HOW TO FIX CORRUPT MULTITRANS XLIFF FILES

This is **not** an official document and is intended for support purposes only. This document assumes basic familiarity with XML syntax.

Recommended Tools

Notepad++ https://notepad-plus-plus.org/downloads/

- Once installed, also install the "XML Tools" plug-in via **Plugins > Plugins Admin…** (https://npp-user-manual.org/docs/plugins/#install-using-plugins-admin)

Liquid Studio (Free Community Edition) https://www.liquid-technologies.com/trial-download

- This software is not necessary but it will speed up the troubleshooting process.

Terminology used in this document

- **Element** – refers to an XML element. That is, everything between (and including) the opening and closing tag.
- **Close off** – add a closing tag after a corresponding opening tag.
- **Segment & trans-unit** – used somewhat interchangeably in this document, but strictly speaking a trans-unit (short for translation unit) encompasses both the source text (source segment) and the target translation (target segment).
- **Inline formatting** – formatting (font size, emphasis, colour etc.) that is applied only to a specific selection of text.

Overview of the MultiTrans XLIFF schema

The MultiTrans XLIFF schema is approximately based on the OASIS Standard XLIFF 1.2 specification. The original OASIS specification can be found here:
http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.html
While the MultiTrans XLIFF schema is slightly different, this specification can still be used as a reference to familiarise yourself with the overall file structure. Custom elements and attributes specific to MultiTrans are typically prefixed with **mc:**

The overall structure of a typical MultiTrans XLIFF file looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<xliff>
    <file>
        <header>
            <mc:skeleton></mc:skeleton>
        </header>
        <body>
            <!-- A group may contain one or many trans-units -->
            <group>
                <!-- A trans-unit represents a segment's source & target translation -->
                <trans-unit>
                    <source>Bonjour</source>
                    <target>Hello</target>
                </trans-unit>
            </group>
        </body>
    </file>
    <!-- There may be more than one file, e.g. when using the Merge Flow Machine Task -->
</xliff>
```

To summarise each part of this structure:

- Each XLIFF file has exactly one root **<xliff>** element.
- There may be one or many **<file>** elements containing one **<header>** and one **<body>** element. There will only be more than one **<file>** element if the XLIFF file was created using the “MultiTrans Merge” Flow Machine Task.
- The **<header>** contains an **<mc:skeleton>** element which is used when re-constructing the document back into its native format.
- The **<body>** contains one or many **<group>** elements.
- Each **<group>** element contains one or more **<trans-unit>** elements.
- Each **<trans-unit>** must contain exactly one **<source>** and one **<target>** element. The source and target contains the text from the original source document and its translation to the target language. These may contain just plain text, but often there will be additional tags to specify various formatting or layout options.

Trans-units in more detail

Aside from the mandatory **<source>** and **<target>** elements, a **<trans-unit>** may also contain many **<alt-trans>** (short for "alternative translation") elements which are used to track the changes made to each segment. Here is a simplified example:

```
<trans-unit extype="DOCX" id="57">
    <source xml:space="preserve">Please translate this sentence.</source>
    <target state="translated" xml:space="preserve">Veuillez traduire cette phrase.</target>
    <alt-trans alttranstype="previous-version" origin="1">
        <source xml:space="preserve">Please translate this sentence.</source>
        <target xml:space="preserve">Veuillez traduire ceci.</target>
    </alt-trans>
    <mc:props>
        <mc:prop mc:name="Created by" mc:type="STRING">Alice</mc:prop>
        <mc:prop mc:name="Created date" mc:type="DATE">20210101T090000Z</mc:prop>
        <mc:prop mc:name="Modified by" mc:type="STRING">Alice</mc:prop>
        <mc:prop mc:name="Modified date" mc:type="DATE">20210101T103000Z </mc:prop>
    </mc:props>
</trans-unit>
```

In this example there is one **<alt-trans>** element which contains its own <source> and <target> elements. Note that the **alttranstype="previous-version"** attribute indicates that this is a previous version of this trans-unit.

Other things to note about this example are:

- The **<mc:props>** element contains metadata about this particular trans-unit, such as who last modified it and when.
- There can be many **<alt-trans>** elements within the same <trans-unit>, each with their own child **<source>** and **<target>** elements, but there must only ever be one <source> and one <target> that is a direct child of the <trans-unit> element.

Here’s an example where there is extra formatting information applied to the text in the segment:

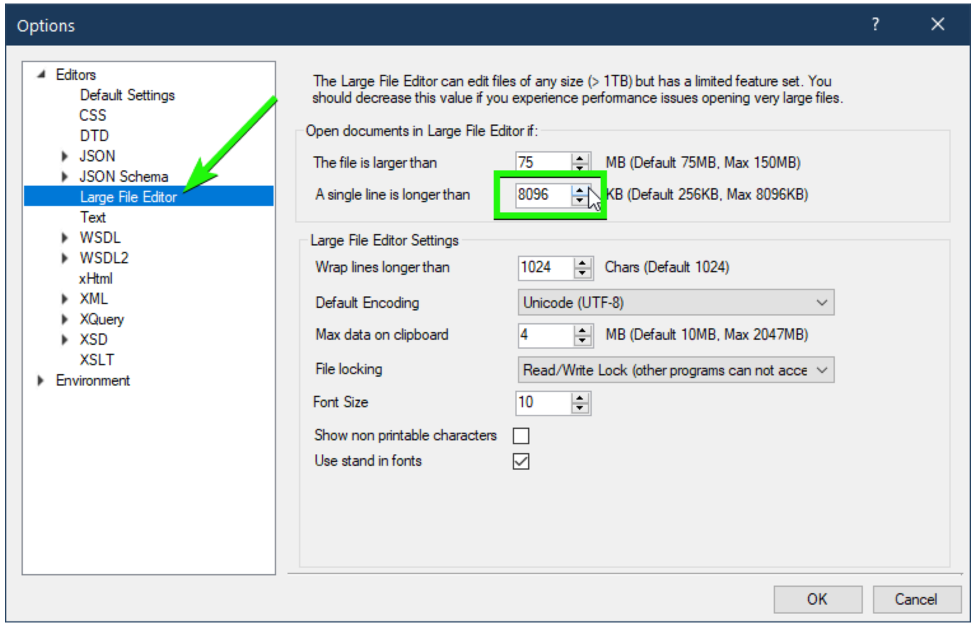
```
<source xml:space="preserve">
    <g ctype="bold" id="65" mc:smlformat="Biu">Click here to search for information: </g>
    <bpt ctype="link" id="b66" rid="r66">&lt;w:hyperlink r:id="rId7" w:history="1"&gt;</bpt>
    <g ctype="bold underlined x-mc-char-color x-mc-char-kern" id="67" mc:smlformat="BiU">https://duckduckgo.com</g>
    <ept id="e66" rid="r66" xml:ctype="link">&lt;/w:hyperlink&gt;</ept>
</source>
<target state="translated" xml:space="preserve">
    Cliquez ici pour rechercher des informations:
    <bpt ctype="link" id="b66" rid="r66">&lt;w:hyperlink r:id="rId7" w:history="1"&gt;</bpt>
        https://duckduckgo.com/?kad=fr_FR
    <ept id="e66" rid="r66" xml:ctype="link">&lt;/w:hyperlink&gt;</ept>
</target>
```

The **<g>** element is a common tag used to apply inline formatting to the text that it surrounds. In the above example, the text that comes before the hyperlink has bold formatting applied to it (**ctype=”bold”**) in the source element, but there is no formatting applied to the equivalent translation in the target segment because there are no surrounding **<g>** tags.

Wherever there is a **<bpt>** element, there will also be a corresponding **<ept>** element somewhere after it. BPT and EPT stand for **B**egin **P**aired **T**ag and **E**nd **P**aired **T**ag respectively and they are used to provide native code that the source format will understand. In the example above, these tags are used to provide **<hyperlink>** tags which surround the URL in the text. When the target Word document is exported, these hyperlink tags are used to display the text https://duckduckgo.com/?kad=fr_FR as a clickable link in Microsoft Word.

How to set up Liquid Studio for troubleshooting

When you first open Liquid Studio, go to ***Tools > Options > Editors > Large File Editor*** then change the **A single line is longer than** setting to the maximum value (**8096KB**). If an XLIFF file is opened in the Large File Editor, you cannot use the validation features, so changing this option prevents this from happening:



The Large File Editor can edit files of any size (> 1TB) but has a limited feature set. You should decrease this value if you experience performance issues opening very large files.

Open documents in Large File Editor if:

The file is larger than 75 MB (Default 75MB, Max 150MB)

A single line is longer than 8096 KB (Default 256KB, Max 8096KB)

Large File Editor Settings

Wrap lines longer than 1024 Chars (Default 1024)

Default Encoding Unicode (UTF-8)

Max data on clipboard 4 MB (Default 10MB, Max 2047MB)

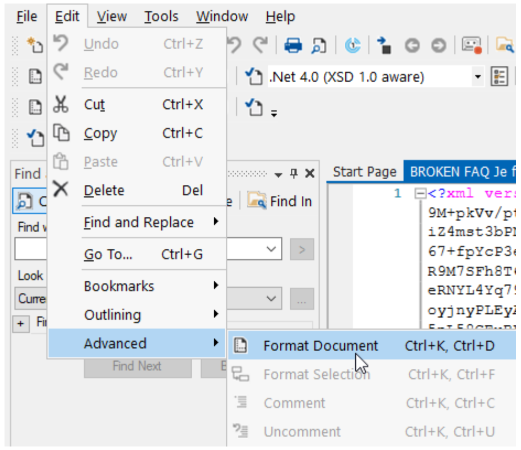
File locking Read/Write Lock (other programs can not acce

Font Size 10

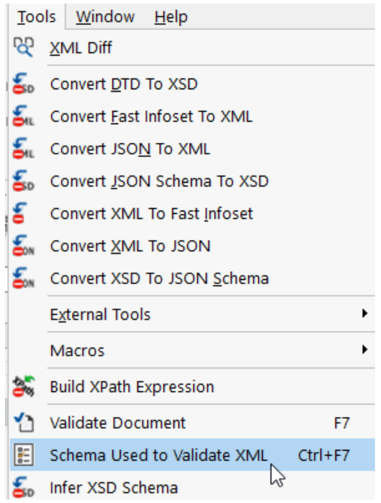
Show non printable characters ☐

Use stand in fonts ☑

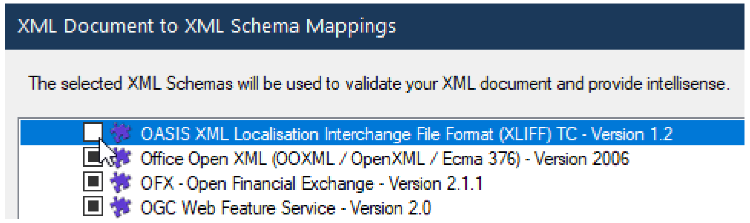
Go to ***File > Open > File…*** and browse to the corrupt XLIFF file. Once open, go to ***Edit > Advanced > Format Document*** to make the XML more readable:



To set up validation, go to ***Tools > Schema used to validate XML***:



In the list of XML Schemas, scroll down to the **OASIS XML Localisation Interchange File Format (XLIFF) TC – Version 1.2** option and **deselect** it:

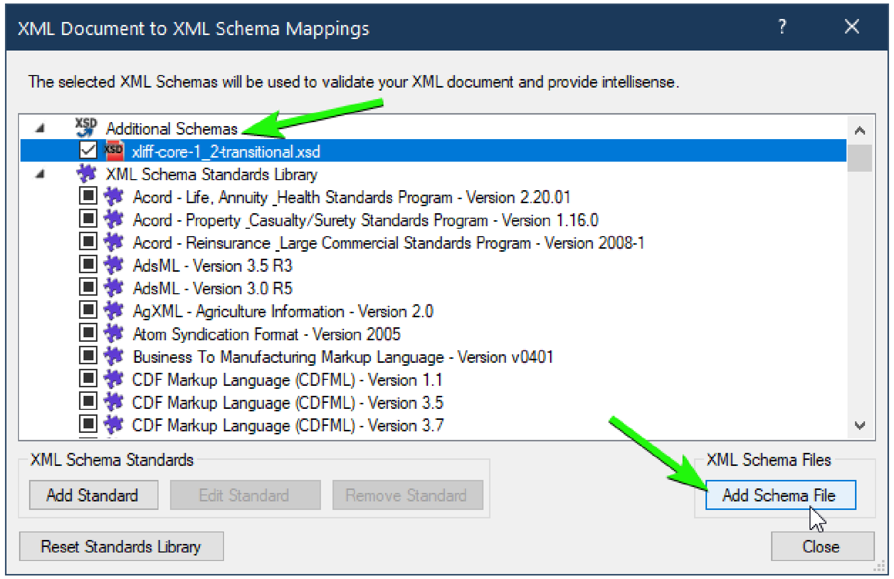


XML Document to XML Schema Mappings

The selected XML Schemas will be used to validate your XML document and provide intellisense.

- [ ] OASIS XML Localisation Interchange File Format (XLIFF) TC - Version 1.2
- [x] Office Open XML (OOXML / OpenXML / Ecma 376) - Version 2006
- [x] OFX - Open Financial Exchange - Version 2.1.1
- [x] OGC Web Feature Service - Version 2.0

Select the **Add Schema File** button and browse to the **xliff-core-1_2-transitional.xsd** XML schema file. You should then see the schema file listed under **Additional Schemas** at the top of the list:

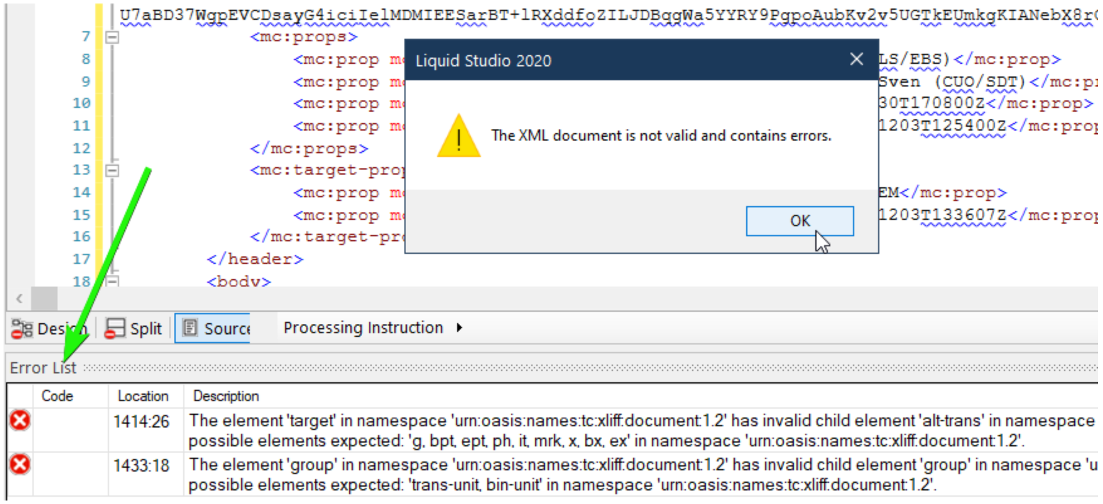


XML Document to XML Schema Mappings

The selected XML Schemas will be used to validate your XML document and provide intellisense.

- Additional Schemas
  - ✓ xliff-core-1_2-transitional.xsd
- XML Schema Standards Library
  - Acord - Life, Annuity _Health Standards Program - Version 2.20.01
  - Acord - Property _Casualty/Surety Standards Program - Version 1.16.0
  - Acord - Reinsurance _Large Commercial Standards Program - Version 2008-1
  - AdsML - Version 3.5 R3
  - AdsML - Version 3.0 R5
  - AgXML - Agriculture Information - Version 2.0
  - Atom Syndication Format - Version 2005
  - Business To Manufacturing Markup Language - Version v0401
  - CDF Markup Language (CDFML) - Version 1.1
  - CDF Markup Language (CDFML) - Version 3.5
  - CDF Markup Language (CDFML) - Version 3.7

Close this window and you'll see validation errors appear in the Error List at the bottom of the screen. **Double click an error** to go to the corresponding line in the file:



```
U7aBD37WgpEVCDsayG4iciIelMDMIEESarBT+1RXddfoZILJDBqgWa5YYRY9PgpoAubKv2v5UGTkEUmkgKIANebX8r
        <mc:props>
            <mc:prop m                                    LS/EBS)</mc:prop>
            <mc:prop m                                    Sven (CUO/SDT)</mc:p
            <mc:prop m                                    30T170800Z</mc:prop>
            <mc:prop m                                    1203T125400Z</mc:prop
        </mc:props>
        <mc:target-pro
            <mc:prop m                                    EM</mc:prop>
            <mc:prop m                                    1203T133607Z</mc:prop
        </mc:target-pr
    </header>
    <body>
```

Liquid Studio 2020

The XML document is not valid and contains errors.

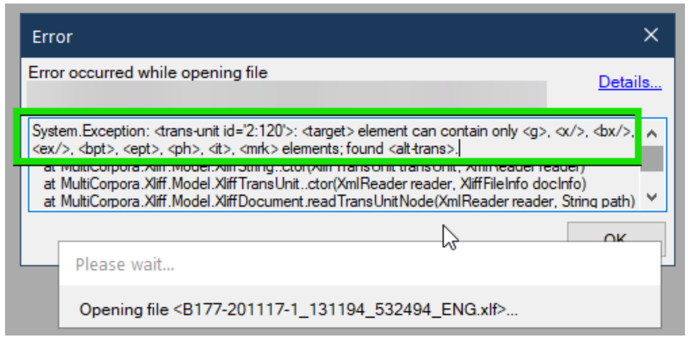
Code	Location	Description
	1414:26	The element 'target' in namespace 'urn:oasis:names:tc:xliff:document:1.2' has invalid child element 'alt-trans' in namespace possible elements expected: 'g, bpt, ept, ph, it, mrk, x, bx, ex' in namespace 'urn:oasis:names:tc:xliff:document:1.2'.
	1433:18	The element 'group' in namespace 'urn:oasis:names:tc:xliff:document:1.2' has invalid child element 'group' in namespace 'u possible elements expected: 'trans-unit, bin-unit' in namespace 'urn:oasis:names:tc:xliff:document:1.2'.

Once you’re done fixing an XLIFF file in Liquid Studio, you will need to remove the extra whitespace created by using the Format Document option. See **step 5 of the next section below** for instructions on how to do this using Notepad++.

Overview of how to fix files without Liquid Studio (using just Notepad++)

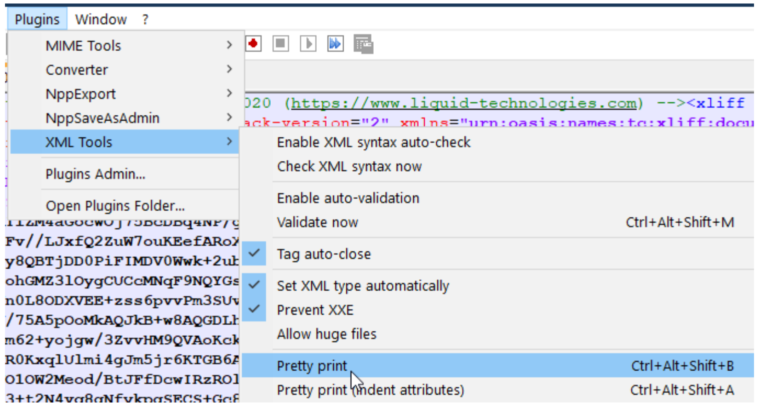
If for some reason you are unable to use Liquid Studio, here is how you can use Notepad++ to fix an XLIFF file:

1. Open the corrupt XLIFF file in the MultiTrans XLIFF Editor and make a note of the error that appears:



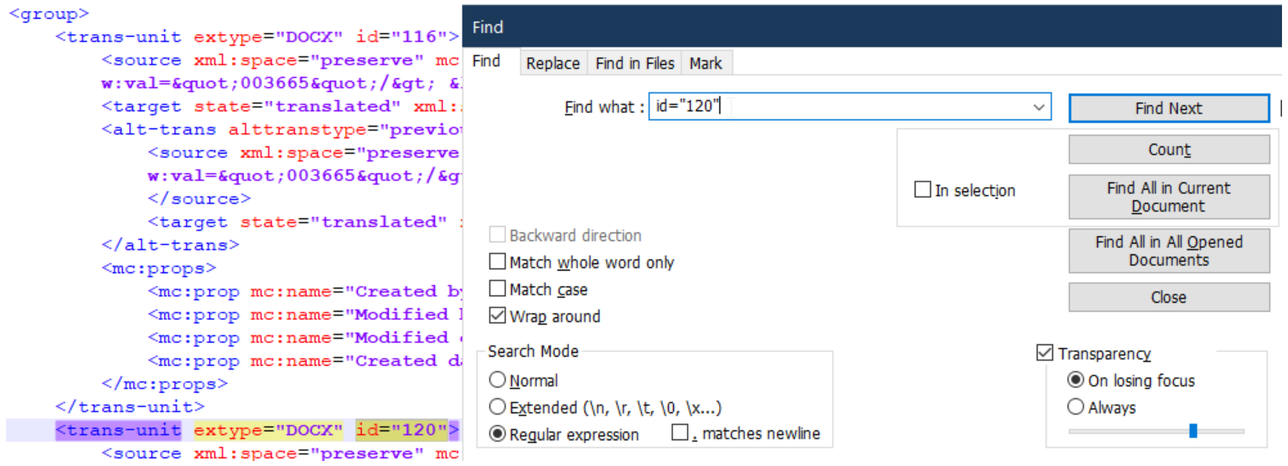
```
Error
Error occurred while opening file
System.Exception: <trans-unit id='2:120'>: <target> element can contain only <g>, <x/>, <bx/>, <ex/>, <bpt>, <ept>, <ph>, <it>, <mrk> elements; found <alt-trans>.
   at MultiCorpora.Xliff.Model.XliffTransUnit..ctor(XmlReader reader, XliffFileInfo docInfo)
   at MultiCorpora.Xliff.Model.XliffDocument.readTransUnitNode(XmlReader reader, String path)
Please wait...
Opening file <B177-201117-1_131194_532494_ENG.xlf>...
```

2. Open the XLIFF file in Notepad++ then select ***Plugins > XML Tools > Pretty Print*** (or press Ctrl+Alt+Shift+B) to make the XML more readable:



```
020 (https://www.liquid-technologies.com) --><xliff
ack-version="2" xmlns="urn:oasis:names:tc:xliff:docu
Fv//LJxfQ2ZuW7ouKEefARoX
y8QBTjDD0PiFIMDV0Wwk+2ub
ohGMZ31OygCUCcMNqF9NQYGs
n0L8ODXVEE+zss6pvvPm3SUv
/75A5pOoMkAQJkB+w8AQGDLh
m62+yojgw/3ZvvHM9QVAoKck
R0KxqlUlmi4gJm5jr6KTGB6A
O1OW2Meod/BtJFfDcwIRzRO1
```

3. The example message from step 1 contains **<trans-unit id=’2:120’>** which refers to the <trans-unit> element with an ID of 120 that is the 2^{nd} <trans-unit> element within its parent <group> element. To find this faulty <trans-unit> element, press **Ctrl+F** in Notepad++ and search for **id=”120”**:



```
<group>
    <trans-unit extype="DOCX" id="116">
        <source xml:space="preserve" mc
        w:val=&quot;003665&quot;/&gt; &
        <target state="translated" xml:
        <alt-trans alttranstype="previo
            <source xml:space="preserve
            w:val=&quot;003665&quot;/&g
            </source>
            <target state="translated" 
        </alt-trans>
        <mc:props>
            <mc:prop mc:name="Created b
            <mc:prop mc:name="Modified 
            <mc:prop mc:name="Modified 
            <mc:prop mc:name="Created d
        </mc:props>
    </trans-unit>
    <trans-unit extype="DOCX" id="120">
        <source xml:space="preserve" mc
```

Find what : id="120"

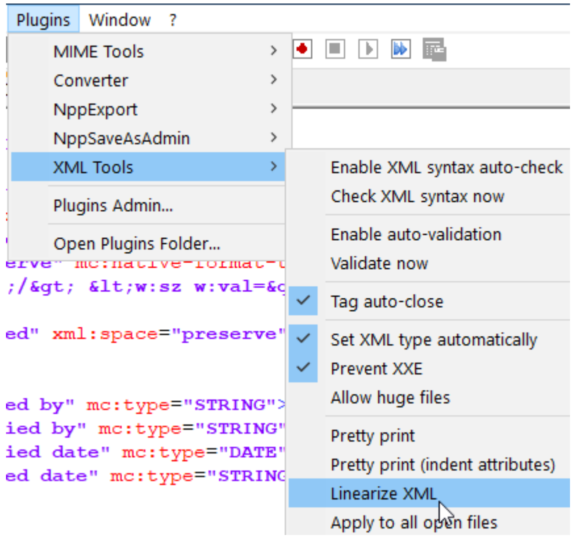
☐ In selection
☐ Backward direction
☐ Match whole word only
☐ Match case
☑ Wrap around

Search Mode
○ Normal
○ Extended (\n, \r, \t, \0, \x...)
◉ Regular expression ☐ . matches newline

☑ Transparency
◉ On losing focus
○ Always

Find
Find
Replace
Find in Files
Mark
Find Next
Count
Find All in Current Document
Find All in All Opened Documents
Close

4. Fix the problem with this **<trans-unit>**, save the file then open it again in the MultiTrans XLIFF Editor. Repeat steps 1 and 3 until the XLIFF opens without any issues.
5. Once the XLIFF file has no errors, make sure to select ***Plugins > XML Tools > Linearize XML*** (or press Ctrl+Alt+Shift+L) to remove the extra whitespace:



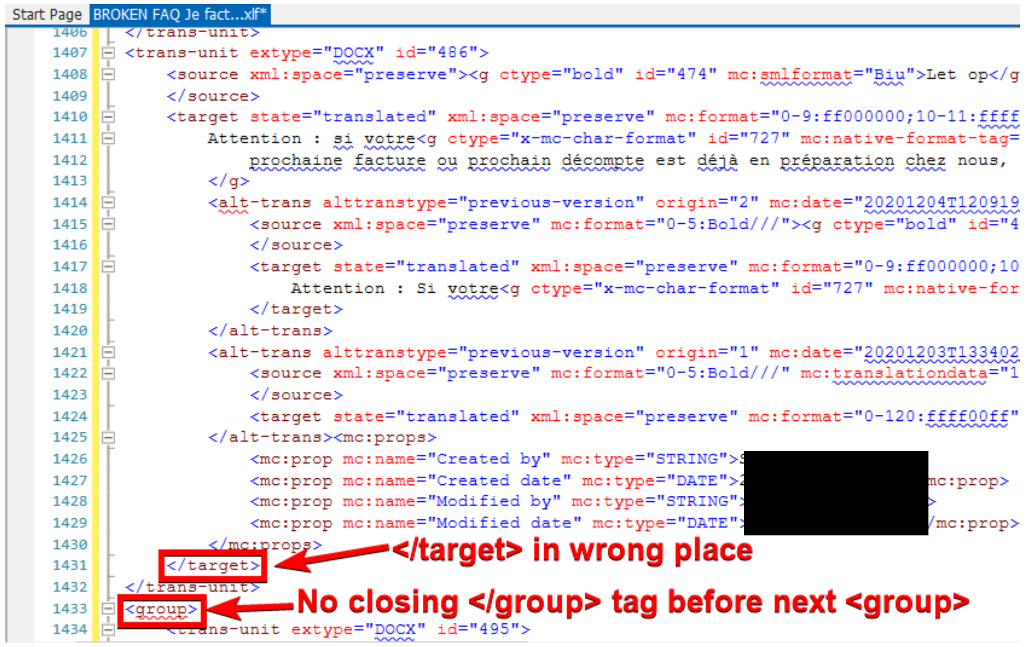
Common Errors

Missing or misplaced closing tags

The following error messages typically indicate that an XML element has not been closed properly:

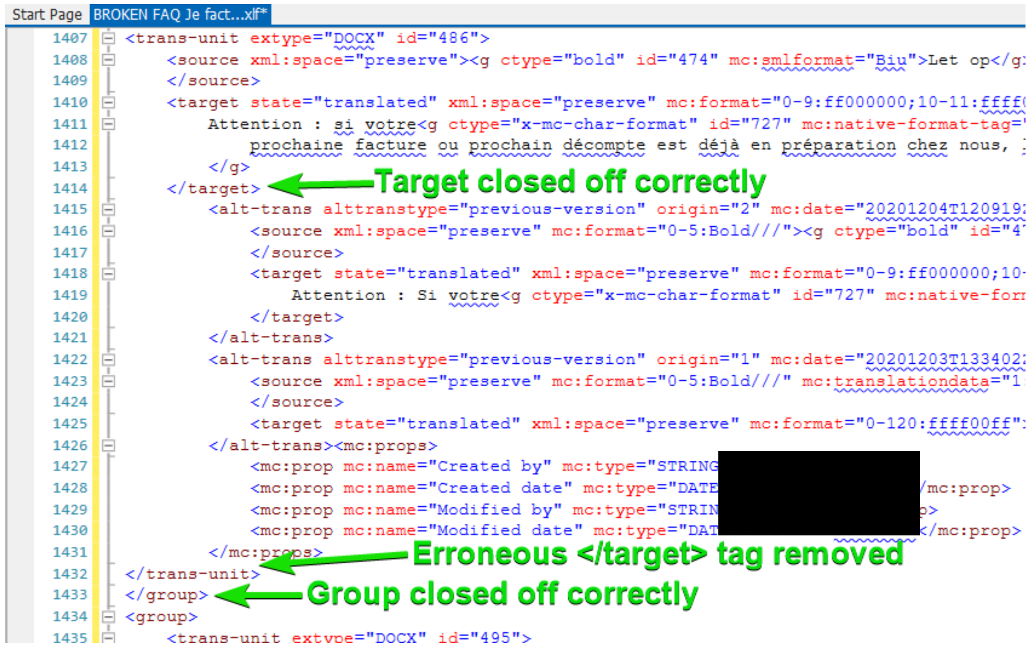
- The element ‘**target**’ in namespace ‘urn:oasis:names:tc:xliff:document:1.2’ has **invalid child element ‘alt-trans’**…
- The element ‘**group**’ in namespace ‘urn:oasis:names:tc:xliff:document:1.2’ has **invalid child element ‘group’**…

The example below shows how to fix both of these errors. The key is to make sure that closing tags are where they should be:



Start Page
BROKEN FAQ Je fact...xlf*
</target> in wrong place
No closing </group> tag before next <group>

```
</trans-unit>
<trans-unit extype="DOCX" id="486">
    <source xml:space="preserve"><g ctype="bold" id="474" mc:smlformat="Biu">Let op</g
    </source>
    <target state="translated" xml:space="preserve" mc:format="0-9:ff000000;10-11:ffff
        Attention : si votre<g ctype="x-mc-char-format" id="727" mc:native-format-tag=
            prochaine facture ou prochain décompte est déjà en préparation chez nous,
        </g>
        <alt-trans alttranstype="previous-version" origin="2" mc:date="20201204T120919
            <source xml:space="preserve" mc:format="0-5:Bold///"><g ctype="bold" id="4
            </source>
            <target state="translated" xml:space="preserve" mc:format="0-9:ff000000;10
                Attention : Si votre<g ctype="x-mc-char-format" id="727" mc:native-for
            </target>
        </alt-trans>
        <alt-trans alttranstype="previous-version" origin="1" mc:date="20201203T133402
            <source xml:space="preserve" mc:format="0-5:Bold///" mc:translationdata="1
            </source>
            <target state="translated" xml:space="preserve" mc:format="0-120:ffff00ff"
        </alt-trans><mc:props>
            <mc:prop mc:name="Created by" mc:type="STRING">[redacted]
            <mc:prop mc:name="Created date" mc:type="DATE">          mc:prop>
            <mc:prop mc:name="Modified by" mc:type="STRING">         >
            <mc:prop mc:name="Modified date" mc:type="DATE">         /mc:prop>
        </mc:props>
    </target>
</trans-unit>
<group>
    <trans-unit extype="DOCX" id="495">
```



Start Page
BROKEN FAQ Je fact...xlf*
Target closed off correctly
Erroneous </target> tag removed
Group closed off correctly

```
<trans-unit extype="DOCX" id="486">
    <source xml:space="preserve"><g ctype="bold" id="474" mc:smlformat="Biu">Let op</g
    </source>
    <target state="translated" xml:space="preserve" mc:format="0-9:ff000000;10-11:ffff
        Attention : si votre<g ctype="x-mc-char-format" id="727" mc:native-format-tag=
            prochaine facture ou prochain décompte est déjà en préparation chez nous,
        </g>
    </target>
        <alt-trans alttranstype="previous-version" origin="2" mc:date="20201204T120919
            <source xml:space="preserve" mc:format="0-5:Bold///"><g ctype="bold" id="4
            </source>
            <target state="translated" xml:space="preserve" mc:format="0-9:ff000000;10
                Attention : Si votre<g ctype="x-mc-char-format" id="727" mc:native-for
            </target>
        </alt-trans>
        <alt-trans alttranstype="previous-version" origin="1" mc:date="20201203T133402
            <source xml:space="preserve" mc:format="0-5:Bold///" mc:translationdata="1
            </source>
            <target state="translated" xml:space="preserve" mc:format="0-120:ffff00ff"
        </alt-trans><mc:props>
            <mc:prop mc:name="Created by" mc:type="STRING[redacted]
            <mc:prop mc:name="Created date" mc:type="DATE          /mc:prop>
            <mc:prop mc:name="Modified by" mc:type="STRIN          p>
            <mc:prop mc:name="Modified date" mc:type="DAT          </mc:prop>
        </mc:props>
</trans-unit>
</group>
<group>
    <trans-unit extype="DOCX" id="495">
```

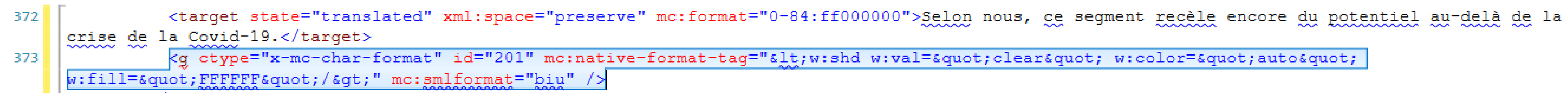
Duplicate / Erroneous <g> inline formatting tags

<g> elements can be long and contain several attributes, so you may wish to enable Word Wrap when dealing with these. In **Liquid Studio** this is done via ***Tools > Options > Editors > Default Settings > Word Wrap*** (important: turn this off when finished, otherwise you may struggle to open subsequent files if they're particularly large). In **Notepad++** this is done via ***View > Word Wrap***.

Here are a few simple tips to keep in mind when correcting <g> elements:

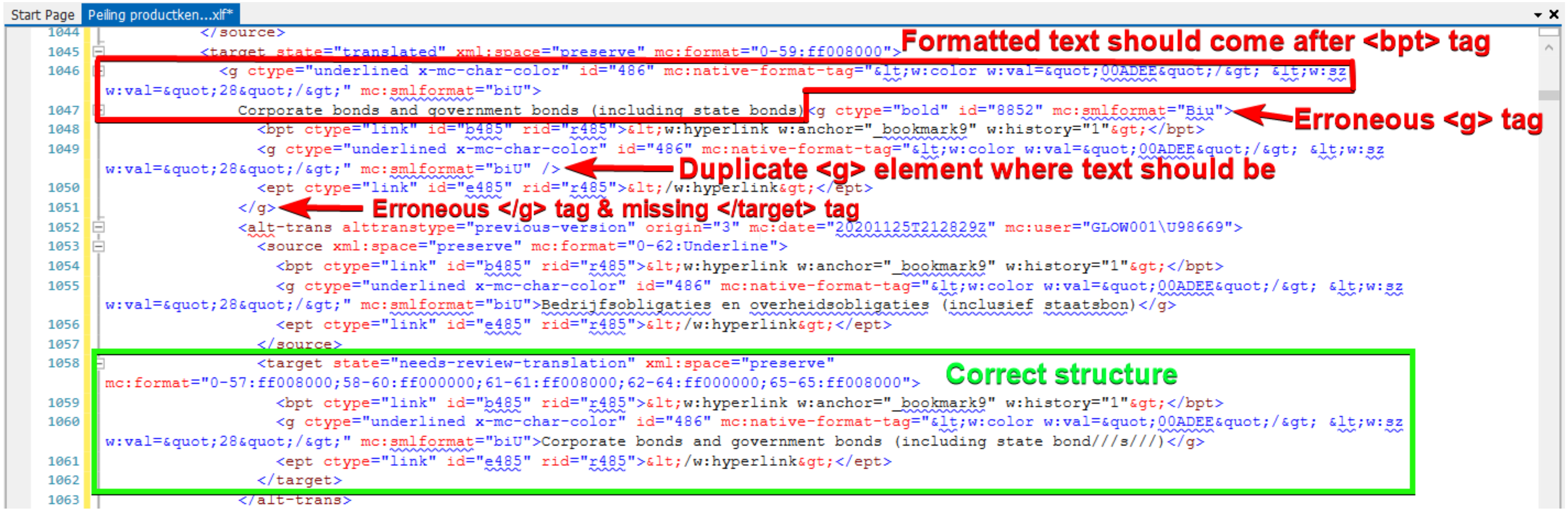
- They should not contain other XML elements, including other **<g>** tags.
- They should also only appear within a **<source>** or **<target>** element.
- They can be self-closing (i.e. **<g />** rather than **<g></g>**).
- The **<alt-trans>** elements containing previous translations will typically remain correctly formatted, so you can usually use their corresponding **<source>** and **<target>** elements as a reference to see roughly how the main **<source>** and **<target>** segments should be formatted.

Example 1 – A self-closing **<g>** element erroneously placed outside of a **<target>** element. In this case, it can either be deleted or moved before the closing **</target>** tag:



```
            <target state="translated" xml:space="preserve" mc:format="0-84:ff000000">Selon nous, ce segment recèle encore du potentiel au-delà de la crise de la Covid-19.</target>
            <g ctype="x-mc-char-format" id="201" mc:native-format-tag="&lt;w:shd w:val=&quot;clear&quot; w:color=&quot;auto&quot; w:fill=&quot;FFFFFF&quot;/&gt;" mc:smlformat="biu" />
```

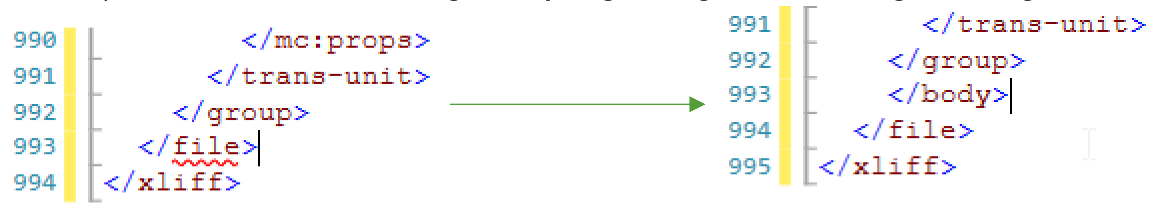
Example 2 – A typical problem where an opening **<g>** tag is placed where there should be a closing **</g>** tag, causing multiple other issues.


Start Page
Peiling productken...xlf*
Formatted text should come after <bpt> tag
Erroneous <g> tag
Duplicate <g> element where text should be
Erroneous </g> tag & missing </target> tag
Correct structure

```
          </source>
          <target state="translated" xml:space="preserve" mc:format="0-59:ff008000">
            <g ctype="underlined x-mc-char-color" id="486" mc:native-format-tag="&lt;w:color w:val=&quot;00ADEE&quot;/&gt; &lt;w:sz w:val=&quot;28&quot;/&gt;" mc:smlformat="biU">
              Corporate bonds and government bonds (including state bonds)<g ctype="bold" id="8852" mc:smlformat="Biu">
                <bpt ctype="link" id="b485" rid="r485">&lt;w:hyperlink w:anchor="_bookmark9" w:history="1"&gt;</bpt>
                <g ctype="underlined x-mc-char-color" id="486" mc:native-format-tag="&lt;w:color w:val=&quot;00ADEE&quot;/&gt; &lt;w:sz w:val=&quot;28&quot;/&gt;" mc:smlformat="biU" />
                <ept ctype="link" id="e485" rid="r485">&lt;/w:hyperlink&gt;</ept>
              </g>
              <alt-trans alttranstype="previous-version" origin="3" mc:date="20201125T212829Z" mc:user="GLOW001\U98669">
                <source xml:space="preserve" mc:format="0-62:Underline">
                  <bpt ctype="link" id="b485" rid="r485">&lt;w:hyperlink w:anchor="_bookmark9" w:history="1"&gt;</bpt>
                  <g ctype="underlined x-mc-char-color" id="486" mc:native-format-tag="&lt;w:color w:val=&quot;00ADEE&quot;/&gt; &lt;w:sz w:val=&quot;28&quot;/&gt;" mc:smlformat="biU">Bedrijfsobligaties en overheidsobligaties (inclusief staatsbon)</g>
                  <ept ctype="link" id="e485" rid="r485">&lt;/w:hyperlink&gt;</ept>
                </source>
                <target state="needs-review-translation" xml:space="preserve" mc:format="0-57:ff008000;58-60:ff000000;61-61:ff008000;62-64:ff000000;65-65:ff008000">
                  <bpt ctype="link" id="b485" rid="r485">&lt;w:hyperlink w:anchor="_bookmark9" w:history="1"&gt;</bpt>
                  <g ctype="underlined x-mc-char-color" id="486" mc:native-format-tag="&lt;w:color w:val=&quot;00ADEE&quot;/&gt; &lt;w:sz w:val=&quot;28&quot;/&gt;" mc:smlformat="biU">Corporate bonds and government bonds (including state bond///s///)</g>
                  <ept ctype="link" id="e485" rid="r485">&lt;/w:hyperlink&gt;</ept>
                </target>
              </alt-trans>
```

‘body’ start tag does not match the end tag of ‘file’

This likely means that there is a closing **</body>** tag missing before a closing </file> tag, as in this example:



```
            </mc:props>
          </trans-unit>
        </group>
      </file>
    </xliff>
```

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

<alt-trans> should have at least one <target> element.

The segment history of a segment is missing a <source> or <target> element. For example, this <alt-trans> element only contains a <source> element and the <target> element is missing:

```
<alt-trans alttranstype="previous-version" origin="1" mc:user="SYSTEM">
     <source xml:space="preserve" />
     <!-- No target element! -->
</alt-trans>
```