

Workaround for misused XLIFF



Angelika Zerfaß

zaac

XLIFF

XML Localization Interchange File Format

- What it is:
 - Format for translation
 - Bilingual setup
 - Original file format gets converted to XLIFF
 - Segmentation
- What it is NOT:
 - It is NOT supposed to contain other file formats, like HTML (as you can do with XML itself).
 - It is NOT format where you can customize attributes to contain user-defined comments, IDs or length information.
 - It is NOT format that gets segmented when importing it into a CAT tool.
 - The segmentation is supposed to happen when creating the XLIFF.

XLIFF

XML Localization Interchange File Format

- A lot of developers who create XLIFF don't seem to be aware of what XLIFF is supposed to be.
- We have to deal with "misused" XLIFF.
- Most of the time, you can use an XML filter instead 😊.
 - Copy source to target
 - Lock the content between the source tags.
 - Use the content between the target tags for translation.

1. XLIFF with HTML

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>  
<xliff xmlns="urn:oasis:names:tc:xliff:document:1.2" version="1.2">  
...
```

```
<source>
```

<![CDATA[This is some text for a website. The text is written in HTML and therefore contains several HTML-specific tags. For example **** formatting ****. In order to embed the text in XML the user has chosen the CDATA element (which signifies that everything that follows is pure text. Tags are not to be interpreted as such). It also contains several sentences within one source segment as the user has not done any segmentation while producing the XLIFF file. **<p> </p>
** Linebreaks are created through P and BR tags. And the user has also used an entity, here the non-breaking ** **space. **]]>**

```
</source>
```

```
<target> </target>
```

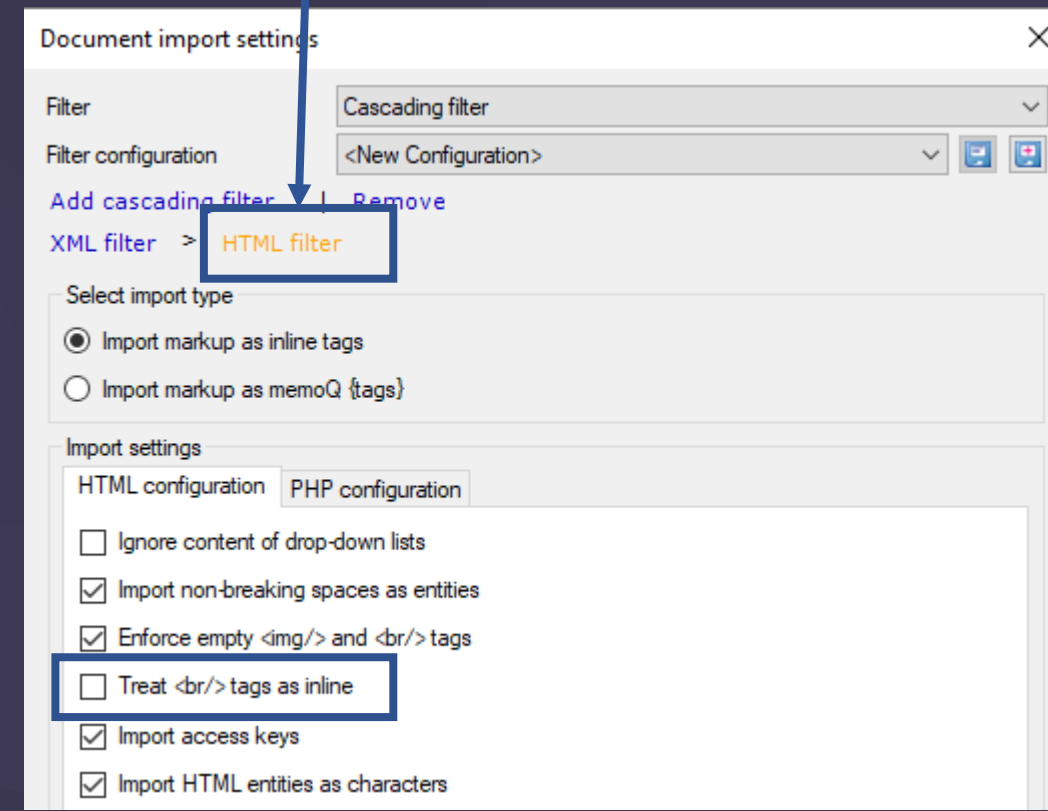
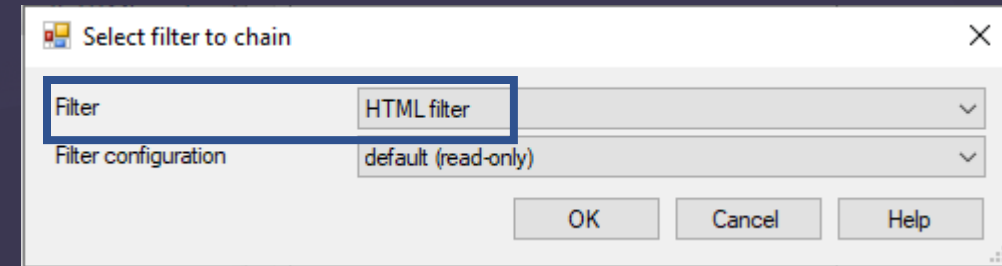
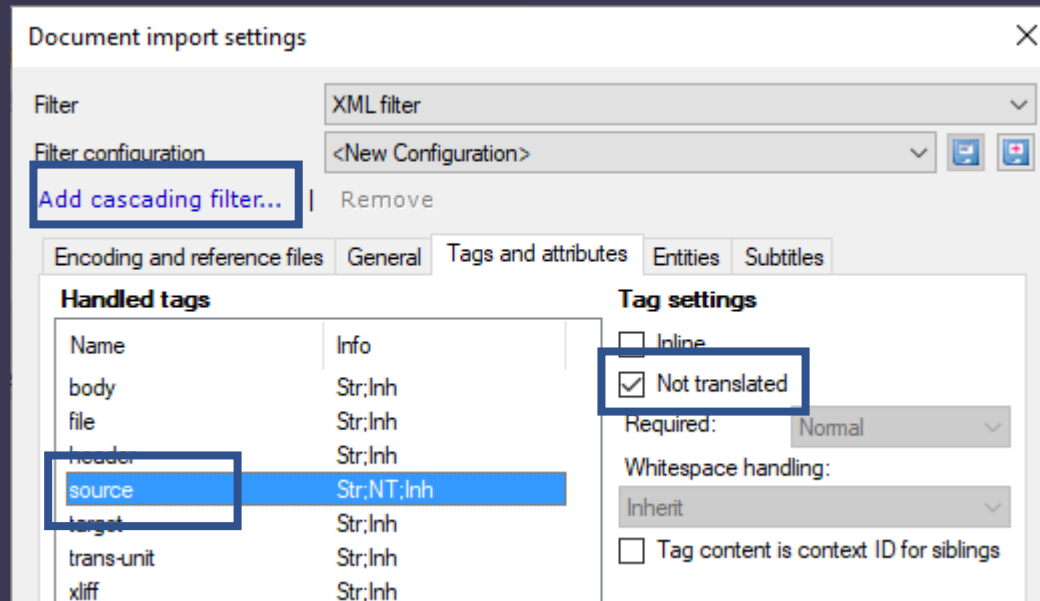
```
...  
</trans-unit> </body> </file> </xliff>
```

XLIFF Filter

Source	Target
1. This is some text for a website. The text is written in HTML and therefore contains several HTML-specific tags. For example <code></code> formatting <code></code> . In order to embed the text in XML the user has chosen the CDATA element (which signifies that everything that follows is pure text. Tags are not to be interpreted as such). It also contains several sentences within one source segment as the user has not done any segmentation while producing the XLIFF file. <code><p></p>
</code> Linebreaks are created through P and BR tags. And the user has also used an entity, here the non-breaking <code>&#xA0;</code> space.	

Importing as XML (+ HTML)

- Use an XML filter where you lock the content of the `<source>...</source>` sections, effectively importing the text between the `<target>...</target>` tags as source text.
 - Add a cascading HTML filter and define whether a `
` tag should break a segment or not.
- This text will get overwritten with your translation.



XML filter + HTML filter

XML filter + HTML filter

The screenshot displays a translation tool interface with a list of source and target text segments. The source text is in English and contains several HTML-specific tags and entities. The target text is in a different language (likely Chinese) and contains the same text but with some tags and entities removed or replaced. Annotations include blue arrows pointing from the source text to the target text, highlighting the removal of tags and entities. A blue box labeled "Used for segmentation" points to a specific part of the target text. The bottom pane shows the XML source and target code, with annotations highlighting the removal of tags and entities.

Source	Target	Progress
1. This is some text for a website.		0%
2. The text is written in HTML and therefore contains several HTML-specific tags.		0%
3. For example formatting .		0%
4. In order to embed the text in XML the user has chosen the CDATA element (which signifies that everything that follows is pure text.		0%
5. Tags are not to be interpreted as such).		0%
6. It also contains several sentences within one source segment as the user has not done any segmentation while producing the XLIFF file.		0%
7. Linebreaks are created through P and BR tags.		0%
8. And the user has also used an entity, here the non-breaking nbsp space.		0%

Used for segmentation

```
<trans-unit datatype="html" id="10-0">  
<source><![CDATA[This is some text for a website. The text is written in HTML and therefore contains several HTML-specific tags. For example  
<strong> formatting </strong> in order to embed the text in XML the user has chosen the CDATA element (which signifies that everything that follows is pure text.  
Tags are not to be interpreted as such). It also contains several sentences within one source segment as the user has not done any segmentation while producing the  
XLIFF file. <p></p><br /> Linebreaks are created through P and BR tags. And the user has also used an entity, here the non-breaking &#xA0; space.]]></source>  
<target><![CDATA[[This is some text for a website. The text is written in HTML and therefore contains several HTML-specific tags. For  
example <strong> formatting </strong>. In order to embed the text in XML the user has chosen the CDATA element (which signifies that  
everything that follows is pure text. Tags are not to be interpreted as such). It also contains several sentences within one source  
segment as the user has not done any segmentation while producing the XLIFF file. Linebreaks are created through P and BR tags. And  
the user has also used an entity, here the non-breaking space.]]></target>  
</trans-unit>
```

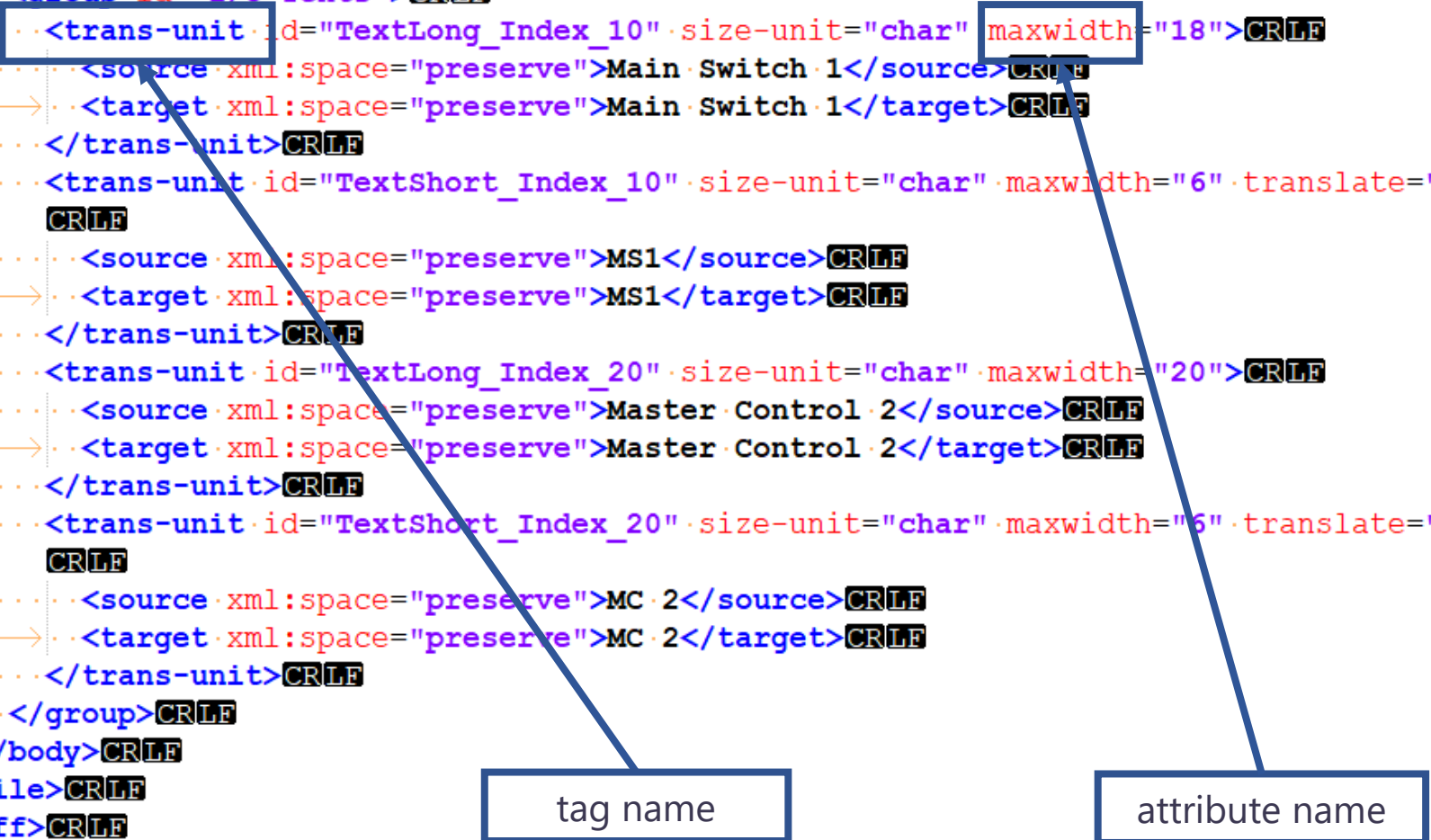

2. Length Restriction in Attribute

- Use the XML filter to specify the content of an attribute as a comment, for example a number as length restriction.

Length Restriction in Attribute

```

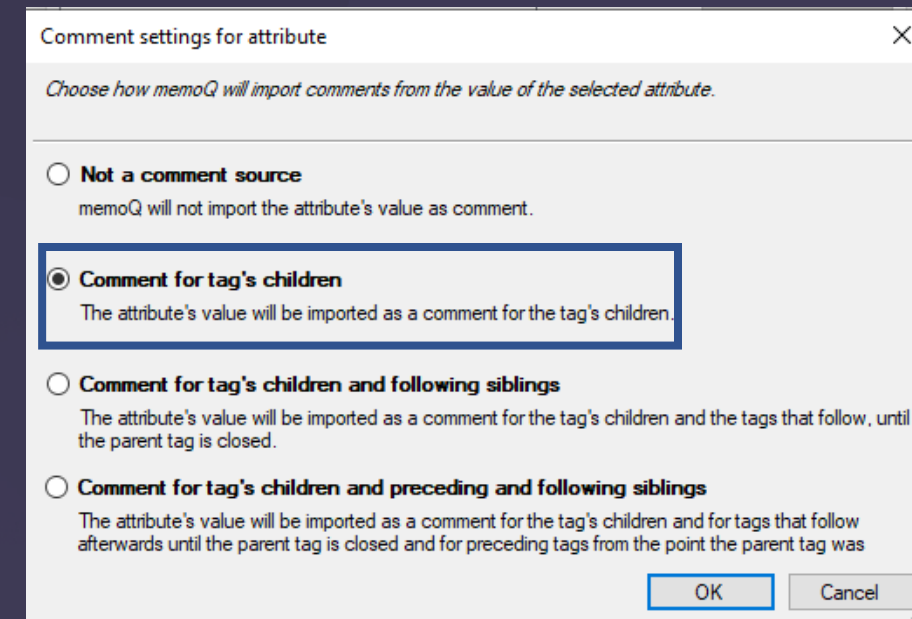
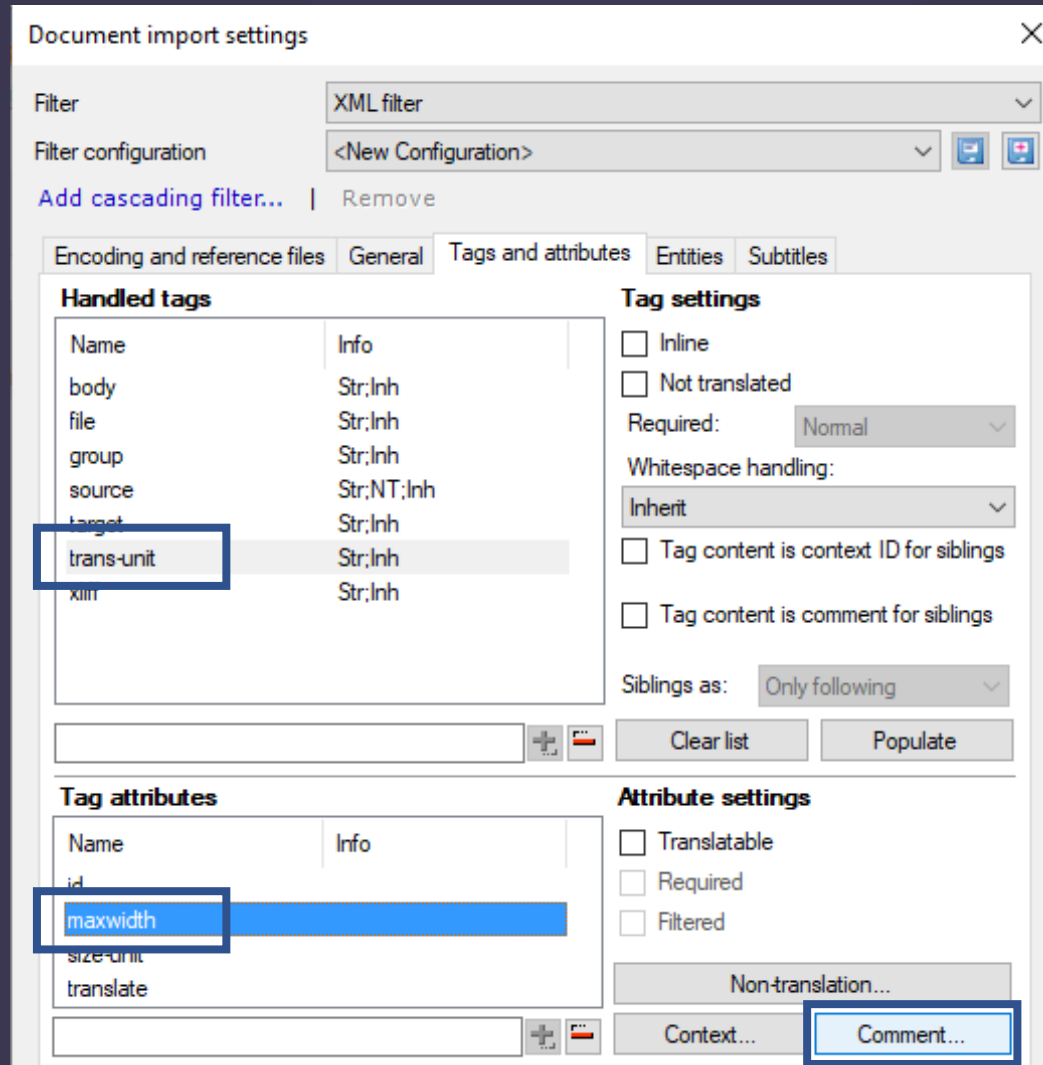
1  <?xml version="1.0" encoding="utf-8"?>
2  <xliff version="1.2" xmlns="urn:oasis:names:tc:xliff:document:1.2">
3  ..<file original="demo.txt" source-language="en-US" target-language="de-DE" datatype="plaintext" date="2019-08-16T13:39:27Z" build-num="1.0">
4  ....<body>
5  .....<group id="I/O-Texts">
6  .....<trans-unit id="TextLong_Index_10" size-unit="char" maxwidth="18">
7  .....<source xml:space="preserve">Main Switch 1</source>
8  .....<target xml:space="preserve">Main Switch 1</target>
9  .....</trans-unit>
10 .....<trans-unit id="TextShort_Index_10" size-unit="char" maxwidth="6" translate="no">
11 .....<source xml:space="preserve">MS1</source>
12 .....<target xml:space="preserve">MS1</target>
13 .....</trans-unit>
14 .....<trans-unit id="TextLong_Index_20" size-unit="char" maxwidth="20">
15 .....<source xml:space="preserve">Master Control 2</source>
16 .....<target xml:space="preserve">Master Control 2</target>
17 .....</trans-unit>
18 .....<trans-unit id="TextShort_Index_20" size-unit="char" maxwidth="6" translate="no">
19 .....<source xml:space="preserve">MC 2</source>
20 .....<target xml:space="preserve">MC 2</target>
21 .....</trans-unit>
22 .....</group>
23 .....</body>
24 .....</file>
25 </xliff>
  
```



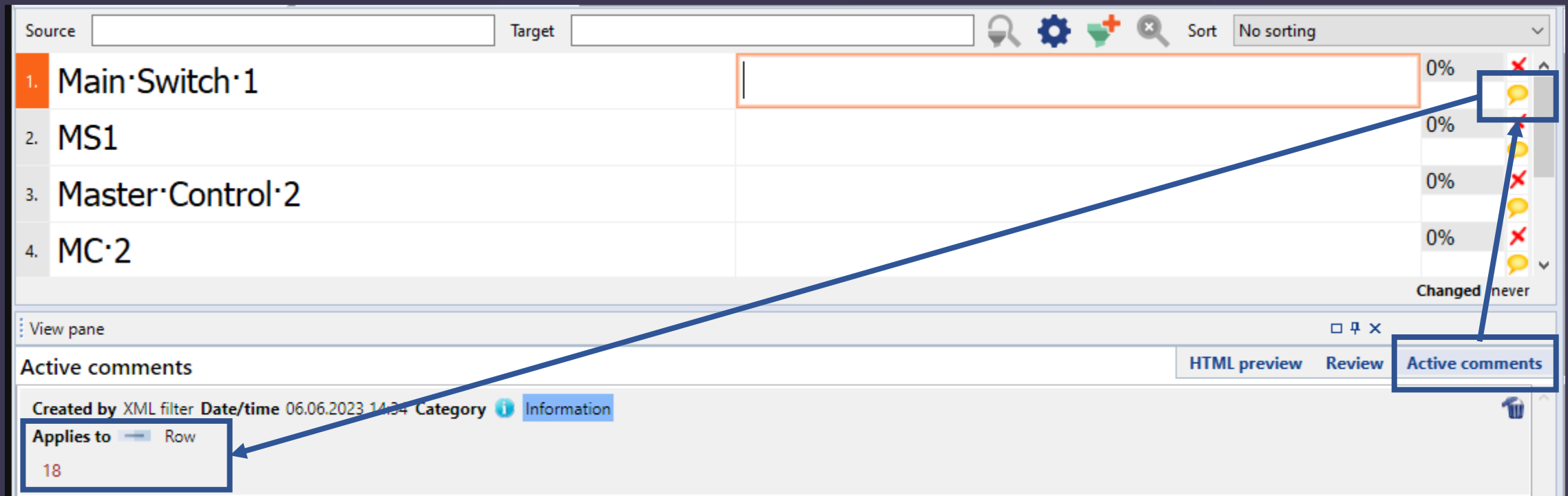
tag name

attribute name

Length Restriction in Attribute



Number in Comment for Length Check



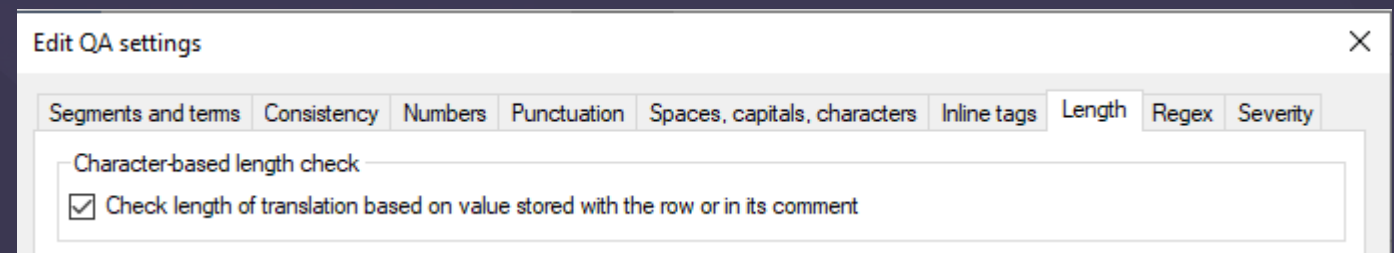
Source Target Sort No sorting

1.	Main·Switch·1	<input type="text"/>	0%	
2.	MS1		0%	
3.	Master·Control·2		0%	
4.	MC·2		0%	

View pane HTML preview Review Active comments

Created by XML filter Date/time 06.06.2023 14:34 Category Information

Applies to Row
18



Edit QA settings

Segments and terms Consistency Numbers Punctuation Spaces, capitals, characters Inline tags **Length** Regex Severity

Character-based length check

Check length of translation based on value stored with the row or in its comment

3. Normalization

- If the XLIFF file contains line breaks or indents, make sure to set up the XML filter to preserve whitespace!
- Otherwise the XML filter will "normalize" all whitespace within the text (i.e. reduce several whitespaces to just one).

Normalization

Source text with line breaks and indents

```

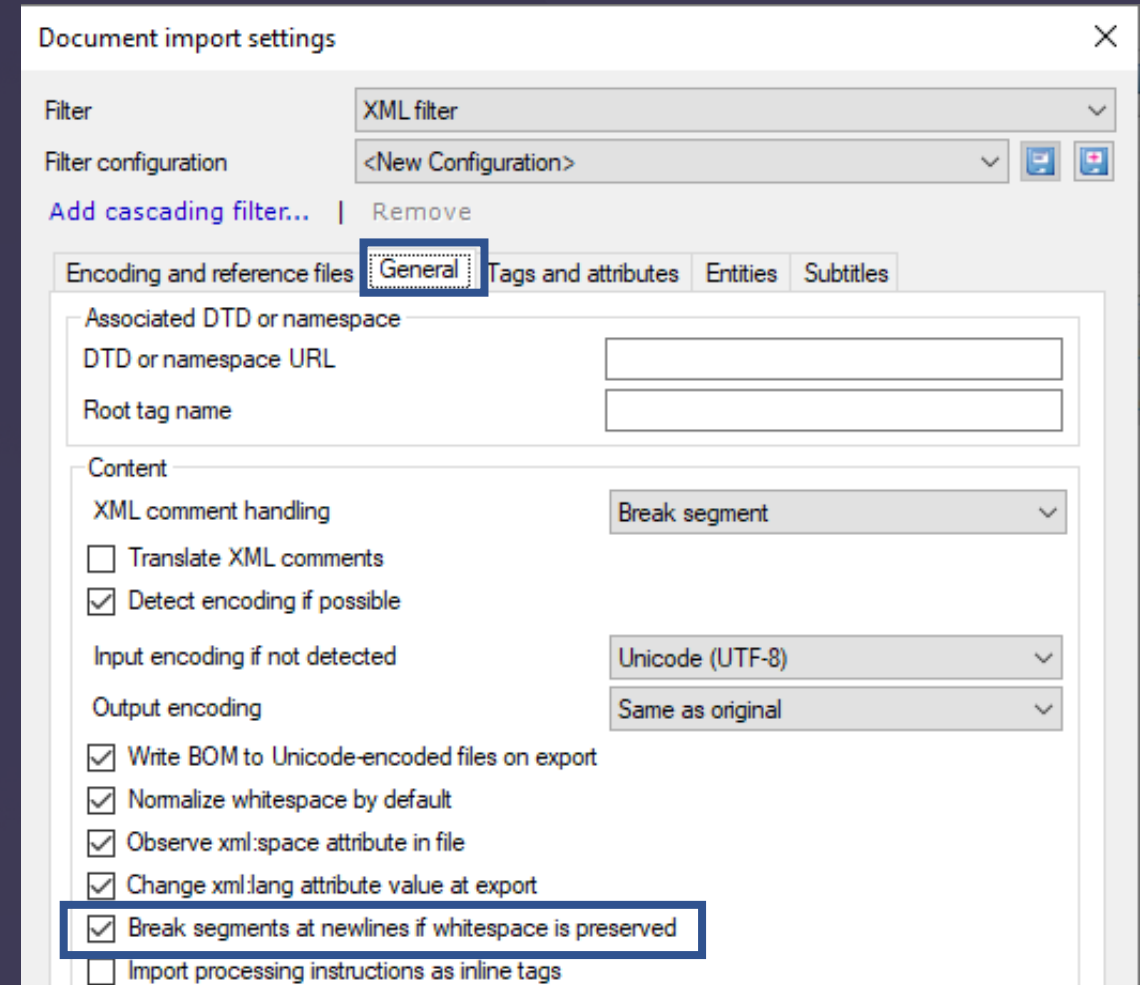
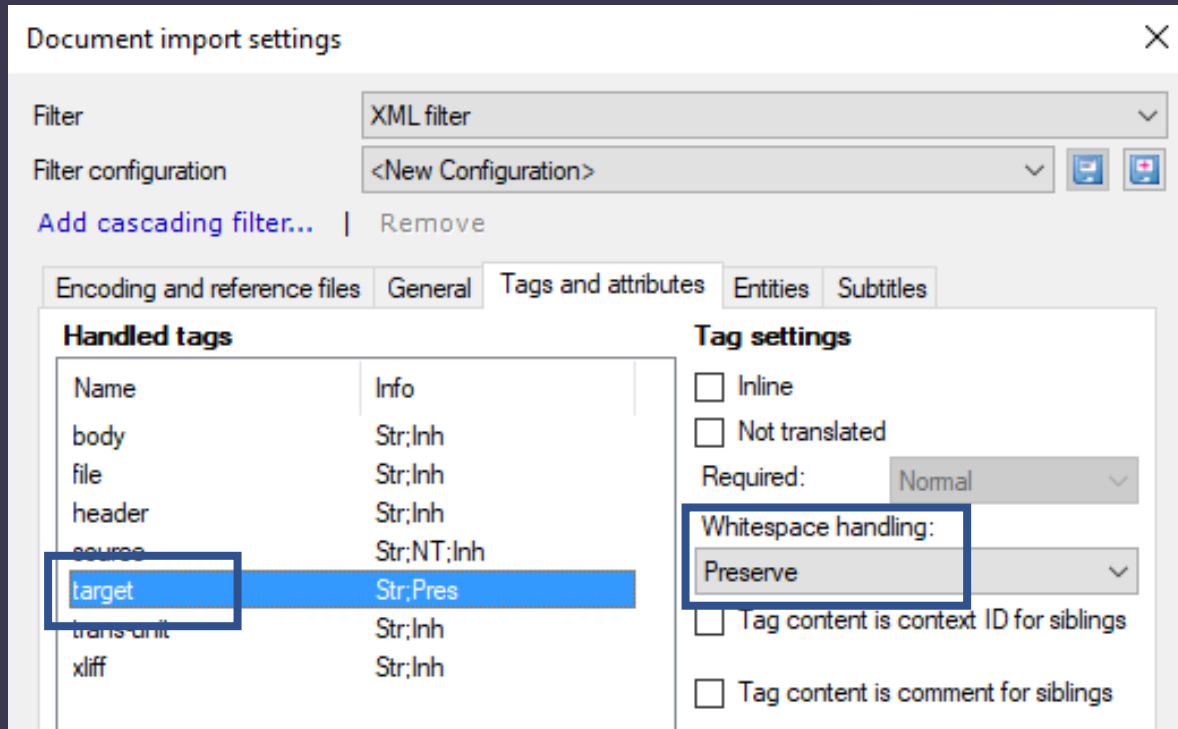
<?xml version="1.0" encoding="UTF-8"?>
<xliff version="1.0">
  <file source-language="en-us" target-language="de-de" datatype="plaintext" original="messages" date="2022-12-15T10:45:00Z" product-name="prod_name">
    <header />
    <body>
      <trans-unit id="UserEmailNewsletter.content" xml:space="preserve">
        <source><![CDATA[
          Dear {salutation},

          Thank you for your interest in our newsletter. Please enter your email address by clicking on the following link:

          <a href="{uri}">Complete registration</a>

          If the link does not work, please copy/paste the link address into the address field of your browser.
        ]]></source>
        <target state="translated"><![CDATA[
          Guten Tag {salutation}, vielen Dank für Ihr Interesse an unserem Newsletter. Bitte geben Sie Ihre E-Mail-Adresse ein, indem Sie dem folgenden Link folgen: <a href="{uri}">Registrierung abschließen</a> Sollte der Link nicht funktionieren, kopieren Sie ihn bitte und fügen ihn im Adressfeld Ihres Browsers ein.
        ]]></target>
      </trans-unit>
    </body>
  </file>
</xliff>
  
```

After translation - normalized text



XML filter + HTML filter

Normalization

2nd step: HTML filter

3rd step: Regex Tagger

Source	Target	Progress	Actions
1. Dear {salutation}		0%	✖
2. Thank you for your interest in our newsletter.		0%	✖
3. Please enter your email address by clicking on the following link:		0%	✖
4. Complete registration		0%	✖
5. If the link does not work, please copy/paste the link address into the address field of your browser.		0%	✖
6. Registrierung abgeschlossen		0%	✖

```
View pane Changed never □ 🔍 ×  
<source><![CDATA[  
Dear {salutation},  
Thank you for your interest in our newsletter. Please enter your email address by clicking on the following link:  
<a href="{uri}">Complete registration</a>  
If the link does not work, please copy/paste the link address into the address field of your browser.  
></source>  
<target state="translated"><![CDATA[  
Dear {salutation},  
Thank you for your interest in our newsletter. Please enter your email address by clicking on the following link:  
<a href="{uri}">Complete registration</a>  
If the link does not work, please copy/paste the link address into the address field of your browser.  
></target>  
</trans-unit>
```

Source and target text with same line breaks and indents.

4. Source Text in Target Section

The `<source>...</source>` tags do not contain the source text.

```
<trans-unit id="1">  
<source xml:lang="en-US"><![CDATA[company.articleOverview.all]]></source>
```

```
<target xml:lang="de-DE"><![CDATA[All Articles]]></target>  
</trans-unit>
```

Use an XML filter as previously described.

5. Translation in the Note

The client wants to have the translation in a different area.

```
<trans-unit id="1">  
  <source xml:lang="en-US">This is the source segment.</source>  
  <note>TRANSLATE TO HERE</note>  
</trans-unit>
```

Use a **multilingual XML** filter to define the source text area and the target text area.

5. Translation in the Note

XML import rules

XML file (click on row for its XPath expression)

```

<xiff version="1.2" xmlns1="urn:oasis:names:tc:xliff:document:1.2">
  <file original="demo.txt" source-language="en-US" note-language="de-DE" datatype="plaintext" date="2007-01-01">
    <body>
      <group id="I/O-Texts">
        <trans-unit id="001">
          <source xml:space="preserve">
            Source segment one.
          </source>
          <note xml:space="preserve">
            TRANSLATE TO HERE
          </note>
        </trans-unit>
        <trans-unit id="002">
        </trans-unit>
        <trans-unit id="003">
        </trans-unit>
        <trans-unit id="004">
        </trans-unit>
      </group>
    </body>
  </file>
</xiff>

```

XPath for selected row

Content rule

Content XPath

Fill from selected row

memoQ language

German (Germany)

Context (ID) XPath for content

Fill from selected row

Length restriction XPath for content

Fill from selected row

Comment XPath for content

Fill from selected row

memoQ comment type

Information

Save to rule set

Import rule set

eng-US;//source;;;0;
ger-DE;//note;;;0;

Please note that this filter does not allow a cascading HTML filter (at the moment).

5. Translation in the Note

Source
Target

1.	Source segment one.	TRANSLATE TO HERE
2.	Source segment two.	TRANSLATE TO HERE
3.	Source segment three.	TRANSLATE TO HERE
4.	Source segment four.	TRANSLATE TO HERE

View pane

Source	Source segment one.
ger-DE	TRANSLATE TO HERE
ID	

Source	Source segment two.
ger-DE	TRANSLATE TO HERE
ID	

6. Non-HTML Tags

The text contains custom tags to signify a line break.

```
color info.xf
13 <source>The names of the following buttons are customizable:</source>
14 <target>The names of the following buttons are customizable:</target>
15 </trans-unit>
16 <trans-unit xml:space="preserve" id="chapter:2_1_0_0_0_6-page:0-object:Interaktion10_0-type:inter_tit4.tit">
17 <source>Button small</source>
18 <target>Button small</target>
19 </trans-unit>
20 <trans-unit xml:space="preserve" id="chapter:2_1_0_0_0_6-page:0-object:Interaktion11_0-type:inter_tit4.tit">
21 <source>Button long</source>
22 <target>Button long</target>
23 </trans-unit>
24 <trans-unit xml:space="preserve" id="chapter:2_1_0_0_0_6-page:0-object:Interaktion12_0-type:inter_tit4.tit">
25 <source>Button large/multi-line</source>
26 <target>Button large/multi-line</target>
27 </trans-unit>
28 <trans-unit xml:space="preserve" id="chapter:2_1_0_0_0_6-page:1-object:Text1_0-type:text">
29 <source>Edit the button properties via a right-click. &lt;x id='lb1' ctype='lb' />
   Type the new name and save it.</source>
30 <target>Edit the button properties via a right-click. &lt;x id='lb1' ctype='lb' />
   Type the new name and save it.</target>
31 </trans-unit>
```

<x id='lb1' ctype='lb' />

<x id='lb1' ctype='lb' />

Non-HTML Tags

Tags show up as text when imported with XLIFF or XML filter.

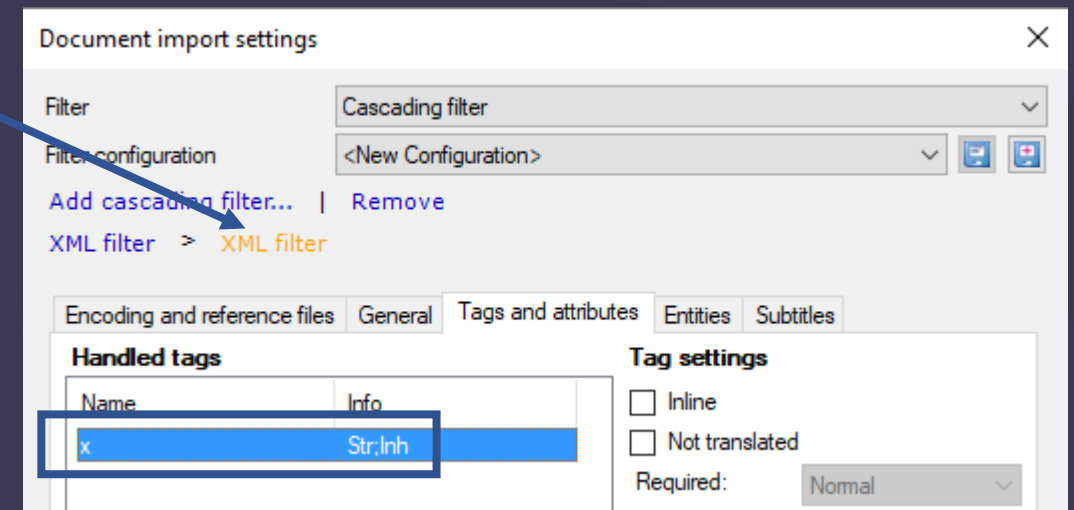
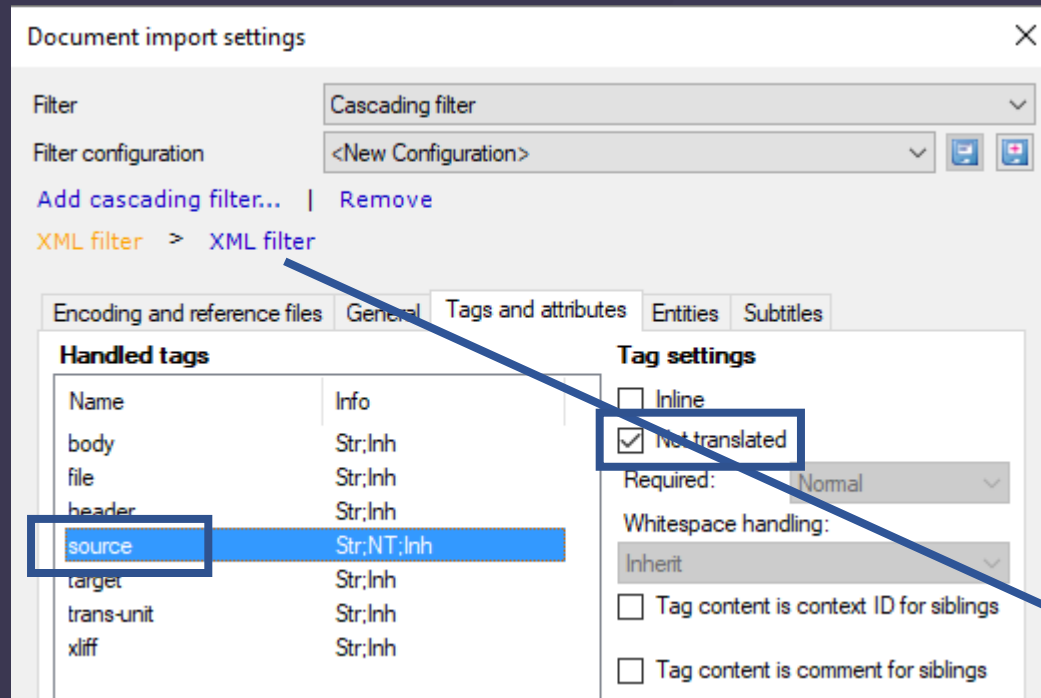
Project home color info.xlf x

Source Target

1. Template:
2. Buttons
3. The names of the following buttons are customizable:
4. Button·small
5. Button·long
6. Button·large/multi-line
7. Edit the button properties via a right-click.
`<x id='lb1' ctype='lb'/>` Type the new name and save it.
8. Template:

Non-HTML Tags

- Use 2 XML filter steps
- to convert the x tag text to a "real" tag
- to use the x tag as a structure tag



Non-HTML Tags

x tag used for segmentation

Project home color info.xlf

Source Target

1. Template:
2. Buttons
3. The names of the following buttons are customizable:
4. Button small
5. Button long
6. Button large/multi-line
7. Edit the button properties via a right-click.
8. Type the new name and save it.
9. Template:
10. Panel 1 - base colors
11. Blue, Red and Black.
12. In addition, 3 shades of gray are available.

View pane

```
</trans-unit>  
<trans-unit xml:space="preserve" id="chapter:2_1_0_0_1_6-page:1-objec:1ext:10-type:text">  
  <source>Edit the button properties via a right-click. &lt;x id='&apos;lb1&apos; ctype='&apos;lb&apos; it.</source>  
  <target>Edit the button properties via a right-click.Type the new name and save it.</target>  
</trans-unit>
```


6. G-Tags for Tags in the Original File

XLIFF has a tag called `<g>` which can be used to act as a placeholder for tags in the source file format.

XLIFF filter

- large segments
- a lot of tags

Here the `<g>` tag could be defined as a structure tag.

But here the `<g>` tag acts as an inline tag as well.

Source Target

`<g>` `<g>` `<g>` Title° `<g>` `<g>` `<g>` `<g>` `<g>` `<g>` `<g>` Use of this document `<g>` `<g>` `<g>` `<g>` `<g>` This document is intended for the end user of the product.° `<g>` `<g>` `<g>` `<g>` It contains information on all editions of the product. `<g>` `<g>` `<g>` `<g>` We therefore recommend that you print out the section that deals with the product edition that you have purchased. `<g>` `<g>` `<g>` `<g>`° `<g>` `<g>`

2. `<g>` `<g>` Each section contains all the information you need for your product edition. `<g>` `<g>` `<g>` `<g>` If your product edition does not include a certain feature, this feature will not be described. `<g>` `<g>` `<g>` `<g>` `<g>` `<g>` `<g>` If you have any questions, please contact us through our online support platform. `<g>` `<g>` `<g>`

3. `<g>` Our products combine `<g>` style `<g>`, `<g>` design `<g>`, `<g>` and `<g>` and `<g>` Innovation `<g>` into one `<g>` highly creative `<g>` package. `<g>` `<g>` `<g>` See for yourself! `<g>` `<g>`

XML filter

g tag is inline tag

Segmentation is slightly better, segments are no longer so big.

Source Target

1. Title of the document

2. Use of this document

3. This document is intended for the end user of the product. It contains information on all editions of the product. We therefore recommend that you print out the section that deals with the product edition that you have purchased.

4. Each section contains all the information you need for your product edition. If your product edition does not include a certain feature, this feature will not be described. If you have any questions, please contact us through our online support platform.

Cascading filter


















Regex Textfilter 1

Regex Textfilter 2

Regex Textfilter 3

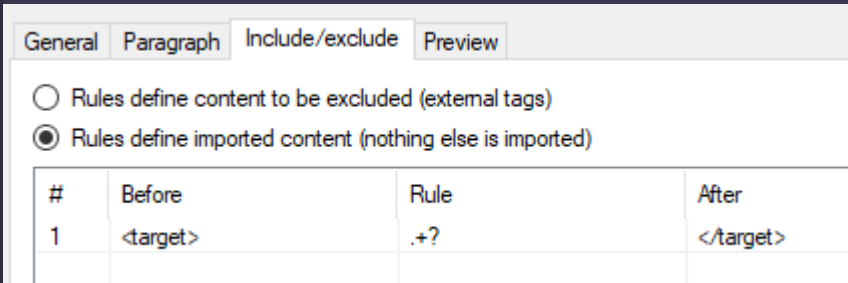
XML

Segmentation is even better and the amount of tags has been reduced in segments where tags are not used as internal tags.

4.	
5.	 It contains information on all editions of the product. 
6.	
7.	We therefore recommend that you print out the section that deals with the product edition that you have purchased.
8.	Each section contains all the information you need for your product edition.
9.	If your product edition does not include a certain feature, this feature will not be described.
10.	
11.	If you have any questions, please contact us through our online support platform.
12.	 Our products combine  style  ,  design  ,  and  and  Innovation  into one  highly creative  package. 
13.	See for yourself!

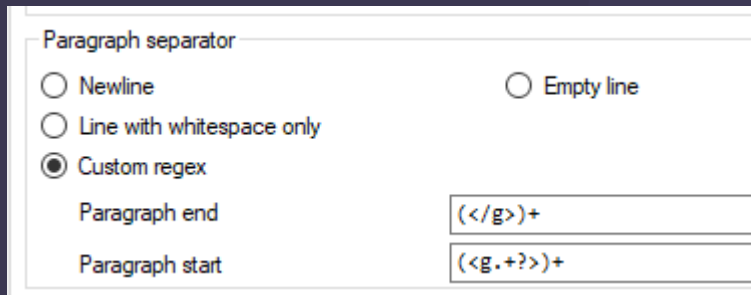
Cascading filter

Regex Textfilter 1 to extract text between <target> tags



#	Before	Rule	After
1	<target>	.+?	</target>

Regex Textfilter 2 to use several g tags as segmentation point



Paragraph separator

Newline Empty line

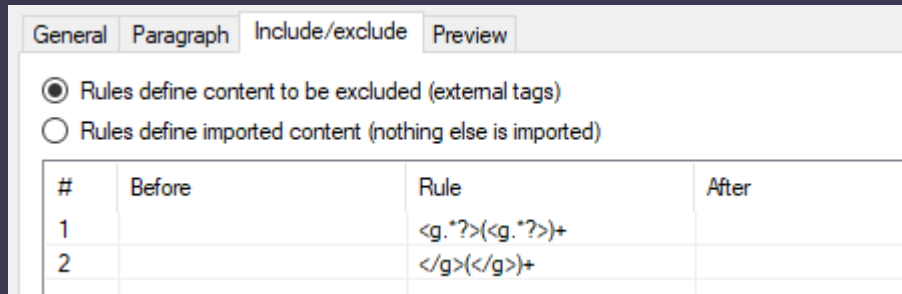
Line with whitespace only

Custom regex

Paragraph end: (</g>)+

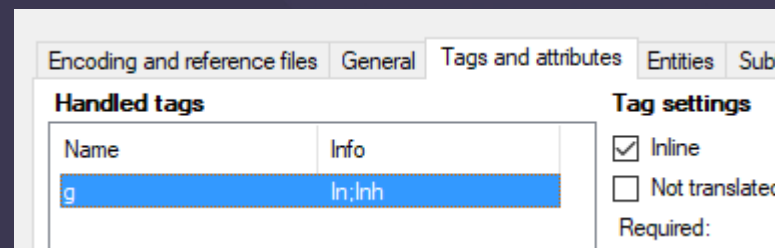
Paragraph start: (<g.+?>)+

Regex Textfilter 3 to exclude g tag sequences



#	Before	Rule	After
1		<g.*?><g.*?>+	
2		</g></g>+	

XML filter to show remaining g tags as internal tags



Name	Info
g	In;Inh

Tag settings

Inline

Not translated

Required:

More...

- XLIFF files that show the *.XML extension instead of *.XLIFF
 - Rename the file extension
- XLIFF files that do not import, but show an error message that the language pair of the XLIFF is not the same as the project's language settings
 - Check the XLIFF in a text editor and search for "source-language=" and "target-language=" and type in the correct language codes
 - Sometimes XLIFF files do not show language codes, but the information "default"

More...

- XLIFF files that show the *.XLIFF extension, but once opened with a text editor you can see that they are not XLIFF files at all, because the XLIFF header is missing.



Thank you!

Any questions?

zerfass@zaac.de