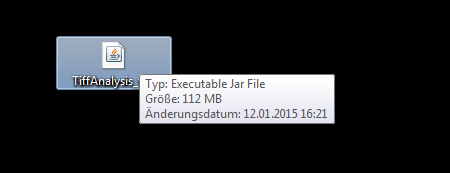
# Tiff Analysis

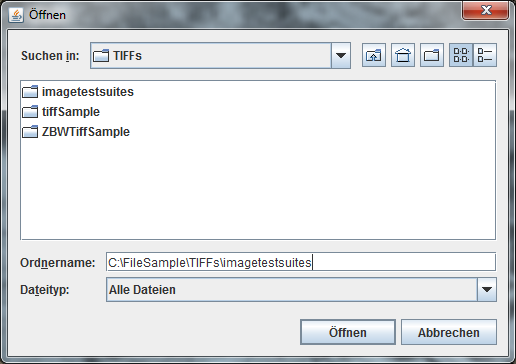
## Documentation for the user

Currently, you only need to download (and maybe unzip) the jar-file and then double-click on the jar.



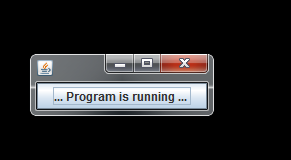
It’s java and platform-agnostic.

Next, a Folder Browser Dialog opens. Navigate to the folder you want to examine.



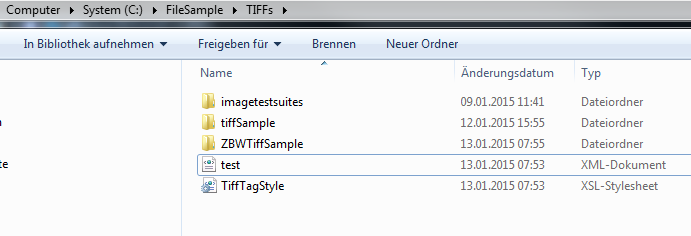
Following a Dialog opens and you are asked to name your xml Outputfile. It can be “outputXML”, “Test”, “auntDebbie” or whatever. The .xml-Extension will be added automatically and the File will be put in the same folder you want to examine. If the filename already exists in the folder, you are asked if you want to stop and try again (“Ja”) or overwrite the file and continue (“Nein”). The program is not a 100% English yet.

You see that the program is running:



This window might be at the corner of your monitor up in the left side. It will vanish when the program is done.

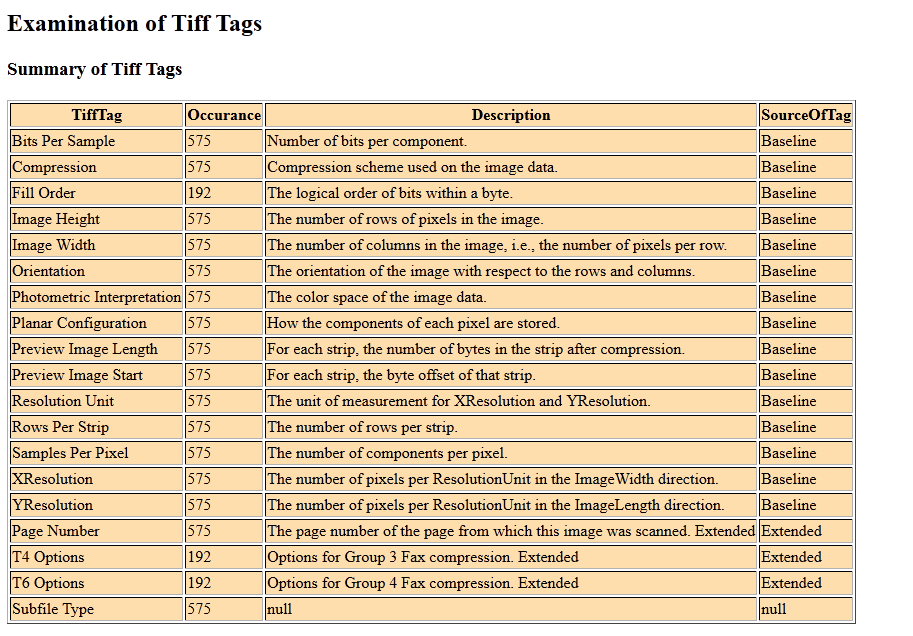
You will find two new files in your examined folder now:



Go to your favorite Browser (e. g. firefox, I think there are issues with the internet explorer) and open the xml-file there, in this case the path would be: <file:///C:/FileSample/TIFFs/ZBWTiffSample/test.xml>

A – hopefully – nicely looking page with many tables will open.

### Table 1: Summary of Tiff Tags



You can see the names of the Tiff Tags – which are usually known and can be displayed. The description, though, can only be seen if I personally have put it into the program, which I have not done with nearly all the possible Tiff Tags yet. The same is true for the SourceOfTag, but here, at least, it is known if a tag is private, if the dec value is higher than a certain value (32768), so at least this should always be displayed.

If an information is not there, it’s “null”. Null means no more than “I do not have any clue”.

SubFileType, however, is very baseline and should be known to the program. In the next release, this might show the description: "A general indication of the kind of data contained in this subfile."

### Table 2: Tiff Statistic

This table is short and just shows how many Tiff files are in the folder and how many of them could be examined. Some are so difficult, that the program cannot get the metadata about the Tiff Tags out of them.

Other files, however, are ignored. The program deals with the “.tif”-Extension (not context-sensitive) and with the file header, which has to start with “II” or “MM”.

### Table 3: Examination Failed for following Tiff Files

All Tiff-Files that could not be examined are displayed by file path and the error message is shown.

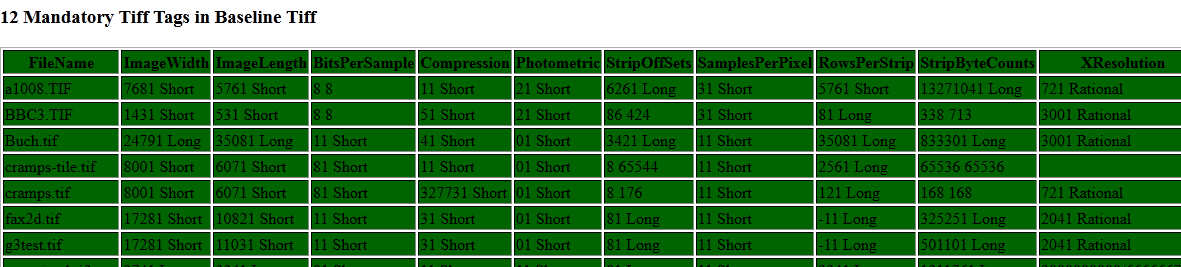


One day, I might deal with them and look for ways to fix them, as there are some tools out there and Tiff should (in some cases) not be too hard to fix.

### Table 4: Mandatory Tiff Tags in Baseline Tiff

These Tags should be there. Sometimes they should even have a certain value (not all of the compression types are allowed or good for longterm archiving), but I have not dealt with that yet.

*YResolution* and *ResolutionUnit* are not seen on this screenshot, because the table is too wide, but they are there.



### Table 5: Nice to have Tiff Tags optional in Baseline Tiff

These are (for now)

* MinSampleValued
* MaxSampleValue
* GrayResponseUnit
* GrayResponseCurve
* Software
* DateTime
* Artist
* ColorMap
* Copyright

### Table 6: No Tags for Baseline Tiff but nice to have for Digital Preservation

ExifIdf & ExtraSamples

### Table 7: Other known Tiff Tags

Only a few I have come across already and the number might grow a bit in the near future.

The others are only seen in the Table 1 and of course, if you examine the XML-Output in an editor, you will have all the talkative information.

<DateTime>'2005:01:24 11:05:43'20 ASCII</DateTime>

<Predictor>21 Short</Predictor>

<ExtraSamples>01 Short</ExtraSamples>

<UnknownTiffTag>Application Notes</UnknownTiffTag>

<IPTC\_NAA>469893120 33554944</IPTC\_NAA>

<UnknownTiffTag>IPTC- NAA</UnknownTiffTag>

<UnknownTiffTag>Photoshop Settings</UnknownTiffTag>

<UnknownTiffTag>Photoshop Settings</UnknownTiffTag>

<ExifIfd>24540921 Long</ExifIfd>

<UnknownTiffTag>Exif Offset</UnknownTiffTag>

<InterColorProfile>0 0</InterColorProfile>

<UnknownTiffTag>ICC\_ Profile</UnknownTiffTag>

I will enhance this soon that it outputs the contents of the unknown tags as well. There might be another table as well.

## Documentation for the developer

To be really short for now, I have used the org.apache.sanselan.formats.tiff.\* and org.apache.sanselan.common.IImageMetadata; libraries and the program has more or less only one heart:

IImageMetadata metadata = Sanselan.*getMetadata*(file);

TiffDirectory tiffDirectory = ((TiffImageMetadata) metadata).findDirectory(TiffDirectoryConstants.***DIRECTORY\_TYPE\_ROOT***);

ArrayList<TiffField> allEntries = tiffDirectory.getDirectoryEntrys();

tiffDirectory.dump();

The rest is just a lot of XML, XSL and running through folders to look for Tiff-Files.