead of being recreated.

Tag out of sequence

Description

In a TIFF file, tags are information fields used to interpret image data. They include information like the width and the length of the TIFF, camera and software information, and the date the TIFF was created. The tags are arranged in a framework called an Image File Directory (IFD). The TIFF specification defines these tags and gives them a specific Code with a decimal and a hexadecimal value.

This error indicates that the tags in a TIFF file are not sorted in ascending numerical order. This is a violation of the TIFF specification, which requires that TIFF tags in an IFD must be sorted in an ascending order by their respective tag code. Using tiffdump, a tool the views TIFF tags, we can view how the tags are sorted.

In tiffdump, each line is a tag. The lines are arranged as so:

```
     Tag name |(Tag Code)| Tag’s Data Type|(Numerical Value of the
Tag’s DataType)| Data Count| Data
```

In a well-formed TIFF, the tiffdump output can look like this:

```
$ tiffdump ya-rg82-f2494-001.tif

ya-rg82-f2494-001.tif:

Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 8 (0x8) next 0 (0)
SubFileType (254) LONG (4) 1<0>
ImageWidth (256) LONG (4) 1<7024>
ImageLength (257) LONG (4) 1<4744>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<5>
Photometric (262) SHORT (3) 1<2>
Make (271) ASCII (2) 6<Canon\0>
Model (272) ASCII (2) 16<Canon EOS 5DS R\0>
StripOffsets (273) LONG (4) 297<22048 168834 316424 462996 610726
758176 906074 1054098 1202966 1352120 1501762 1651624 1801874
1952482 2102820 2251448 2403222 2560864 2717464 2874074 3032112
3190906 3352608 3516358 ...>
SamplesPerPixel (277) SHORT (3) 1<3>
RowsPerStrip (278) LONG (4) 1<16>
StripByteCounts (279) LONG (4) 297<146785 147589 146571 147730
147450 147897 148023 148868 149153 149641 149861 150250 150608
150337 148628 151773 157642 156599 156609 158038 158793 161702
163749 162817 ...>
XResolution (282) RATIONAL (5) 1<440>
YResolution (283) RATIONAL (5) 1<440>
PlanarConfig (284) SHORT (3) 1<1>
ResolutionUnit (296) SHORT (3) 1<2>
```

```
Software (305) ASCII (2) 49<Adobe Photoshop Lightroo ...>
DateTime (306) ASCII (2) 20<2021:06:22 17:11:24\0>
Artist (315) ASCII (2) 12<Saul Hankin\0>
Predictor (317) SHORT (3) 1<2>
700 (0x2bc) BYTE (1) 13244<0x3c 0x3f 0x78 0x70 0x61 0x63 0x6b
0x65 0x74 0x20 0x62 0x65 0x67 0x69 0x6e 0x3d 0x22 0xef 0xbb 0xbf
0x22 0x20 0x69 0x64 ...>
33723 (0x83bb) LONG (4) 48<5898524 1193614083 540 470024194
352327938 1685217608 1701994871 1126186272 1852796513 1396978976
419568722 1699482368 1025536878 1831875872 419568749 1867714560
1635218534 1025533298 1868841248 35415394 839385143 808530480
472920374 184564738 ...>
34377 (0x8649) BYTE (1) 4622<0x38 0x42 0x49 0x4d 0x4 0xc 00 00 00
00 0x11 0xe5 00 00 00 0x1 00 00 0x1 00 00 00 00 0xad ...>
34665 (0x8769) LONG (4) 1<21434>
ICC Profile (34675) UNDEFINED (7) 560<00 00 0x2 0x30 0x41 0x44
0x42 0x45 0x2 0x10 00 00 0x6d 0x6e 0x74 0x72 0x52 0x47 0x42 0x20
0x58 0x59 0x5a 0x20 ...>
```

In the above example, all the TIFF’s tags are presented in ascending numerical order via the tag code. The tiffdump output reflects the order the TIFF tags are sequenced in the file.

In a TIFF with an out of sequence error, the tiffdump output looks like this:

```
$ tiffdump 5212791_AJHS_P_24_1_7_000092.tif

5212791_AJHS_P_24_1_7_000092.tif:
Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 8 (0x8) next 0 (0)
ImageWidth (256) LONG (4) 1<3833>
ImageLength (257) LONG (4) 1<2961>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<1>
Photometric (262) SHORT (3) 1<2>
StripOffsets (273) LONG (4) 2961<24032 35532 47032 58532 70032
81532 93032 104532 116032 127532 139032 150532 162032 173532
185032 196532 208032 219532 231032 242532 254032 265532 277032
288532 ...>
SamplesPerPixel (277) SHORT (3) 1<3>
RowsPerStrip (278) LONG (4) 1<1>
StripByteCounts (279) LONG (4) 2961<11499 11499 11499 11499 11499
11499 11499 11499 11499 11499 11499 11499 11499 11499 11499 11499
11499 11499 11499 11499 11499 11499 11499 11499 ...>
XResolution (282) RATIONAL (5) 1<300>
YResolution (283) RATIONAL (5) 1<300>
PlanarConfig (284) SHORT (3) 1<1>
ResolutionUnit (296) SHORT (3) 1<2>
Software (305) ASCII (2) 32<LIBFORMAT (c) Pierre-e G ...>
```

```
Predictor (317) SHORT (3) 1<1>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
DateTime (306) ASCII (2) 20<2010:05:06 13:12:13\0>
Artist (315) ASCII (2) 22<Zeutschel Omniscan 11\0>
```

In this example, all tags up to the first 305 tag are sorted in ascending order (tag 256 to 257 to 248, etc). After the first 305 tag, we see a 317 tag, followed by a another 305, a 306, and a 315.

Mickey Lindlar wrote about this error here: https://openpreservation.org/blogs/troubles-with-tiff-tag-270-out-of-sequence/Issue Risk Rating

Medium: The specification states: `The entries in an IFD must be sorted in ascending order by Tag.` This error is a violation of the TIFF standard on a structural level. However, viewers are still able to render these files correctly.

In the future, it may cause issues with extracting technical metadata, or with viewers that rely on a sequential structure of tags, but that is not the case with the viewers and tools we are using in our current preservation and delivery environment. Rosetta's tools can extract the technical metadata and place it in the files DNX, as well as render the files.

Impact

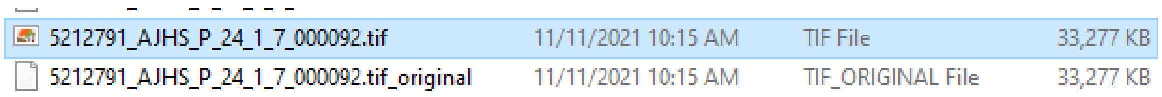
High: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, 35903 TIFFS had this error.

Possible Fix

Exiftool, a tool for reading, writing, and editing file metadata, can copy TIFFs' tags into a new file in the corrects sequence. The command for this would be:

```
exiftool -P -Software= -tagsfromfile @ -Software image.tif
```

A new file is created with the same filename as the original TIFF, and the original TIFF is renamed filename.tif_original. The tags' data stays the same, but they are reordered in the correct sequence. The -P flag retains the filesystem dates and times:



5212791_AJHS_P_24_1_7_000092.tif	11/11/2021 10:15 AM	TIF File	33,277 KB
5212791_AJHS_P_24_1_7_000092.tif_original	11/11/2021 10:15 AM	TIF_ORIGINAL File	33,277 KB

Running tiffdump shows that the tags are now in the correct sequence.

```
$ tiffdump 5212791_AJHS_P_24_1_7_000092.tif

5212791_AJHS_P_24_1_7_000092.tif:
Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 8 (0x8) next 0 (0)
ImageWidth (256) LONG (4) 1<3833>
ImageLength (257) LONG (4) 1<2961>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<1>
Photometric (262) SHORT (3) 1<2>
```

```
StripOffsets (273) LONG (4) 2961<24018 35518 47018 58518 70018 81518
93018 104518 116018 127518 139018 150518 162018 173518 185018 196518
208018 219518 231018 242518 254018 265518 277018 288518 ...>
SamplesPerPixel (277) SHORT (3) 1<3>
RowsPerStrip (278) LONG (4) 1<1>
StripByteCounts (279) LONG (4) 2961<11499 11499 11499 11499 11499
11499 11499 11499 11499 11499 11499 11499 11499 11499 11499 11499
11499 11499 11499 11499 11499 11499 11499 11499 ...>
XResolution (282) RATIONAL (5) 1<300>
YResolution (283) RATIONAL (5) 1<300>
PlanarConfig (284) SHORT (3) 1<1>
ResolutionUnit (296) SHORT (3) 1<2>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
DateTime (306) ASCII (2) 20<2010:05:06 13:12:13\0>
Artist (315) ASCII (2) 22<Zeutschel Omniscan 11\0>
Predictor (317) SHORT (3) 1<1>
```

Possible Fix's Risks

- Possible corruption of image data
  - Description: Image data can become corrupted in the copy process of altering the tags.
  - Mitigation: the tool Imagemagick can calculate a hash (like a MD5) for the image data instead of the entire file using the following command:

    ```
    $ identify -quiet -format "%#" image.tif
    ```

    This enables us to compare the image signature of the new and original files to ensure the image data is still the same.

- Updating the wrong images in Rosetta
  - Description: Updating files in Rosetta is a manual process. Given the scale of this issue in our preservation environment, human error in replacing one file with another's repaired copy is likely.
  - Mitigation: Quality Control checks of updated representations.

- Cost may outweigh the benefit.
  - Description: With this many TIFFS with the same error, repairing and replacing the files may take an inordinate amount of time given the Risk Rating
  - Mitigation: Keep a record of all the files currently in the preservation system with this issue and repair them gradually overtime. Check newly created TIFFs for this error and ensure TIFF creation software is creating structurally valid files.

Type Mismatch

Description

In the Image File Directory (IFD), tags' data types are listed along with their Tag IDs. A data type is a number signifying the way the tag's data is expressed. There are 12 data types that a tag can use. They are as follows:

Numerical Value	Data Type Name	Data Type Definition
1	BYTE	and 8-bit number within the range of 0 to 255,
2	ASCII	an 8-bit character string
3	SHORT	16-bit number within the range of 0 to 65535
4	LONG	32-bit number within the range of 0 to 4,294,967,295
5	RATIONAL	two 32-bit numbers within the range of 0 to 4,294,967,295
6	SBYTE	an 8-bit number withing a range from -128 to 127
7	UNDEFINE	an 8-bit byte of data of any type
8	SSHORT	16-bit number within the range of -32,768 to 32,767
9	SLONG	32-bit number within the range of -2,147,483,648 to 2,147,483,647
10	SRATIONAL	two 32-bit numbers within the range of -2,147,483,648 to 2,147,483,647
11	FLOAT	4-byte single-precision IEEE floating point value
12	DOUBLE	8-byte double-precision IEEE floating-point value

Data types 1 through 5

Every tag is assigned to a specific data type. A type mismatch error occurs if a data type does not match the tags' assigned data type.

For example, an error that reads: Type mismatch for tag 34675; expecting 7, saw 1, means that Tag 34575, the ICC Profile tag, is assigned to type 7, the UNDEFINE type. However, the tag's data type reads 1, or the BYTE type.

Running tiffdump confirms the type mismatch:

```
5515455_AJHS_I_11_3_3_000024.tif:
Magic: 0x4949 <little-endian> Version: 0x2a <ClassicTIFF>
Directory 0: offset 8 (0x8) next 0 (0)
ImageWidth (256) LONG (4) 1<2791>
ImageLength (257) LONG (4) 1<3554>
BitsPerSample (258) SHORT (3) 3<8 8 8>
Compression (259) SHORT (3) 1<1>
Photometric (262) SHORT (3) 1<2>
StripOffsets (273) LONG (4) 3554<332416 340790 349164 357538 365912 374286 382660 391034
399408 407782 416156 424530 432904 441278 449652 458026 466400 474774 483148 491522 499896
508270 516644 525018 ...>
SamplesPerPixel (277) SHORT (3) 1<3>
RowsPerStrip (278) LONG (4) 1<1>
StripByteCounts (279) LONG (4) 3554<8373 8373 8373 8373 8373 8373 8373 8373 8373 8373 8373 8373
8373 8373 8373 8373 8373 8373 8373 8373 8373 8373 8373 8373 ...>
XResolution (282) RATIONAL (5) 1<300>
YResolution (283) RATIONAL (5) 1<300>
PlanarConfig (284) SHORT (3) 1<1>
```

```
ResolutionUnit (296) SHORT (3) 1<2>
Software (305) ASCII (2) 32<LIBFORMAT (c) Pierre-e G ...>
Predictor (317) SHORT (3) 1<1>
ICC Profile (34675) BYTE (1) 303627<00 0x4 0xa2 0xb 0x6c 0x63 0x6d 0x73 0x2 0x30 00 00 0x73 0x63
0x6e 0x72 0x52 0x47 0x42 0x20 0x4c 0x61 0x62 0x20 ...>
Software (305) ASCII (2) 17<Certifi Pedigree\0>
DateTime (306) ASCII (2) 20<2010:07:02 15:14:40\0>
Artist (315) ASCII (2) 22<Zeutschel Omniscan 11\0>
```

Issue Risk Rating

Low: In all the errors of this type in our current collection, the tag was assigned to data type 7 in the TIFF standard, but the data type in the file was either a 1 or a 2. These data types are similar. A 1 (BYTE) data type is an 8-bit number, a 2 (ASCII) is an 8-bit character, and a 7(UNDEFINE) is 8-bits of any kind of data. Since the UNDEFINE data type can be 8-bits of anything, it could be an 8-bit number or an 8-bit character. Files with this error are still viewable.

However, there may be cases where the type mismatches are dissimilar, like a Tag that is assigned to a number data type like 3 (SHORT) listing a character data type, 2 (ASCII). I have not encountered a type mismatch like this so far. We will evaluate the risk rating for dissimilar tag mismatches if we encounter an error like that in the future.

Impact

Medium: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, 2440 TIFFs had this error.

Possible Fix

N/A

Action

These errors can be ignored if the data types are similar, and the TIFF displays.

Value offset not word-aligned

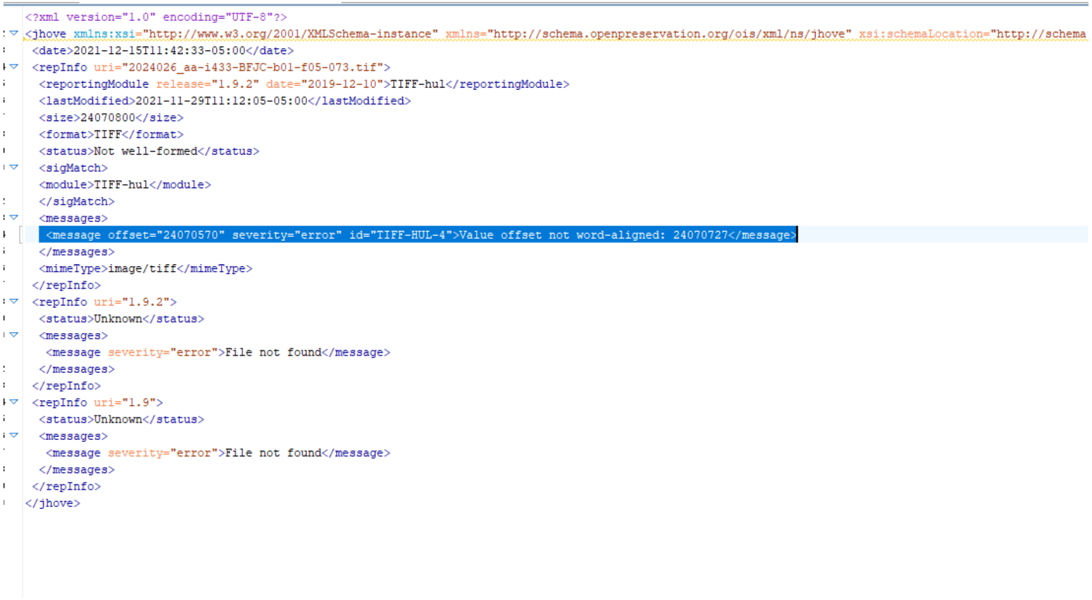
Description

This issue seems to be caused by an issue with Adobe Photoshop's handling of some larger EXIF files. The issue was written up by the NYU Library, Phase One, and Digital Transitions, as noted on Adobe's feedback site here: https://web.archive.org/web/20211215191442/https://feedback-readonly.photoshop.com/conversations/photoshop/some-tiff-files-saved-from-photoshop-are-not-wellformed-in-jhove-validation/5f5f45df4b561a3d4264642d

To quote the feedback message linked above: "If an EXIF tag is larger than four bytes, and the byte count is not a multiple of two, then it must be stored on an offset of the file with padding. This ensures the byte count is always a multiple of two. Photoshop eliminates the padding, which causes these tags to fall out of compliance with the TIFF specification."

Adobe fixed this error in February 2019 (https://web.archive.org/web/20211215192048/https://heritage-digitaltransitions.com/adobe-fixes-bug-uncovered-by-jhove/)

A TIFF with this error will have a message like this



```
<?xml version="1.0" encoding="UTF-8"?>
<jhove xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schema.openpreservation.org/ois/xml/ns/jhove" xsi:schemaLocation="http://schema
 <date>2021-12-15T11:42:33-05:00</date>
 <repInfo uri="2024026_aa-i433-BFJC-b01-f05-073.tif">
  <reportingModule release="1.9.2" date="2019-12-10">TIFF-hul</reportingModule>
  <lastModified>2021-11-29T11:12:05-05:00</lastModified>
  <size>24070800</size>
  <format>TIFF</format>
  <status>Not well-formed</status>
  <sigMatch>
  <module>TIFF-hul</module>
  </sigMatch>
  <messages>
   <message offset="24070570" severity="error" id="TIFF-HUL-4">Value offset not word-aligned: 24070727</message>
  </messages>
  <mimeType>image/tiff</mimeType>
 </repInfo>
 <repInfo uri="1.9.2">
  <status>Unknown</status>
  <messages>
   <message severity="error">File not found</message>
  </messages>
 </repInfo>
 <repInfo uri="1.9">
  <status>Unknown</status>
  <messages>
   <message severity="error">File not found</message>
  </messages>
 </repInfo>
</jhove>
```

It appears that this error also prevents JHOVE from extracting any technical metadata.

Issue Risk Rating

Medium: While the image will still display in most viewers with this error, this is still a structural issue that prevents JHOVE from being able to extract any technical metadata, preventing the metadata from being read and indexed by our preservation system, Rosetta.

Impact

Low: In a survey of 45212 TIFFs that were flagged with metadata extraction errors, only 43 TIFFS had this error.

Possible Fix

Exiftool, a tool for reading, writing, and editing file metadata, can copy TIFFs' tags into a new file that corrects the padding issue. The command for this would be:

```
exiftool -P - Software= -tagsfromfile @ - Software image.tif
```

A new file is created with the same filename as the original TIFF, and the original TIFF is renamed filename.tif_original. The metadata stays the same, but the tags' data is padded correctly. The -P flag retains the filesystem dates and times:

Possible Fix's Risks

- Possible corruption of image data
    - Description: Image data can become corrupted in the copy process of altering the tags.
    - Mitigation: the tool Imagemagick can calculate a hash (like a MD5) for the image data instead of the entire file using the following command:

      ```
      $ identify -quiet -format "%#" image.tif
      ```

      This enables us to compare the image signature of the new and original files to ensure the image data is still the same.

- Updating the wrong images in Rosetta
    - Description: Updating files in Rosetta is a manual process. Human error like incorrectly replacing a file is a possibility.
    - Mitigation: Quality Control checks of updated representations.