Written by Uwe Noack
Reviewed by Juha Lintula

The ExifTool doesn't write Daminion custom tags by default. One has to customize the file ".ExifTool_config".
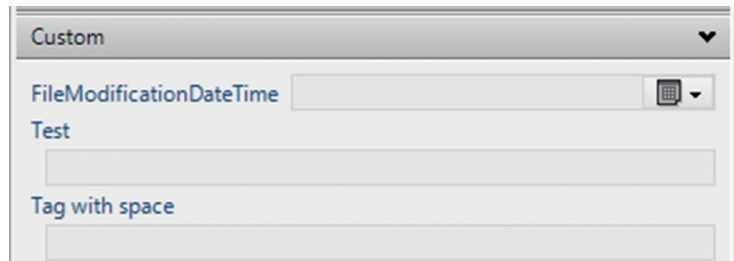This is the result of the first „try and error" activity. Juha made the first review of this document
and the example of the configuration file. Thank you very much, Juha. Maybe someone can make further suggestions for improvement or can add customizing for more and more complex custom tags.
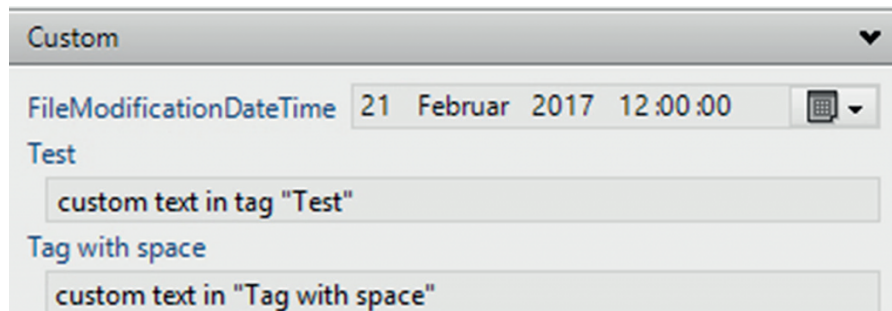

Sample image

Custom tags created in Daminion

Custom tags in the Properties panel

Example

There are three custom tags. The TIF file (above) is a screenshot and doesn't have any metadata. Below you can see the text/date information entered in Daminion.

Information entered in Daminion

This is the XMP section Daminion writes to the file.

XMP metadata written to the file by Daminion

1. Different from the definition and what one can see in the Daminion client the tags are written in lower case.

2. Each „space" of the tag name is substituted by „__" (double underscores).

Knowing this, one has to modify the file ".ExifTool_config". The file has to be in the folder of the ExifTool. For the Daminion Server this folder is: „C:\ProgramFiles(x86)\DaminionSoftware\DaminionServer\MediaProcessors\ImageProcessor\". Using other folders and other options please read the comment in the example: „http://www.sno.phy.queensu.ca/~phil/exiftool/config.html"

Below you can find an example of the file ".ExifTool_config" modified to work with these three custom tags defined in Daminion:

```
#
# Example how to configure Daminion Custom Tags in ExifTool
#
# replace the tag names with your custom tags
# copy paste this snippet into your ExifTool config file ".ExifTool_config"
#
```

add new XMP namespace 'daminion' to the main XMP table

```
%Image::ExifTool::UserDefined = (
  'Image::ExifTool::XMP::Main' => {
        daminion => { SubDirectory => { TagTable => 'Image::ExifTool::UserDefined::daminion', }, },
  },
);
```

see http://www.sno.phy.queensu.ca/~phil/exiftool/index.html #groups for Family 2 groups
data types are
– 'string' = Text (in Daminion)
– 'integer'
– 'real' = Fraction (in Daminion)
– 'date' = Time (in Daminion)
– 'boolean' = – (not supported in Daminion)
#
Daminion custom tags below default to string type variables in Family 2 group "Other"
%Image::ExifTool::UserDefined::daminion = (
GROUPS => { 0 => 'XMP', 1 => 'XMP-daminion', 2 => 'Other' },
NAMESPACE => { 'daminion' => 'http://ns.daminion.net/1.0/' },
WRITABLE => 'string',

add custom tags below
– all tag names are lower case letters
– spaces are subsituted with double underscores ' ' ==> '__'
#
Examples
customtag1 = { },   # default string type tag
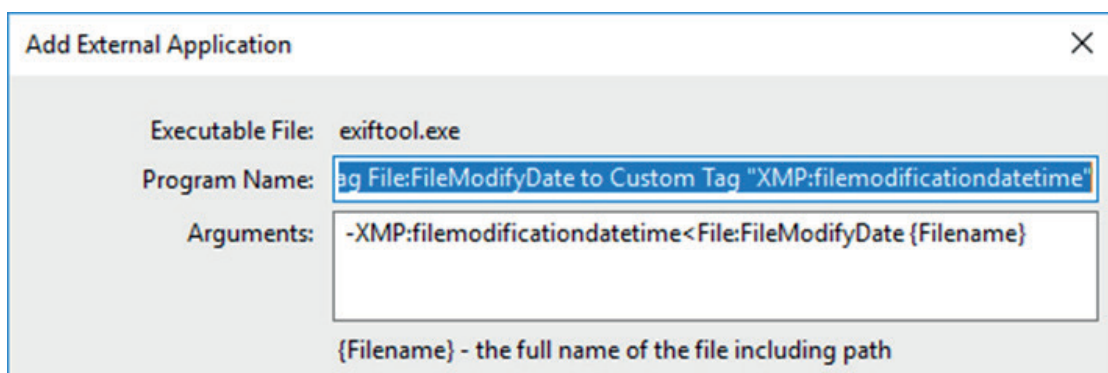# customtag2 = { Writable => 'integer' }, # change data type to integer
customtag3 = { Writable => 'date', Groups => { 2 => 'Time' } }, # data type date and family 2 'Time'
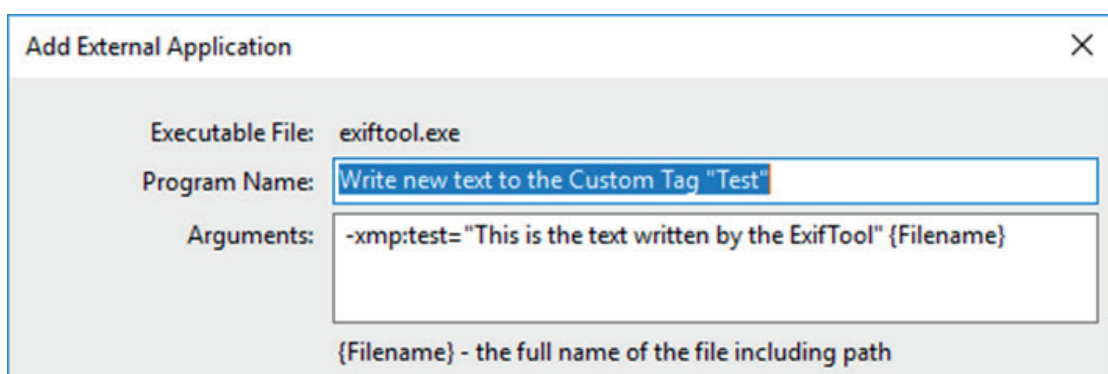# customtag4 = { List => 'Bag' }, # multivalue tag (like Keywords), can be also hierarchical
#

filemodificationdatetime => { Writable => 'date' },
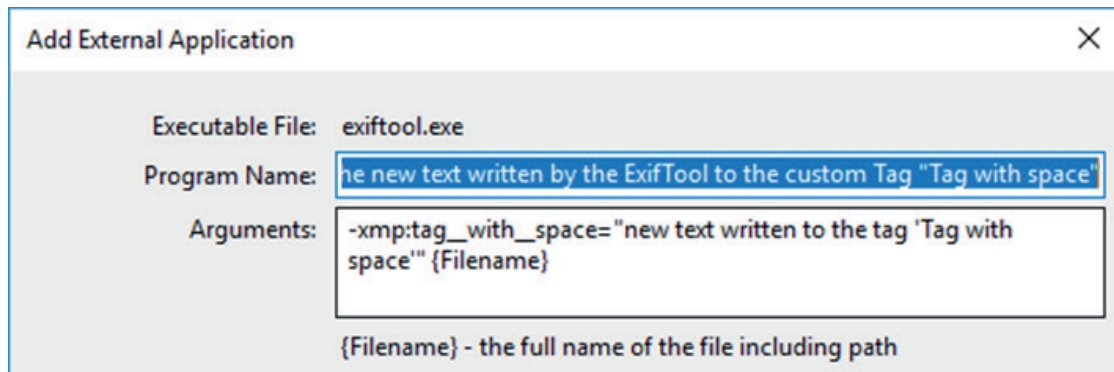test => { },
tag__with__space => { },
);

Here is the customizing of the „Open with…" action and the result:
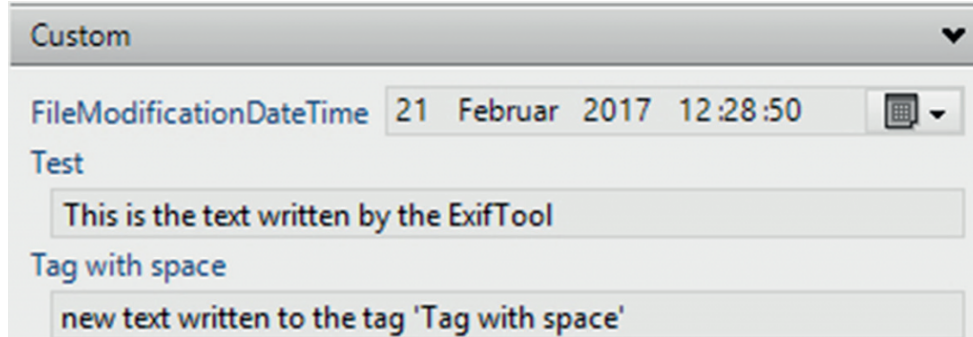

Write custom tag "FileModificationDateTime"

Write custom tag "Test"

Write custom tag "Tag with space"

The result in the Properties panel:

The result in the XMP section of metadata:

```
<x:xmpmeta xmlns:x='adobe:ns:meta/' x:xmptk='Image::ExifTool 10.43'>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>

 <rdf:Description rdf:about=''
  xmlns:daminion='http://ns.daminion.net/1.0/'>
  <daminion:filemodificationdatetime>2017-02-21T12:28:50+01:00</daminion:filemodificationdatetime>
  <daminion:tag__with__space>new text written to the tag &#39;Tag with space&#39;</daminion:tag__with__space>
  <daminion:test>This is the text written by the ExifTool</daminion:test>
 </rdf:Description>

 <rdf:Description rdf:about=''
  xmlns:xmp='http://ns.adobe.com/xap/1.0/'>
  <xmp:MetadataDate>2017-02-21T11:04:14+01:00</xmp:MetadataDate>
  <xmp:ModifyDate>2017-02-21T11:04:14+01:00</xmp:ModifyDate>
 </rdf:Description>
</rdf:RDF>
</x:xmpmeta>
```

If you have any comments or suggestions, please refer to this topic on the Daminion forum.