

Filters and File Formats in Translation Workspace

Project Engineer and Translators Reference Guide



Version 1.19.8
July 4, 2017



Contents

New and Edited	6
Updated in Translation Workspace 1.19.8 -----	6
Updated in Translation Workspace 1.19.7 -----	6
Updated in Translation Workspace 1.19.4 -----	6
Updated in Translation Workspace 1.19.2 -----	6
Updated in Translation Workspace 1.18.7 -----	6
Updated in Translation Workspace 1.18.2 -----	6
Updated in Translation Workspace 1.18.1 -----	6
Updated in Translation Workspace 1.17.7 -----	6
Updated in Translation Workspace 1.17.6 -----	6
Updated in Translation Workspace 1.17.3 -----	7
Updated in Translation Workspace 1.17.2 -----	7
Updated in Translation Workspace 1.17.1 -----	7
Updated in Translation Workspace 1.16.9 -----	7
Updated in Translation Workspace 1.16.8 -----	7
Updated in Translation Workspace 1.16.3 -----	7
Updated in Translation Workspace 1.16.2 -----	7
Supported File Formats	8
On Handling Files -----	9
ZIP Archives -----	9
Paths -----	9
Filenames -----	10
Pre-Processing Stage Filters	10
Converting to Translation Workspace RTF -----	11
Converting to Translation Workspace XLIFF -----	12
Locking Frequent Segments -----	14
Splitting and Merging XLZ Files -----	19
Using Explicit Context Match values in XLZ Files -----	19
Using explicit context match metadata in XLZ files -----	19
Behavior -----	20
Translating	21
What is TagML? -----	21
Bilingual TagML -----	21
TagML Format -----	21
Root element -----	21
Instruction lines: -----	21
Classes of tags -----	21
Preview of TagML -----	22
Understanding and Using TagML -----	22
Translating XLZ Files With Track Changes -----	22
Translating Master Repetitions -----	24
Post-Processing	25



Propagating Frequent Segments	25
Managing Warnings and Errors	27
Error situations	27
Quality Assurance Checks	28
Cleanup	28
"Cleaning" with the Translation Workspace Tools or the Word Client Plug-in	28
"Cleaning" with the XLIFF Editor	29
Converting Back to the Original Format	29
Filtering RTF Files after Translation (Cleanup and Back Conversion)	29
Filtering XLIFF Files after Translation	31
Back Converting Files with Tracked Changes	33
Markup Language File Formats (XML, HTML, SGML, XHTML, ASP, JSP)	34
Profile Selection and Auto-select	36
Actions Common to All File Types	37
Markup Language Assistant Dialog	39
Defining Parsing Rules	45
Using the Flat Tree View	48
Sample and Legend	49
Using the Hierarchical Tree View	50
Sample and Legend	50
Using the Advanced Tree View and XPath expressions	51
Sample and Legend	52
XPath Expression Editor	53
XPath Expression Order	54
Overview of XPath expressions	54
Entity Conversions	56
Text Filter	58
Supported Document Encodings	58
Configuring the Filter	58
Profile Selection	59
Configure Filter Dialog Options	60
Loading a .Imx File	61
Text Filter Profile Edit Dialog	61
The Header Panel of the Profile Edit Dialog	61
The Tree view in the Profile Edit Dialog	62
Visual Edit of the Tree	63
Context Menu	63
Preview of a Sample File	65
The Text Filter Profile Edit Bottom Panel	65
Rule Edit Dialog	66
Adding Regular Expression Items	68
Applying Rules during Parsing	69
Examples of Applying Profiles with Different Parsing Rules	69
Example 1: Two Block Sibling Rules	69
Example 2: Two Block Sibling Rules	70
Example 3: Two Delimiter Sibling Rules	71
Example 4: Two Delimiter Sibling Rules	72
Example 5: A Delimiter Rule Including a Nested (Child) Block Rule	73



Example 6: One delimiter rule with identical delimiters	73
Example 7: Making Comments Translatable in a Java Source File	74
Previewing TXT documents converted with TXT Filter	75
Using the Create New File dialog to Create XLZ files from text -----	76
Microsoft Office Files (DOC, DOCX, RTF, PPT, PPTX, XLS, XLSX)	79
DOCX Filter -----	79
Defining text as Untranslatable	79
DOCX TagML	79
Converting DOCX to RTF or XLIFF	83
DOCX Files with Tracked Changes	83
Converting DOCX to RTF or XLIFF	86
Converting back to DOCX	86
Right-to-left rendering on the DOCX file	88
Additional DOCX filter information	88
What is extracted for translation?	89
Formatting objects extracted for translation	89
Objects extracted for translation	90
Objects not extracted for translation	90
Output options	90
Previewing XLZ documents converted with DOCX Filter	90
Paragraph and Line Breaks Handling	91
Special Characters and Symbols	91
PPTX Filter -----	91
Defining Text as Untranslatable	91
PPTX TagML	91
Converting PPTX to RTF or XLIFF	93
Converting back to PPTX	94
Additional PPTX filter information	94
Text Extraction	94
Segmentation	95
Objects extracted for translation	95
Objects not extracted for translation	95
Output options	95
Previewing XLZ documents converted with PPTX Filter	96
Known Limitations	96
XLSX Filter -----	96
Defining Text as Untranslatable	96
Converting XLSX to RTF or XLIFF	97
Converting Back to XLSX	98
Additional XLSX Filter Information	99
Text Extraction	99
Segmentation	100
Specifying an Excluded Range	100
Objects Extracted for Translation	100
Objects Not Extracted for Translation	101
Output options	101
Previewing XLZ documents converted with XLSX Filter	101
XLSX TagML	101
FrameMaker and InDesign Files (MIF, IDML, INX)	104
MIF Filter -----	104



Defining text as Untranslatable	104
MIF TagML	105
Converting MIF to RTF or XLIFF	108
Converting back to MIF	108
Additional MIF filter information	109
Character Conversion	109
Text Extraction	110
Segmentation	110
Output options	111
Page Breaks	111
Default setting	111
IDML Filter -----	111
Defining Text as Untranslatable	111
Converting IDML to RTF or XLIFF	112
Converting Back to IDML	113
Additional IDML Filter Information	114
INX Filter -----	117
Defining text as Untranslatable	117
INX TagML	117
Converting INX to RTF or XLIFF	118
Converting back to INX	119
Additional INX filter information	119
Paragraph and Line Breaks treatment	119
Entities management	119
Inline graphics	119
Embedded graphics	120
Hidden and locked layers	120
Industry standards (TTX and XLIFF)	121
TTX to RTF Conversion Dialog -----	121
Translation Workspace RTF / XLIFF to TTX Conversion Dialog -----	122
XLIFF 2.0 to XLZ Conversion Dialog -----	124
XLIFF 2.0 Translation Candidates module	125
Translation Workspace XLZ to XLIFF 2.0 Back Conversion Dialog -----	126
Translation Workspace RTF to XLZ Conversion Dialog	127
Translation Workspace XLZ to RTF Conversion Dialog	128



New and Edited

Paragraphs which include previously existing, but for the latest version edited content, have a greyish background color and a dotted red border, as this paragraph does.

Paragraphs which have been added to the latest version of this guide have a yellowish background color and a green dashed border, as this paragraph does.

Removed text and figures and changes in layout only are not highlighted.

Updated in Translation Workspace 1.19.8

The following passage was updated to this guide for Translation Workspace 1.19.8.

The [Translating XLZ Files With Track Changes](#) topic.

Updated in Translation Workspace 1.19.7

The following passage was added to this guide for Translation Workspace 1.19.7.

The [Using the Create New File dialog to Create XLZ files from text](#) topic.

Updated in Translation Workspace 1.19.4

The following passages were updated in this guide for Translation Workspace 1.19.4.

The [MIF Filter](#) topic.

Updated in Translation Workspace 1.19.2

The following passages were added to this guide for Translation Workspace 1.19.2.

The [Previewing TXT documents converted with TXT Filter](#) subtopic.

Updated in Translation Workspace 1.18.7

The following passages were added to this guide for Translation Workspace 1.18.7.

The [ZIP Archives](#) subtopic.

Updated in Translation Workspace 1.18.2

The following passages were updated in this guide for Translation Workspace 1.18.2.

The [Filtering XLIFF Files after Translation](#) subtopic.

Updated in Translation Workspace 1.18.1

The following passages were added in this guide for Translation Workspace 1.18.1.

The Document Cleanup dialog image in the [Cleanup](#) topic.

The [XLIFF 2.0 Translation Candidates module](#) subtopic.

Updated in Translation Workspace 1.17.7

The following passages were updated in this guide for Translation Workspace 1.17.7.

The [Locking Frequent Segments](#) topic.

Updated in Translation Workspace 1.17.6

The following passages were updated in this guide for Translation Workspace 1.17.6.

The [XLIFF 2.0 to XLZ Conversion Dialog](#) topic.

The [Translation Workspace XLZ to XLIFF 2.0 Back Conversion Dialog](#) topic.



Updated in Translation Workspace 1.17.3

The following passages were updated in this guide for Translation Workspace 1.17.3.

The [Previewing XLZ documents converted with DOCX Filter](#) topic.

The [Previewing XLZ documents converted with PPTX Filter](#) topic.

The [Previewing XLZ documents converted with XLSX Filter](#) topic.

Updated in Translation Workspace 1.17.2

The following passages were added to this guide for Translation Workspace 1.17.2.

The [Previewing XLZ documents converted with DOCX Filter](#) topic.

The [Previewing XLZ documents converted with PPTX Filter](#) topic.

The [Previewing XLZ documents converted with XLSX Filter](#) topic.

Updated in Translation Workspace 1.17.1

The following topics were updated in this guide for Translation Workspace 1.17.1.

The [Locking Frequent Segments](#) topic (the description for the [Allow Tag Replacement](#) option).

Updated in Translation Workspace 1.16.9

The following topics were updated in this guide for Translation Workspace 1.16.9.

The [Additional IDML Filter Information](#) topic.

Updated in Translation Workspace 1.16.8

The following topics were added in this guide for Translation Workspace 1.16.8.

The [Using Explicit Context Match values in XLZ Files](#) topic.

The [XLIFF 2.0 to XLZ Conversion Dialog](#) topic.

The [Translation Workspace XLZ to XLIFF 2.0 Back Conversion Dialog](#) topic.

The following topics were updated in this guide for Translation Workspace 1.16.8.

The [Additional IDML Filter Information](#) topic.

The [Supported File Formats](#) topic: XLIFF 2.0 was added to the supported formats table.

Updated in Translation Workspace 1.16.3

The following topics were updated in this guide for Translation Workspace 1.16.3.

The [MIF Filter](#) topic.

Missing supported Excel file formats were added to two pages in this document (not highlighted).

Updated in Translation Workspace 1.16.2

The following topics were updated in this guide for Translation Workspace 1.16.2.

The [MIF Filter](#) topic (the first **Note**).



Filters and File Formats in Translation Workspace

This guide describes the file formats supported by Translation Workspace and the filters for converting these formats to and from the translation formats (RTF and XLIFF) used by translators in the Translation Workspace editors: the Microsoft Word Plug-in and the XLIFF Editor.

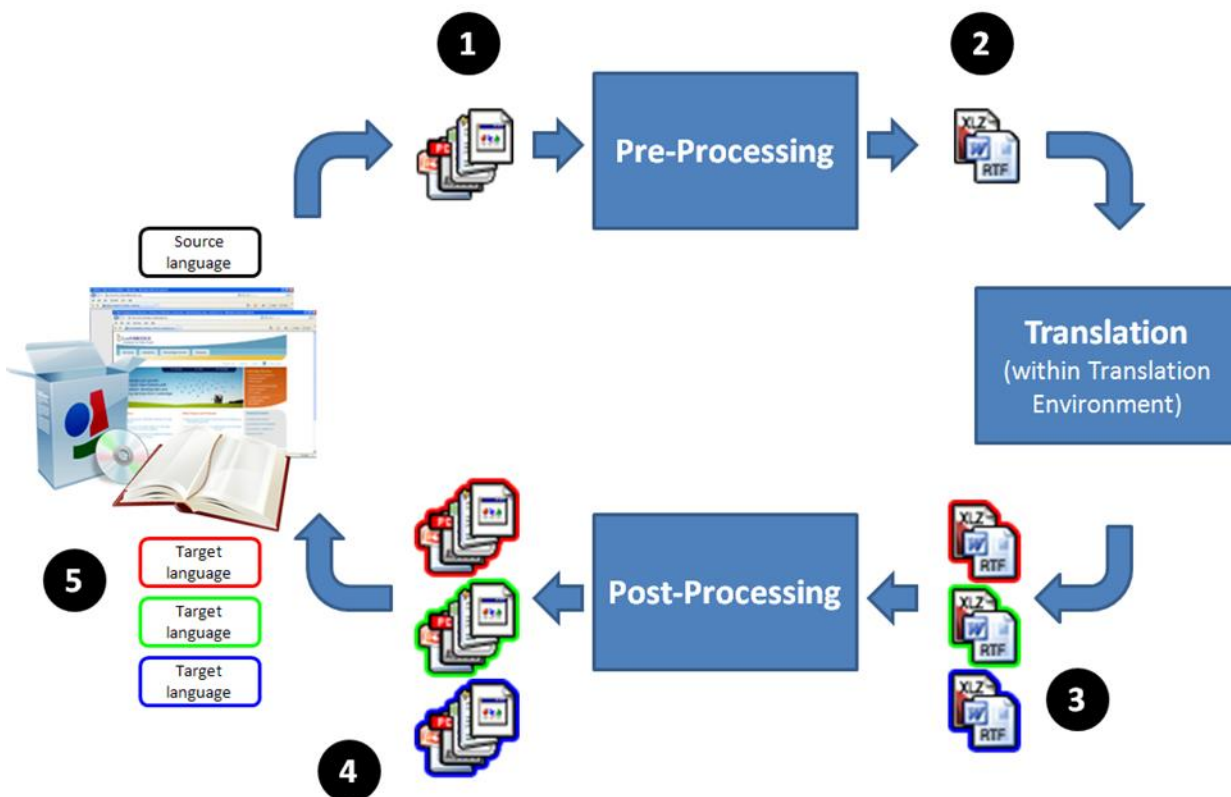
Supported File Formats

Translation Workspace provides built-in filters to convert the most popular file formats (MIF, IDML, INX, DOCX, PPTX, XLSX) and markup language file types (HTML, XML, SGML) into translatable format (RTF or XLIFF) and back to the original format. The following table lists the supported formats, the filters for getting them to a translation format, the Translation Workspace dialog in which the filter is found, and the recommending Translation Workspace editor for translating them:

Format	Filter (Conversion dialog)	Recommended Editor
Rich Text Format (RTF)	Translation Workspace RTF (Convert External Styles to Translation Workspace dialog)	Microsoft Word Plug-in
Microsoft Word 2000-2003 (DOC, DOT)	Translation Workspace RTF	Microsoft Word Plug-in
Microsoft Word 2007-2013 (DOCX, DOTX, DOCM, DOTM)	Translation Workspace DOCX (Document Conversion dialog, Convert External Styles to Translation Workspace dialog)	XLIFF Editor
Microsoft PowerPoint 2007-2013 (PPTX, PPTM, POTX, POTM, PPSX, PPSM)	Translation Workspace PPTX (Document Conversion dialog)	XLIFF Editor
Microsoft Excel 2007-2013 (XLSX, XLAM, XLSM, XLTM, XLTX)	Translation Workspace XLSX (Document Conversion dialog)	XLIFF Editor
InDesign CS4, CS6 (IDML)	Translation Workspace IDML (Document Conversion dialog)	XLIFF Editor
InDesign CS2, CS3 (INX)	Translation Workspace INX (Document Conversion dialog)	XLIFF Editor
FrameMaker versions 5-12 (MIF)	Translation Workspace MIF (Document Conversion dialog)	XLIFF Editor
Markup Languages (XML, HTML, SGML, XHTML, ASP, JSP) and profiles (HTML, HHC, HHK, RESX)	Translation Workspace XML/HTML (Document Conversion dialog)	XLIFF Editor
Trados TTX	Translation Workspace TTX to RTF (Document Conversion dialog, TTX to RTF Conversion dialog)	Microsoft Word Plug-in
Trados TTX (pre-segmented)	XLIFF Editor	XLIFF Editor
Idiom XLIFF (XLZ, XLIFF)	XLIFF Editor (Idiom XLIFF Converter dialog)	XLIFF Editor
OKAPI generic XLIFF	XLIFF Editor	XLIFF Editor
XLIFF 2.0 files	XLIFF 2.0 to Translation Workspace XLZ (XLIFF 2.0 to XLZ Conversion dialog)	XLIFF Editor
Text files (for example, plain text, properties and INI files)	Translation Workspace TEXT (Document Conversion dialog)	XLIFF Editor



The typical translation process has three basic states: Pre-Processing, Translation, and Post-Processing.



The Translation Workspace filters are used during the Pre-Processing stage to convert the files to a translatable format—either Translation Workspace RTF or XLIFF. Filters are also used to convert the translated files back to their original format in the Post-Processing stage.

On Handling Files

ZIP Archives

Translation Workspace can, in input and output, handle ZIP files containing more than 65,535 files, if the size of each file in the ZIP and the size of the ZIP archive itself is maximum 2 GB.

Paths

Windows XP and newer Windows versions prevent you from creating folder+filename paths longer than 260 characters (the file extension included). Such paths can still exist in Windows file system as a result of copying folders into folders or by extracting compressed files (e.g. ZIP). However, the Windows APIs can normally recognize maximum 260 character paths, and there is a 259-character limitation on creating and saving files in the MS Office products. For MS Excel the limitation is 218 characters (including file name extension).

Warning: When the path is longer than 259 characters, several operations of Translation Workspace Word Plug-in, XLIFF Editor and TW Tools (Conversions, Document Analysis, Pretranslation, TM Import, Cleanup) are at risk for not performing as expected. The Translation Workspace client may then be unable to create the needed temporary files.



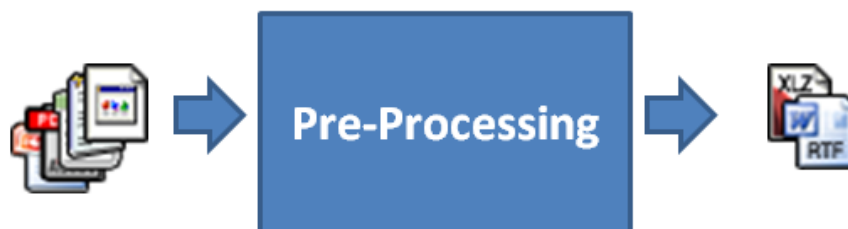
Even when a ZIP file name path is less than 259 characters, the paths of the files inside the ZIP file can include more than 259 characters. If a whole folder structure is compressed into the ZIP file, the ZIP file can itself contain file paths with more than 259 characters. When processed by Translation Workspace files inside such ZIP files may not be processed as expected.

Filenames

A same kind of issue exists with capitalization of filenames in Windows. It is usually not possible to create a file that has the same name as another file except for capitalization. But there are ways by which it can be done and if you end up with two such files you must forever use the exact right capitalization to be able to access any of the files.

Pre-Processing Stage Filters

Your first step with your files is to get them into the Translation Workspace intermediate translation format you will use—RTF or XLIFF.



Note that these RTF and XLIFF formats are specific Translation Workspace variants of RTF and XLIFF, with special tags for using the Translation Workspace commands.

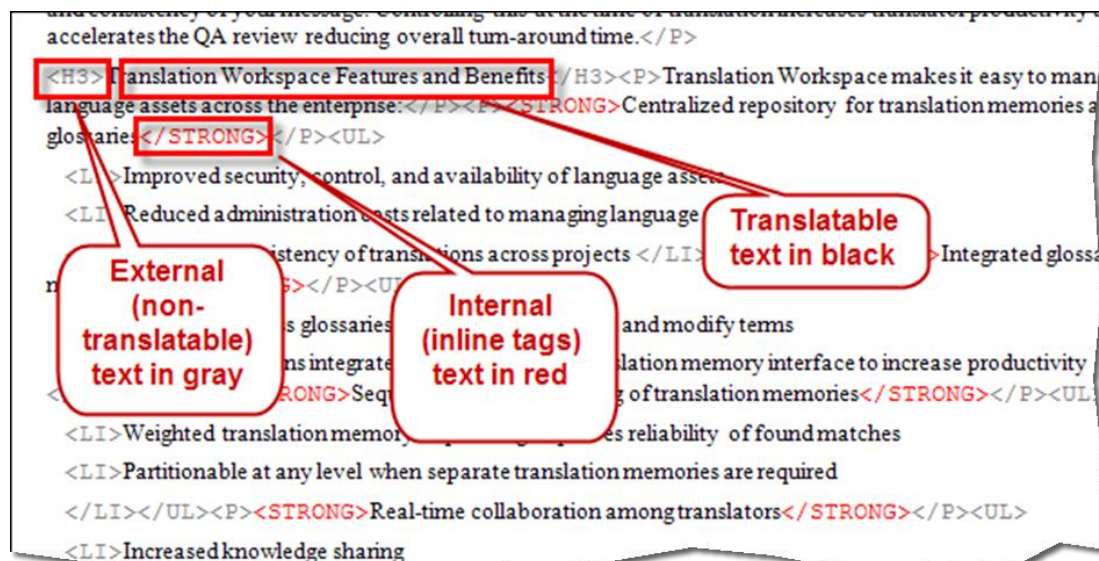
The filters used to convert the files to the Translation Workspace translation (or “pivot”) formats do the following:

1. Convert the source to format editable by the Translation Workspace editors (the Microsoft Word Plug-in or XLIFF Editor).
2. Separate content into what is translatable and what is not translatable (and what is “internal”).
3. Divide-up translatable content into translatable-sized chunks (called “segments”).



Converting to Translation Workspace RTF

RTF uses text formatting to handle parsing, to define the first level of segmentation, to store the bilingual content, and to store other metadata used in the Computer-Aided Translation (CAT) process.



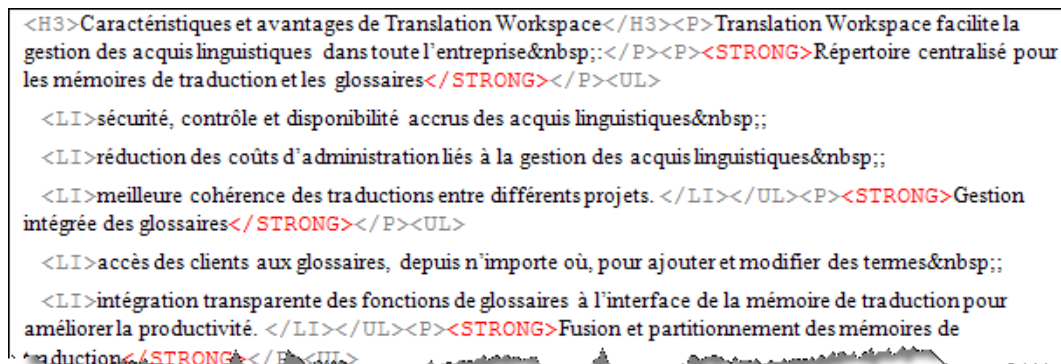
The RTF filter parses text into external (not translatable—indicated as gray), translatable (indicated as black), and internal (text that can't be translated, but can be repositioned and resides within not translatable text—indicated as red).

Content is segmented according to your segmentation rules for the associated TM (which can be delimited by any of these delimiters: tabulator, newline, paragraph, ordinals, middle initials, full stop, question mark, exclamation mark, colon, semicolon, or others you add).

Blocks of translatable text between blocks of external text automatically become segments.

A bilingual segment—one that contains both the original source text and the translated text—is stored in the RTF using a combination of tags and hidden text. The original source text and the tagging syntax are marked as hidden text, where the translated text is visible text.

When the passage above is translated into French, the source segments are hidden by default:



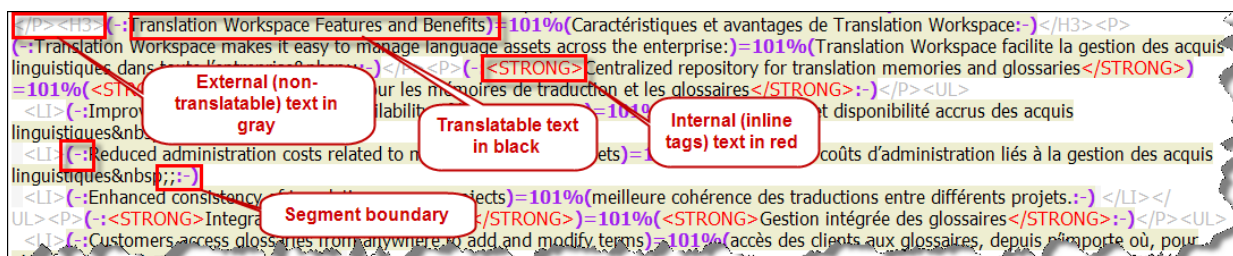
If you toggle the Show/Hide button to Show, the source segments are displayed as well:



The filters for converting a file to Translation Workspace XLIFF format create an XLZ package, which is a “zip” file comprising a skeleton (SKL) file, to maintain the structure so you can generate a target file at any time in the process, and an XLF file that contains the segments, which are bounded within `<trans-unit>` elements, inside of which are `<source>` and `<target>` elements. Internal content is handled with special tags.

Here is a sample from an XLIFF file viewed in a text editor:

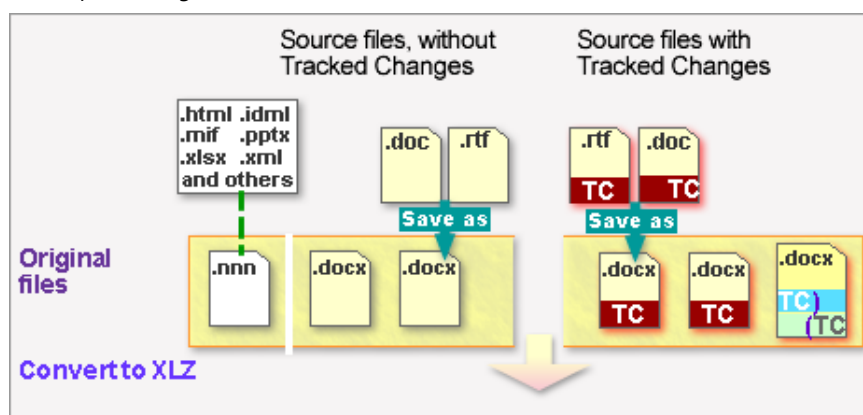
The XLIFF Editor hides these underlying structural tags, presenting (in black) the translatable text, (in gray) the untranslatable external tags, and (in red) the internal tags, which can be moved but not modified.



DOCX Files with Track Changes

DOCX files with track changes can be translated in Translation Workspace.

Original RTF files need to be saved to DOCX. You should save any original RTF files with or without track changes to DOCX format with MS Word before you start processing the files. Direct conversion of RTF files to XLZ accepts all tracked changes in the files and disables tracking during further processing of such files.



Files converted to RTF are not supported. You cannot process RTF files with track changes process without losing the tracking. Files have to be converted to XLZ with the DOCX filter from DOCX to enable the tracking and retain the tracked changes during processing.

Original DOCX and DOC files can include tracked changes that are retained during further processing when converted to XLZ. The DOC files have to be saved to DOCX files before converting to XLZ with the DOCX Filter.

Original DOC and DOCX files can also be already (fully or partially) segmented and they can include tracked changes in both, the source and the target segments.

All other original file types (PPTX, XLSX, IDML, MIF...), (.nnn) are converted with the appropriate filter for those files. Tracked changes can be included first when the XLZ file is edited in XLIFF Editor with Tracked Changes ON, but all changes are accepted and changes are not tracked in the final document generated from the XLZ.

Regardless of the original file type, you can set XLIFF Editor to track changes that are made to the target segments while translating XLZ files.

Tracked changes are formatted as blue underlined text for insertion and ~~green strike through~~ text for deletion in the XLZ files. The same formatting is used for changes made by all the users.

```
size="10" complexscriptsize="12"> This first segment includes additions added text and text removals.  
This second segment does has not been have any changes in it edited by two different users.  
<commentrange start number="14"/>1This segment is <cf bold="on">2a normal</cf>2 segment, which  
was <cf italic="on">3a bit different</cf>3 before it was EDITED edited.<commentrange end number="14"/>  
<commentreference style="CommentReference" number="14"/>5</p>
```



Locking Frequent Segments

With **Lock Frequent Segments** function you can lock the frequently repeated segments in XLZ and XLF files before processing the files further. With **Propagate Frequent Segments** function you can propagate the translations to the frequently repeated segments (locked by the Lock Frequent Segments function), after the XLZ and XLF files have been otherwise processed.

No tag replacement is performed during the **Lock Frequent Segments** operation. Only segments for which the number, position, and the contents of the tags match are identified as repetitions.

Note: **Lock Frequent Segments** and **Propagate Frequent Segments** functionality are available only in Word Client and TW Tools. XLIFF Editor cannot perform the Lock and Propagate processes.

You have login to a Translation Workspace server when launching these two functions, but selecting a TM and a language pair is not required.

Files with Tracked Changes

Running the **Lock Frequent Segments** operation for files with track changes is currently **blocked**. An error message is displayed if you run the operation for such files.

Lock Frequent Segments

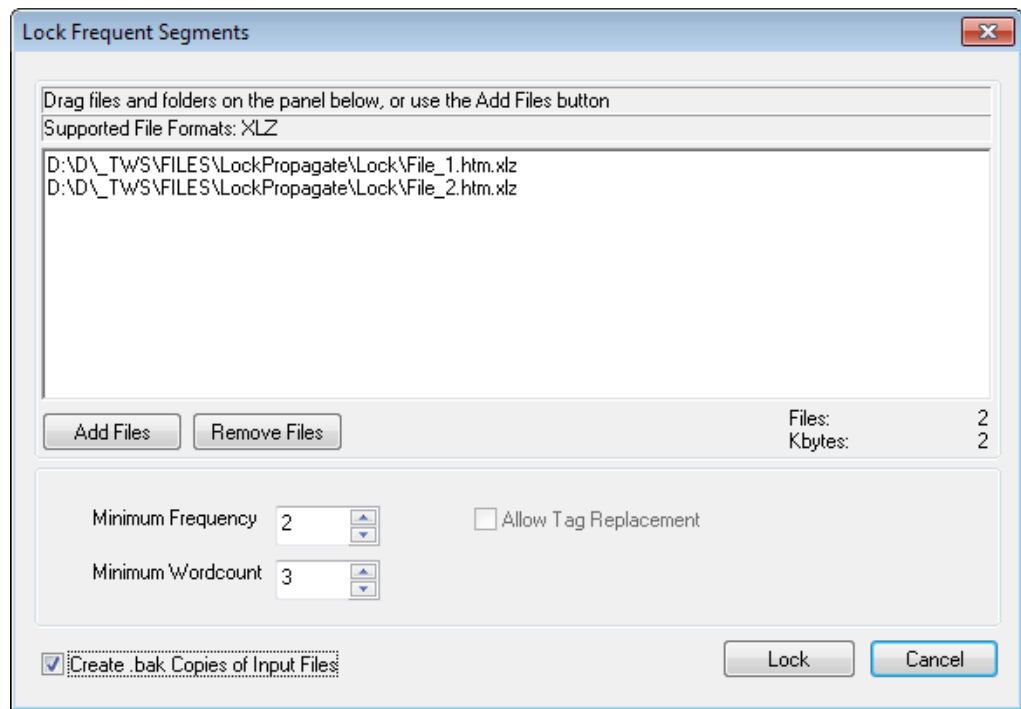
With the **Lock Frequent Segments** function you can lock the frequently repeated segments in a batch of XLZ and XLF files before the files are otherwise processed. The XLZ and the XLF files can be either previously non-processed or already pretranslated files.

Note: In XLIFF Editor you cannot expand or shrink segments which have the master/repetition information coming from the Lock & Propagate functionality..

To lock the frequent segments on a batch of XLZ and XLF files:

1. Drag and drop (or click **Add Files** to add the files) the set of XLZ and XLF files onto the Lock Frequent Segments dialog.
2. Adjust the values for the **Minimum Frequency** and **Minimum Wordcount** options, if needed.
3. Define whether you want backups (.bak) of the input files created. This option is checked by default.
4. Click **Lock**.

The Lock Frequent Segments dialog is accessed from the **Lock Frequent Segments** link at **Start > All Programs > Translation Workspace > Tools > Translation Workspace Tools**. Or, in the Word Client Plug-in, you can access it at **Documents > Lock Frequent Segments**.



When you add files or folders containing the files the Translation Workspace client checks if the documents you chose are of the correct type (XLZ and XLF) for conversion and only displays XLZ and XLF encoded documents.

This dialog includes the following options:

Option	Description
Add Files	You may click on this button to add a number of files through the system's file open dialog. The same restrictions apply on the file type as explained above.
Remove Files	Removes all files from the dialog. Does however not remove or delete files from your disk.
Minimum Frequency	By adjusting the value in this field you may set the minimum frequency of the segments to be considered as to be processed. The default value is 2. If the value is set to for example 4, only those segments appearing in the files to be processed at least 4 times, are locked.
Minimum Wordcount	By adjusting the value in this field you may set the minimum wordcount of the segments to be considered as to be processed. The default value is 3. If the value is set to for example 5, only those segments appearing in the files to be processed which have at least 5 words, are locked.



Note: This option does not work for languages which do not separate words by spaces (mainly double-byte languages).

Allow Tag Replacement	By checking this option you enable tag replacement during the Lock Frequent Segments operation. When the option is not checked, no tags, acronyms and numbers are substituted while the segments are locked. Currently this option cannot be checked by the users and no tag replacement is performed during the locking operation. For segments to be identified as repetitions, the number, position, and the contents of the tags should match..
Create .bak Copies of Input Files	The Lock Frequent Segments process replaces the selected input files. If you want to keep the old files as well, you should check this box. An old file “input.xlz” or “input.xlf” is then renamed to “input.xlz.bak” or “input.xlf.bak” and will not be overwritten with the new file.
Lock	Clicking this button starts the conversion. You will see a progress dialog, and after the conversion of all documents is done the main dialog stays open.
Cancel	Closes this dialog without further action.

Options defined on the **Lock Frequent Segments** dialog are stored and preserved between sessions.

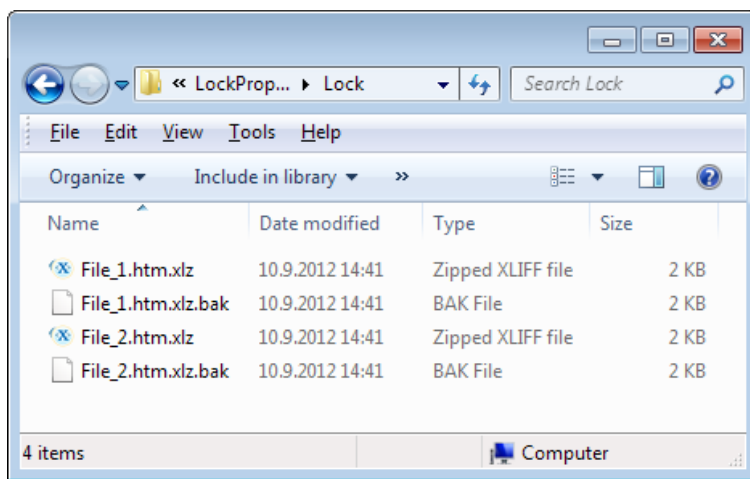
Utilizing Context Matches

If a locked frequent segment is unlocked and translated before the propagation phase, its target is overwritten by the Propagate Frequent Segments process.

To utilize the context matches provided by the TM, you should pretranslate the XLZ and the XLF files before the Lock Frequent Segments operation is performed.

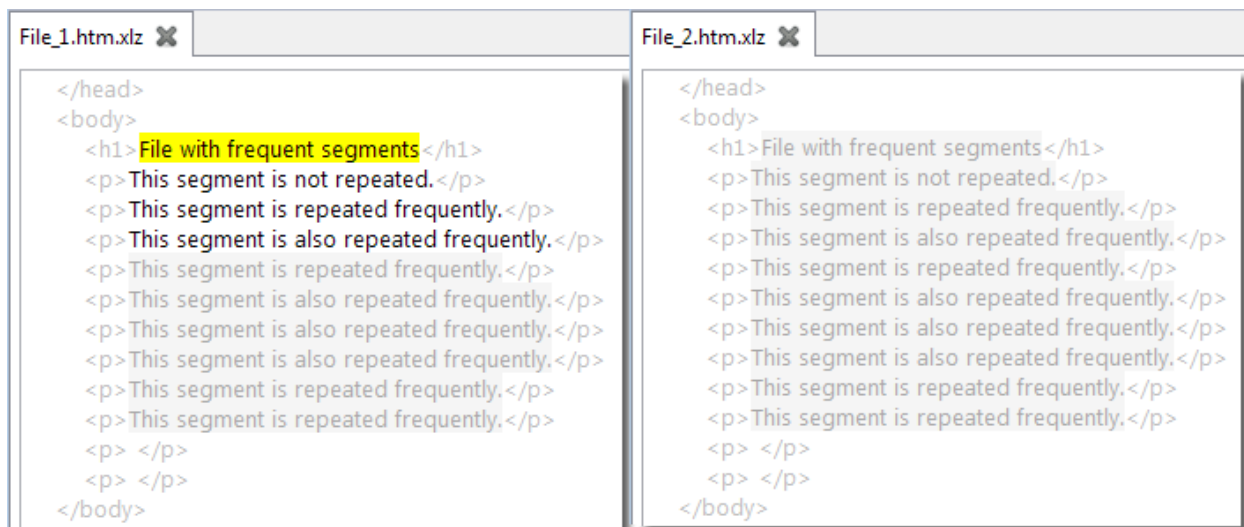


Results



If **Create .bak Copies of Input Files** was checked in the Lock Frequent Segments dialog, the .bak extension is added to the original files and the processed files are created into the same folder.

After the Lock Frequent Segments has been run to a batch of files, the locked segments are grayed in XLIFF Editor. As the segments of the files (run with the default Lock Frequent Segments settings) in the picture below are identical, all the segments in the second file are locked.



The first occurrence of the frequently repeated segment is treated as the *master repetition*. The master repetitions are available for translation in the files.

Other occurrences of the repetitions are locked (set to `translate=no`) in the XLZ and the XLF files. The locked frequent segments are not segmented during pre-translation or interactive translation.

Already translated segments existing in the files are ignored during the lock process, and they are **not**:

- ❑ Counted in the number of repetitions.
- ❑ Considered as master translations.
- ❑ Considered as repetitions.



Lock Frequent Segment functions can be performed several times on the same or different batch of files. Every time, the lock process is restarted from the beginning. Repetitions are first unlocked.

Managing Warnings and Errors

When an error occurs during the Lock Frequent Segments function (due to corrupted XLZ or XLF files), an error is returned in the Lock Frequent Segments dialog, and error (.err) file is created with the same name and path as the corresponding source XLZ or the XLF file.

Note: No global log file is currently created with a summary of all the errors.



Splitting and Merging XLZ Files

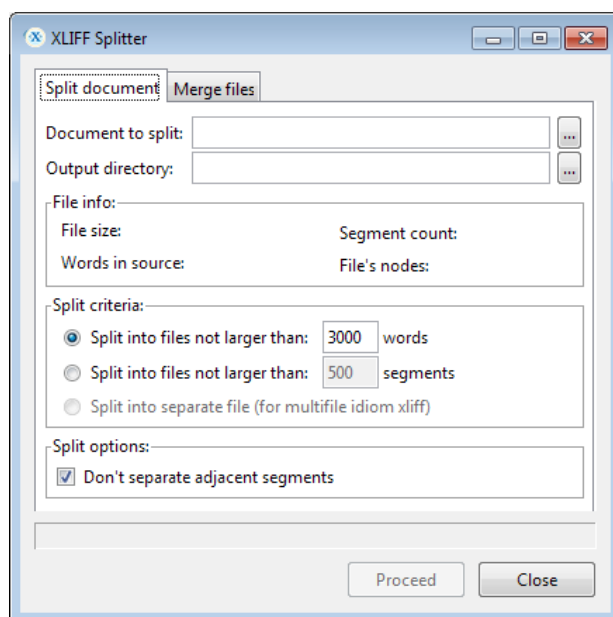
XLIFF Editor has an in-built plug-in (Tools menu), which can be used for splitting large XLIFF files and merging the files back to one file.

You can split any large XLZ file to smaller files, if the XLZ file has been created by the Translation Workspace conversion filters or the XLIFF Editor Idiom plug-in.

You can also split XLZ files that are partly or fully translated. The files to be merged can also be partly or fully segmented XLZ files.

Splitting can be performed by using three different options:

- Split by the number of words.
- Split per segment count.
- Split to separate files (for XLZ files converted and combined from multiple Idiom XLIFF files)



There is also an option to keep segments not separated by any external content together in the same file.

See XLIFF Editor help for more detailed instructions.

Using Explicit Context Match values in XLZ Files

When the Context Matching feature is enabled in the TM, Translation Workspace normally saves a checksum value, based on the immediate surrounding source segments of the current source segment, for the current segment into the TM along with the Translation Unit. The code is then used for identifying exact matches having the same context as the segment currently being processed in a file. (For more information on context matching see the Translation Workspace Web Interface help.)

Using explicit context match metadata in XLZ files

For XLZ files, you can also apply an explicit method for identifying exact matches for which the context match value is not counted from the surrounding segments. This requires that there is already a some kind of string specific identity data available in the files, usable for creating context id values for segments. The primary example for such identity data is a software string ID code.

Identifying the context this way does not take into account the surrounding segments. Context Matching feature needs to be enabled.

Enabling the explicit context-mode

To enable the explicit context mode you need to add the `logoport:metadata` element between the header and the body element of the file to be processed. Inside this metadata element you also add the `<logoport:metaGroup category="liox:match-options">` group, including the `<logoport:meta type="context-mode">` element in it (see below).



You can enable the explicit context mode by including the `explicit` value in the `context-mode` type meta element inside the `logoport:metadata` element. Changing the value to `counted` activates the default context match mode (which creates the match codes based on the surrounding source segments) instead.

```
</header>
<logoport:metadata>
  <logoport:metaGroup category="liox:match-options">
    <logoport:meta type="context-
mode">explicit</logoport:meta>
  </logoport:metaGroup>
</logoport:metadata>
<body>
```

Adding the context CRC value to segments

You can add the context CRC value to the segments by adding the `logoport:context-id` attribute into the `trans-unit` element and applying a value for it.

```
<trans-unit translate="yes" id="1" logoport:context-id="861105140">
  <source>Hello World!</source>
</trans-unit>
```

You need to add the values by using a script or some other means. The *Context Match CRC* should be a 32Bits Integer. If it does not match the criteria, an error message is generated for all batch functions (analysis, auto-translate, cleanup).

Behavior

TM Store interactive and batch functions store the **CRC** value into the TM **NBH** (Context Match) attribute.

TM Lookup interactive and batch functions use the **CRC** value to identify a context match.

- The segment calculated *Context Match CRC* and TM match **NBH** attribute are compared and a 101% match is provided when they are found identical.
- A segment with an identical *Context Match CRC* value as a previously stored segment in the TM is provided with a 101% match by *Pre-translation* and *Analysis*.

For XLZ Translation Units without any *Context Match CRC* value, the *Context Match NBH* value is calculated based on surrounding segments.

Warning: A segment with the **CRC** value set to "0" in the XLIFF Editor is saved to the TM without any NBH attribute. When retrieved, the TM provides the segment as a 100 % match.



Translating

If the files you are translating are DOCX, PPTX, XLSX, IDML, INX, or MIF, when you convert them to the Translation Workspace RTF or XLIFF pivot formats, they will contain a tagging system called TagML.

What is TagML?

TagML (tagged XML) is an intermediate XML format produced by our filters to represent content extracted from the native file format (DOCX, PPTX, XLSX, IDML, INX, or MIF). It is this “intermediate” format converted to the RTF or XLIFF that the translator works on to translate, using either the Word Client Plug-in or XLIFF Editor.

Bilingual TagML

DOCX, PPTX, XLSX, IDML, INX, and MIF converters can do forward and backward conversions. Forward conversion tags a source file, generating TagML in either RTF or XLIFF representations. Then, translators work with TagML, translating the source language into the target language, resulting in a bilingual file. Backward conversion contains two steps: cleanup and conversion. In the cleanup step, the source language content is stripped out, leaving the target language only. Bilingual RTF or XLIFF becomes plain XML. In step two, a localized file is generated based on the structure from the original file and new content from the TagML, leaving plain XML format in the target language only.

TagML Format

TagML has a hierarchical structure.

Root element

TagML is an XML application. The root element is <TagML>.

Instruction lines:

The first two lines are instruction lines. The first one gives the encoding, the second one gives the stylesheet XSL file. For example,

```
<?xml version='1.0' encoding='utf-8' standalone='yes'?>
<?xml-stylesheet type="text/xsl" href="ooxmlpreview.xsl"?>
```

Classes of tags

Tags used in TagML fall into two classes: **structural** and **constructable**, depending on whether they are hierarchically above or below the paragraph tag.

Structural refers to structural elements of the native file, like tables and frames. This Structural element will be in the “External Style” (**LogoportDoNoTranslate**) in the RTF pivot format. It appears in gray and is part of the skeleton file in XLIFF Editor), and is not editable. Structural tags provide context to translators and are used to regenerate the native file format.

Constructable tags refers to inline tags (or placeables) elements which are in “Internal Style” (**LogoportMarkup**) in the RTF pivot format (and appears in red, as tags in XLIFF Editor). These inline tags are part of the translatable text. Some only can be added/removed/modified, but all can be moved.

In a backward conversion, the <p> (paragraph) tag and its dependents (inline tags) are reconstructable. Translators can access them and add/remove/modify the paragraph content and format that will finally appear in localized file.



All tags hierarchically above the <p> (paragraph) tag are included to preserve the document structure. For example, the <table> tag is a structural tag, where each table cell contains one or more paragraphs.

Preview of TagML

When translators work on a document that has TagML, they can preview the current translation state. Preview splits the bilingual TagML into two plain XML files, one with the source language, and the other with the target language. Templates in XSLT use both structural and constructable tags to give the comparison shown in the browser.

Understanding and Using TagML

A description of the TagML tags for each of the five formats (DOCX, PPTX, XLSX, INX, and MIF) is given in the section below for each format.

Translating XLZ Files With Track Changes

You can translate and edit files with track changes in the *Inline UI* of XLIFF Editor.

Note: The *Classic UI* and the *Grid UI* do not support translating files with track changes.

Depending on what is the original file format and how the conversion was done, the prerequisites for handling files differ a bit.

As changes tracking is not supported in TMs, during some phases of processing (Analysis, Unknown Segments Analysis, Pretranslation, and TM Update) the tracked changes are treated *as if they had been accepted*, but the changes history and the tracking are still retained in the files. The same applies to XLIFF Editor **Spell Check** and **QA Check**, which check the quality of the target segments.

Allowing the Revision Tracking

You need to allow revision tracking for the XLZ files in XLIFF Editor by selecting **Tools > Allow Revision Tracking**.

For XLZ files converted from DOCX files with the DOCX Filter while the **Convert Tracked Changes (only for XLZ format)** option is checked in the DOCX Filter, **Allow Revision Tracking** is already enabled by default for the converted XLZ files.

For files converted to XLZ in any other way, you need to select **Tools > Allow Revision Tracking**. Before you can allow revision tracking (access the function), you have to first open any XLZ file with tracked changes in the source, and set the *User Id* by selecting **Track Changes > Track Change Options**.

Note: You cannot *Sub segment* segments in files for which the revision tracking has been enabled.

Note: You cannot *Expand* and *Shrink* segments in files for which the revision tracking has been enabled.

Setting the User ID

To be able to use track changes in any XLZ file, you need to set a User ID in XLIFF Editor.

To set the User ID:

1. Open an XLZ file with tracked changes in XLIFF Editor.
2. Select **Track Changes > Track Change Options**.
3. The Track Changes Options dialog is displayed, in which you can set the Track Changes User ID. This User ID is displayed for the tracked changes in target segments.



Currently, for a newly installed XLIFF Editor, the only way to set the user id is to open a XLZ file with track changes in it. To set the User Id select **Track Changes > Track Changes Options**.

Enabling Track Changes

After you have set a *Track Changes User ID*, track changes are enabled for all files for which the revision tracking is allowed in **Tools > Allow Revision Tracking**.

You can disable and enable the track changes in the active document:

By pressing **Ctrl+Alt+F9**,

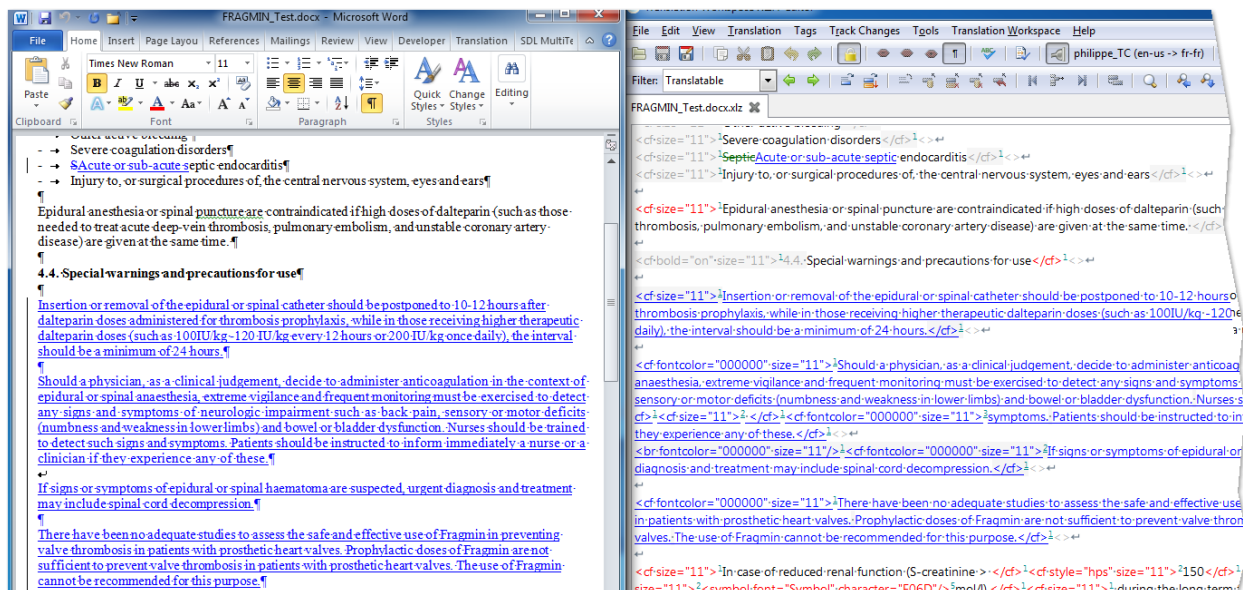
By selecting **Tools > Track Changes**, or

By clicking **Turn On/Off Track Changes** on the Track Changes Toolbar.

Conversion Result

You can see the similarities in the original DOCX file with tracked changes and the converted XLZ file below.

Note: Opening large XLZ files or running some actions on them may take time.



Tracking: The tracked insertions and deletions in the original DOCX file are displayed as tracked insertions and deletions in the XLZ file. A revision is created only if you change the text. Changes in **state**, **state qualifier** or **match percent** do not create a revision.

Warning: Sometimes the tracked changes are displayed differently in the original DOCX, the converted XLZ, and the back-converted DOCX.

Segmentation: Sentences and other text sequences treated as segments by Translation Workspace, which tracked changes merge or split in DOCX files, are segmented in the XLZ file based on the sequencing of the original text. For example, one sentence divided to two sentences by tracked changes in DOCX is segmented as one in XLZ after the conversion.

Whole segment insertions: When the whole segment or a group of segments has been tracked as an insertion in the DOCX file, it is also displayed as a tracked insertion in the XLZ file.

Whole segment deletions: When the whole segment or a group of segments has been tracked as a deletion in the DOCX file, it is also displayed as a tracked deletion in the XLZ file.



Comments: Comments contained in the DOCX are displayed as normal text segments in XLZ, at the end of the document.

Formatting: Tracked changes are formatted as blue underlined text for insertion and ~~green strike through text~~ for deletion in the XLZ files. The same formatting is used for changes made by all the users.

Tooltips: As in the DOCX file, the changes include information which can be displayed. As you point at a changed text in a segment in the XLIFF Editor, a tooltip displays whether the change was an insertion or deletion, the user who made the change and the timestamp when the change was made.

Note: When two different users have added tracked changes in MS Word to the same segment within the same minute, only one user is reported but all changes exist in the segment. This is replicated in XLIFF Editor.

Translating Master Repetitions

After locking the frequent segments in the XLZ files (see [Locking Frequent Segments](#) on page 14), the files can be otherwise processed, like analyzed, pretranslated and translated. The master repetitions are translated along all the non-frequent translatable segments in the files.

The locked segments are not processed during this phase.

After translation and possible other processing, the frequent segments are propagated to the locked segments. See [Propagating Frequent Segments](#) on page 25.



Post-Processing

After you have translated your document, you need to check the translation, propagate the translations from the master repetitions to the locked frequent segments, clean the file (by storing your translations in the TM and stripping the source segments, leaving just the translated file), and finally convert the file back to its original format.

Propagating Frequent Segments

With **Propagate Frequent Segments** function you can propagate the translations to the locked frequently repeated segments from the master repetition in a batch of XLZ files after the files have been otherwise processed.

Files with Tracked Changes

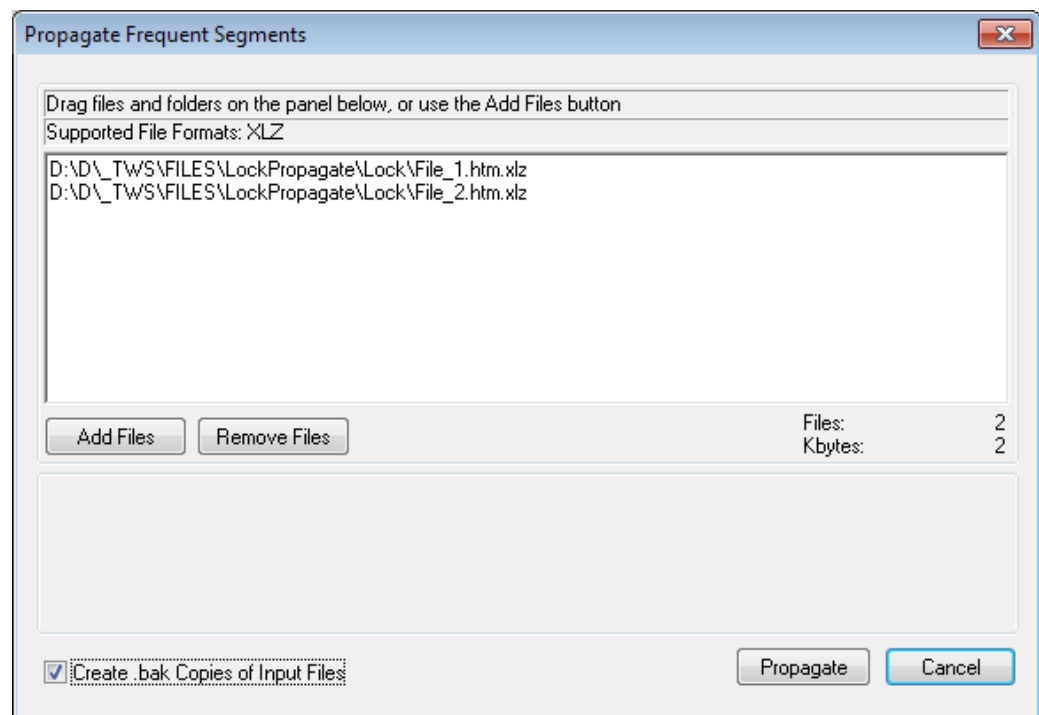
Running the **Lock Frequent Segments** operation for files with track changes is currently **blocked**. An error message is displayed if you run the operation for such files.

Propagation Process

To propagate the frequent segments to a batch of XLZ files:

1. Drag and drop (or click **Add Files** to add the files) the set of XLZ files onto the Propagate Frequent Segments dialog.
2. Define whether you want backups (.bak) of the input files created. This option is checked by default.
3. Click **Propagate**.

The Propagate Frequent Segments dialog is accessed from the **Propagate Frequent Segments** link at **Start > All Programs > Translation Workspace > Tools > Translation Workspace Tools**. Or, in the Word Client Plug-in, you can access it at **Documents > Propagate Frequent Segments**.



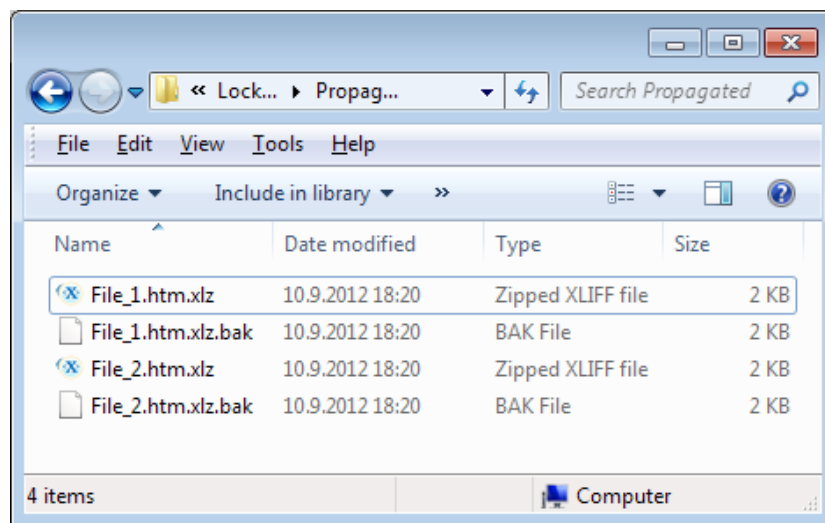


Drag the files or folders containing the files you want to process on the dialog box. The Translation Workspace client checks if the documents are of the correct type (XLZ) for conversion and only displays XLZ encoded documents.

The following options are included in this dialog:

Option	Description
Add Files	You may click on this button to add a number of files through the system's file open dialog. The same restrictions apply on the file type as explained above.
Remove Files	Removes all files from the dialog. Does however not remove or delete files from your disk.
Create .bak Copies of Input Files	The Propagate Frequent Segments process replaces the selected input files. If you want to keep the old files as well, you should check this box. An old file "input.xlz" is then renamed to "input.xlz.bak" and will not be overwritten with the new file.
Propagate	Clicking this button starts the conversion. You will see a progress dialog, and after the conversion of all documents is done the main dialog stays open.
Cancel	Closes this dialog without further action.

If **Create .bakCopies of Input Files** was checked in the Propagate Frequent Segments dialog, the .bak extension is added to the original files and the processed files are created into the same folder.



After the frequent segments have been propagated to the translated files, the locked frequent segments have translations.



```

<body>
  <h1>(:File with frequent segments)=101%(File with frequent segments:-)</h1>
  <p>(:This segment is not repeated.)=0%(This segment is not repeated:-)</p>
  <p>(:This segment is repeated frequently.)=82%(This segment is repeated frequently:-)</p>
  <p>(:This segment is also repeated frequently.)=85%(This segment is also repeated frequently:-)</p>
  <p>(:This segment is repeated frequently.)=82%(This segment is repeated frequently:-)</p>
  <p>(:This segment is also repeated frequently.)=85%(This segment is also repeated frequently:-)</p>
  <p>(:This segment is also repeated frequently.)=85%(This segment is also repeated frequently:-)</p>
  <p>(:This segment is also repeated frequently.)=85%(This segment is also repeated frequently:-)</p>
  <p>(:This segment is repeated frequently.)=82%(This segment is repeated frequently:-)</p>
  <p>(:This segment is repeated frequently.)=82%(This segment is repeated frequently:-)</p>
  <p> </p>
  <p> </p>
</body>

```

After propagation, propagated repetitions stay locked and have the same percentage match than the translated master repetitions, except for the master repetition translated as 101% match for which the match percentage is set to 100% in the propagated repetitions segments.

If a locked frequent segment was unlocked and translated before the propagation phase, it is overwritten by the **Propagate Frequent Segments** process.

The propagated repetitions have the same **State** attribute value as the translated master repetitions. They also have the **state-qualifier** attribute added and set to leverage-inherited.

Propagation of the frequent segments translation can be done at any time and several times on a batch of XLZ files.

Managing Warnings and Errors

When an error occurs during the **Propagate Frequent Segments** function (due to corrupted XLZ files), an error is displayed in the Propagate Frequent Segments dialog, and an error (.err) file is created with the same name and path as the corresponding source XLZ files.

Note: No global log file is currently created with a summary of all the errors.

Error situations

- A. **No translation is propagated to the repeated segment** which stays **locked**. A warning message is returned to the user. An error (.err) file is created with the same name and path as the corresponding source XLZ file.

Possible causes:

1. The master repetition had not been translated before frequent segments were propagated.
2. The master repetition is not available in the batch of XLZ files submitted for propagation.

- B. **No translation is propagated to the repeated segment** which has been **unlocked**. A warning message is returned to the user. An error (.err) file is created with the same name and path as the corresponding source XLZ file.

Possible causes:

1. The master repetition has been merged to another segment during translation.



Quality Assurance Checks

Prior to the cleanup and back conversion post-processing steps, you should run the available quality assurance checks on your documents. The following checks are available:

Consecutive spaces – The Consecutive Spaces option checks whether there are two consecutive white-space characters in the target segment.

Number values – The Number Values option checks whether numbers that appear in the source segment are also present in the target segment and vice versa.

Tag Checker – The Tag checker feature checks the integrity of the internal tags (or text) in the translated segment. The Tag checker works on Bilingual files, comparing the tags in the source segment against those in the target segment.

Length Check – There are two Length Check types, the **Segment length** check and the **Group length** check.

The **Segment length** check is performed when the `maxwidth` or (and) the `maxbytes` attribute has a value in the `trans-unit` node of the file.

The **Group length** check is performed when the `maxwidth` attribute has a value in the `group` node of the file.

These checks are available only in XLIFF Editor.

Omissions – This check verifies whether a document is fully translated. An omission has occurred if a segment in the source text has no acceptable counterpart in the target text.

End-segment punctuation – This check verifies the consistency of the punctuation at the end of the segment between source and target segments.

Cleanup – This check looks for any errors which might occur during cleanup as a result of broken segment markers, using the functionality from the cleanup function/report, but without actually cleaning the segments. The errors are included in the QA report.

Segment Integrity – Segment integrity checks if any segments exist which include untracked changes. This check is available only in XLIFF Editor.

For instructions on using the QA checks, see either the Translation Workspace Tools or the Word Client Plug-in help system.

Cleanup

The techniques for "cleaning" a bilingual file, that is, removing the source segments and, optionally, storing the translations in the TM, differ between the Translation Workspace Tools/Word Client Plug-in and the XLIFF Editor.

"Cleaning" with the Translation Workspace Tools or the Word Client Plug-in

1. With the Translation Workspace Tools installed, select **Start > All Programs > Translation Workspace > Translation Workspace Tools**. Or in the Word Client Plug-in, select **Current Document** or **Documents**.
2. Select **Clean to Target Language**.
3. Be sure **Store Translation to TM** is checked.
4. Select the Source language and Target language of the files. If these languages do not already exist in the Translation Memory, they are automatically added.
5. Add your files and click **Cleanup**.

This will update the TM with all translation in the files.



Note: For RTF files, this operation will leave your files with only target segments. In addition to (optionally) storing the translations in the TM, it "cleans" the files from bilingual to only target segments. However, if you want to keep a backup copy of the bilingual version, you can check the Create .bak copies of input files option.

"Cleaning" with the XLIFF Editor

If you use the XLIFF Editor, you can use the **Add to TM** tab in **TM > Batch Operations** to store the translations. For the XLIFF Editor, the way you "clean" the file from bilingual to only target segments is with the **File > Save Target As** option.

1. Select **TM > Batch Operations**.
2. In the Select TM dialog, select the TM and click **OK**.
3. Select the source and target languages and click **OK**.
4. In the Batch Operations dialog, select the **Add to TM** tab.
5. Select one or more files and click **Open**.
6. Click **Add to TM**.

Converting Back to the Original Format

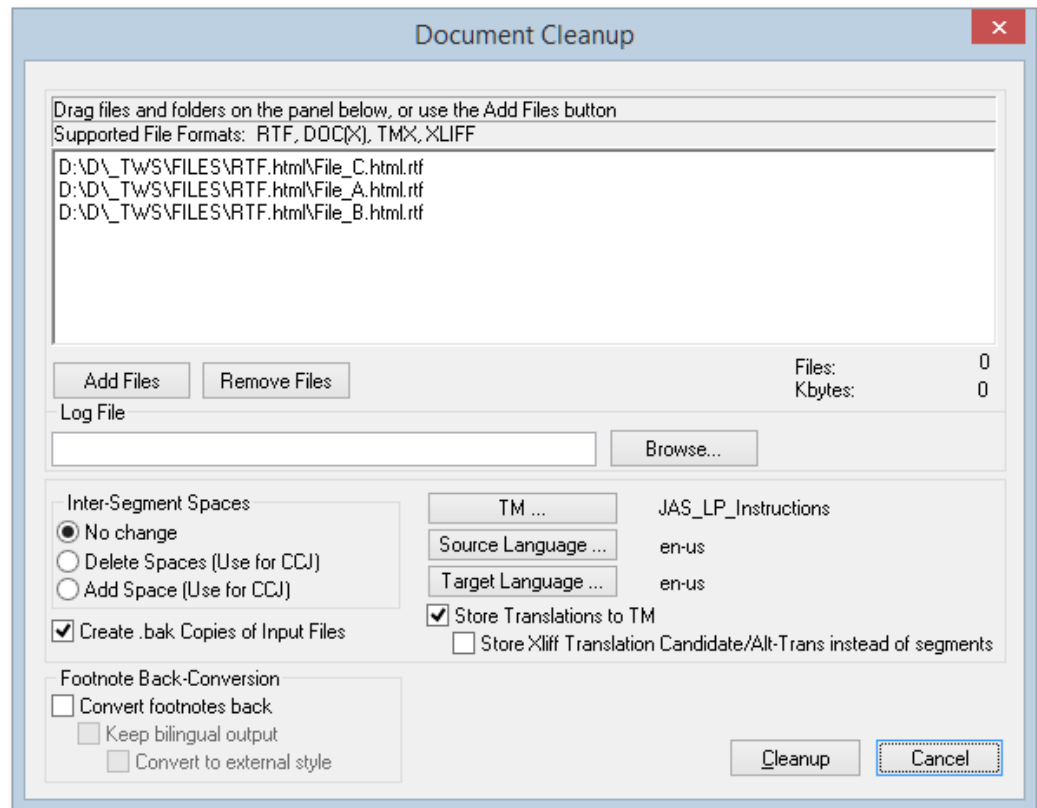
A filter is also required to convert the translation (bilingual) files back into their original format, where they can be integrated back into the product. Target files, when converted back, will be structurally identical to their corresponding source files, but will contain all modifications necessary to call them localized, including:

- Source language content is translated.
- Files have been encoded to the required target.
- Any necessary formatting or re-sizing has been performed.

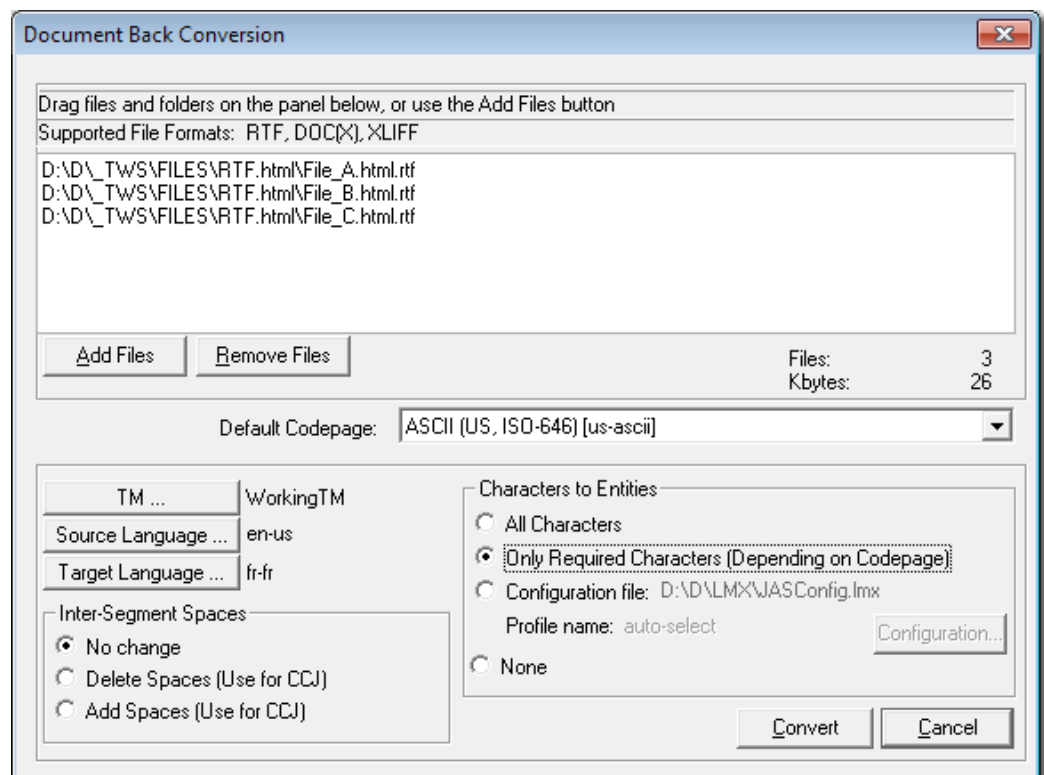
Filtering RTF Files after Translation (Cleanup and Back Conversion)

For RTF, the first filter, to generate a target file from the translation (bilingual) file, is the **Clean to Target Language** (from the **Document Cleanup** dialog) filter.

Note: This filter can also be used for XLIFF files.



After filtering the file from bilingual to only target segments, run the **Back Convert** filter (from the **Document Back Conversion** dialog) to get the file back from XLIFF (or RTF or DOCX) to its original format.





In this example, the Back Convert filter produces an HTML file (*about_translation_workspace.html*) that is in French:

Gestion de l'acquis linguistiques — Translation Workspace™

Premier système de gestion des acquis linguistiques du secteur véritablement conçu pour une architecture Web

Translation Workspace™ est Lionbridge' système innovateur de gestion de patrimoine de langue de système. Translation Workspace stocke les mémoires de traduction et les glossaires dans un répertoire central sécurisé et prend en charge plusieurs milliers d'utilisateurs simultanément. Translation Workspace offre une gestion et un partage efficaces des mémoires de traduction et des glossaires pour plusieurs produits, divisions et groupes fonctionnels d'une société cliente. Les clients tirent pleinement parti de cette plate-forme technologique hébergée en choisissant Lionbridge comme fournisseur de services de globalisation.

Une approche centralisée de la gestion des mémoires de traduction.

Contrairement aux autres systèmes à mémoires de traduction, Translation Workspace place les mémoires de traduction et les glossaires (collectivement nommés acquis linguistiques) dans un répertoire centralisé et les stocke de façon sécurisée en un lieu unique. Les ressources provenant de différents projets et de différents groupes de travail peuvent être facilement ajoutées pour renforcer l'efficacité. Les traducteurs accèdent au serveur de Translation Workspace par Internet, à l'aide de l'application Translation Workspace cliente gratuite. Les mises à jour sont renvoyées au répertoire central en temps réel. Grâce à son architecture en ligne avancée, les importantes ressources des clients ne sont plus stockées sur des dizaines d'ordinateurs à travers le monde. Au contraire, Translation Workspace permet une gestion sécurisée et centralisée des mémoires de traduction et des glossaires, pour offrir une efficacité et une cohérence accrues au cours du processus de traduction, et une meilleure protection des données sensibles.

L'approche centralisée de la gestion des MT de Translation Workspace offre de nombreux avantages.

This conversion is the last step in the document translation workflow.

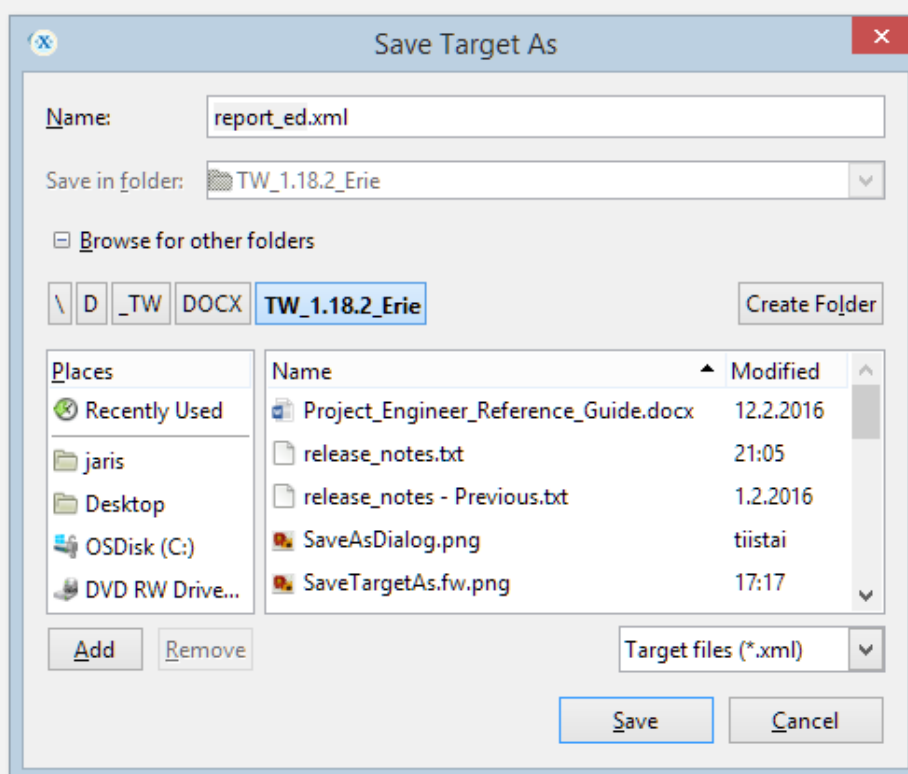
For HTML/XML/SGML files you can select the encoding of the converted file, as well as the converted extended characters to entities to ensure a good display in your browser.

These options do not apply when converting other file formats (MIF, DOCX, PPTX, IDML, INX).

For Text files you must select the encoding of the converted file. Choose the **None** option.

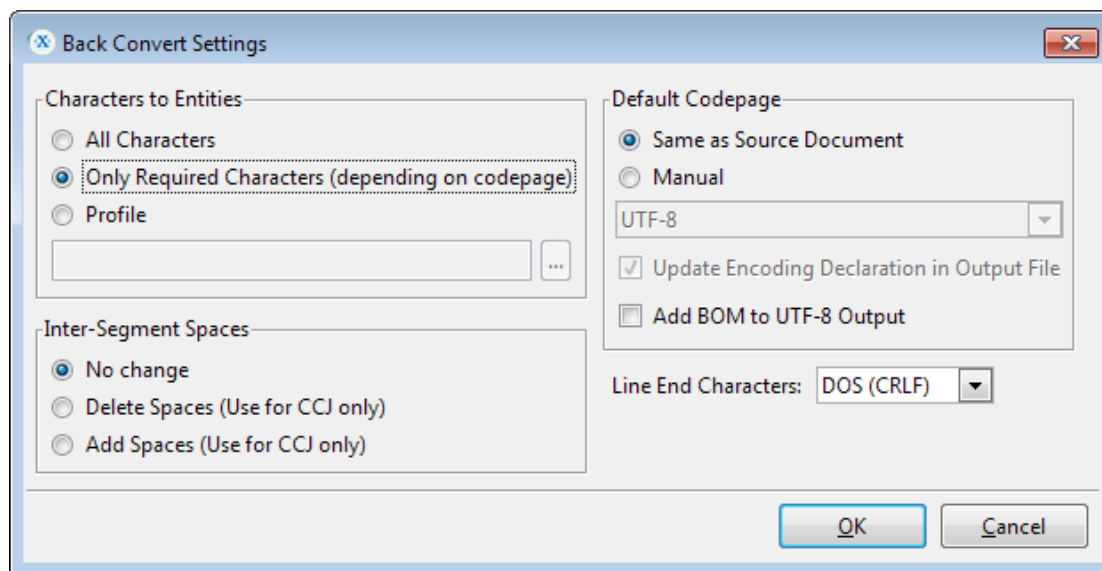
Filtering XLIFF Files after Translation

For XLIFF, the first filter, to generate a target file from the translation (bilingual) file, is the **File > Save Target As** option, which displays the **Save Target As** dialog.



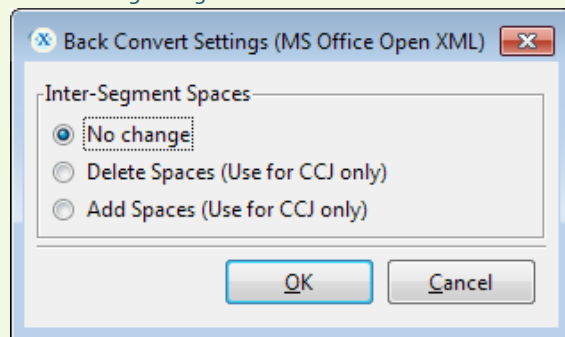
You can use the **Save Target As** dialog for saving the target file into a different folder than where the current XLZ file is in and for renaming the target file.

When you select **Save** in the **Save Target As** dialog, the **Back Convert Settings** dialog is displayed.



See the “File Menu” topic in the Translation Workspace XLIFF Editor help for a description of these options.

Note: When you run back convert on a Microsoft Office XML file (DOCX, PPTX, or XLSX), you get the following dialog:



When you run **Back Convert**, it creates a new translated file in the original format. For example, if the XLIFF file you run it on is *about_translation_workspace.html.xlf*, it will create a file named *about_translation_workspace.html*.

Gestion de l'acquis linguistiques — Translation Workspace™

Premier système de gestion des acquis linguistiques du secteur véritablement conçu pour une architecture Web

Translation Workspace™ est Lionbridge® système innovateur de gestion de patrimoine de langue de système. Translation Workspace stocke les mémoires de traduction et les glossaires dans un répertoire central sécurisé et prend en charge plusieurs milliers d'utilisateurs simultanément. Translation Workspace offre une gestion et un partage efficaces des mémoires de traduction et des glossaires pour plusieurs produits, divisions et groupes fonctionnels d'une société cliente. Les clients tirent pleinement parti de cette plate-forme technologique hébergée en choisissant Lionbridge comme fournisseur de services de globalisation.

Une approche centralisée de la gestion des mémoires de traduction.

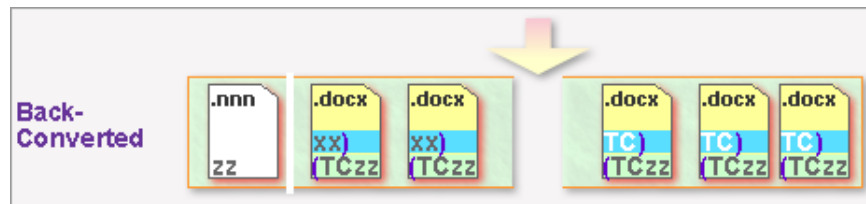
Contrairement aux autres systèmes à mémoires de traduction, Translation Workspace place les mémoires de traduction et les glossaires (collectivement nommés acquis linguistiques) dans un répertoire centralisé et les stocke de façon sécurisée en un lieu unique. Les ressources provenant de différents projets et de différents groupes de travail peuvent être facilement ajoutées pour renforcer l'efficacité. Les traducteurs accèdent au serveur de Translation Workspace par Internet, à l'aide de l'application Translation Workspace cliente gratuite. Les mises à jour sont renvoyées au répertoire central en temps réel. Grâce à son architecture en ligne avancée, les importantes ressources des clients ne sont plus stockées sur des dizaines d'ordinateurs à travers le monde. Au contraire, Translation Workspace permet une gestion sécurisée et centralisée des mémoires de traduction et des glossaires, pour offrir une efficacité et une cohérence accrues au cours du processus de traduction, et une meilleure protection des données sensibles.

L'approche centralisée de la gestion des MTL de Translation Workspace offre de nombreux avantages :

You can also use Back Convert from the **TM > Batch Operations** dialog to get the file back from XLIFF to its original format.



Back Converting Files with Tracked Changes



When generating a translated document using the Back Convert function (TW Tools, Word Client) or **File > Save Target** in the Xliff Editor from a XLZ file containing tracked changes:

If the original source document is a Microsoft Word 2007-2010 document (.docx), then tracked changes in target segments are preserved and appear in the final DOCX file.

For all other original file types (PPTX, XLSX, IDML, MIF...), (.nnn), target segments tracked changes are automatically accepted in the translated XLZ file and then the final document is generated.

If a segment does not contain a target (translation), the source segment with all the tracked changes is retained in the back converted file.

Note: You can back convert XLZ files with tracked changes and XLZ files without tracked changes to DOCX in the same run.

Back conversion of XLZ files with tracked changes to DOCX format retains the tracked changes existing in the XLZ files in the resulting DOCX files when **Convert Tracked Changes (only for XLZ format)** is enabled in the DOCX Filter.

To generate translated DOCX files without tracked changes from XLZ files which include tracked changes, *disable* the **Convert Tracked Changes (only for XLZ format)** option in the DOCX Filter.

In XLIFF Editor you can open the DOCX Filter configuration by selecting **Tools > Open Translation Workspace Filter Configuration ... > Configure Filters > DOCX**.

Known Limitations:

Back conversion attempts to add track changes to fields such as comments and alternate text that are not supported by Word. In case of comments, the changes are joined together and in case of alternate text, the translation is not displayed.

Track changes to field codes are accepted during back conversion.

Back conversion changes deletions to insertions when the content is inside a deleted text box.

Track changes in index entries are not supported during forward and back conversion. Changes are accepted.

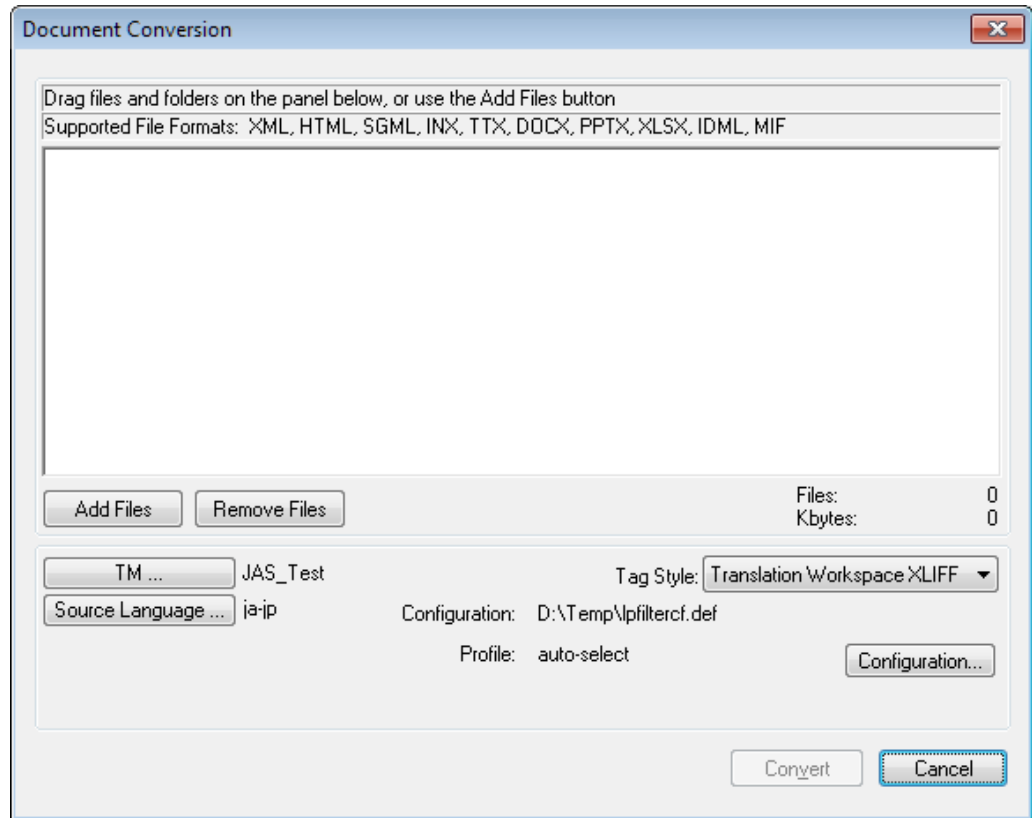
Track changes on paragraph marks are not accepted during back conversion, when the option **Convert Track Changes** is **OFF**.

Note: For XLZ files which were originally converted from other file formats than DOCX, by using other filters, all the tracked changes in the files are accepted and the tracking is not enabled in the files created by the back conversion.



Markup Language File Formats (XML, HTML, SGML, XHTML, ASP, JSP)

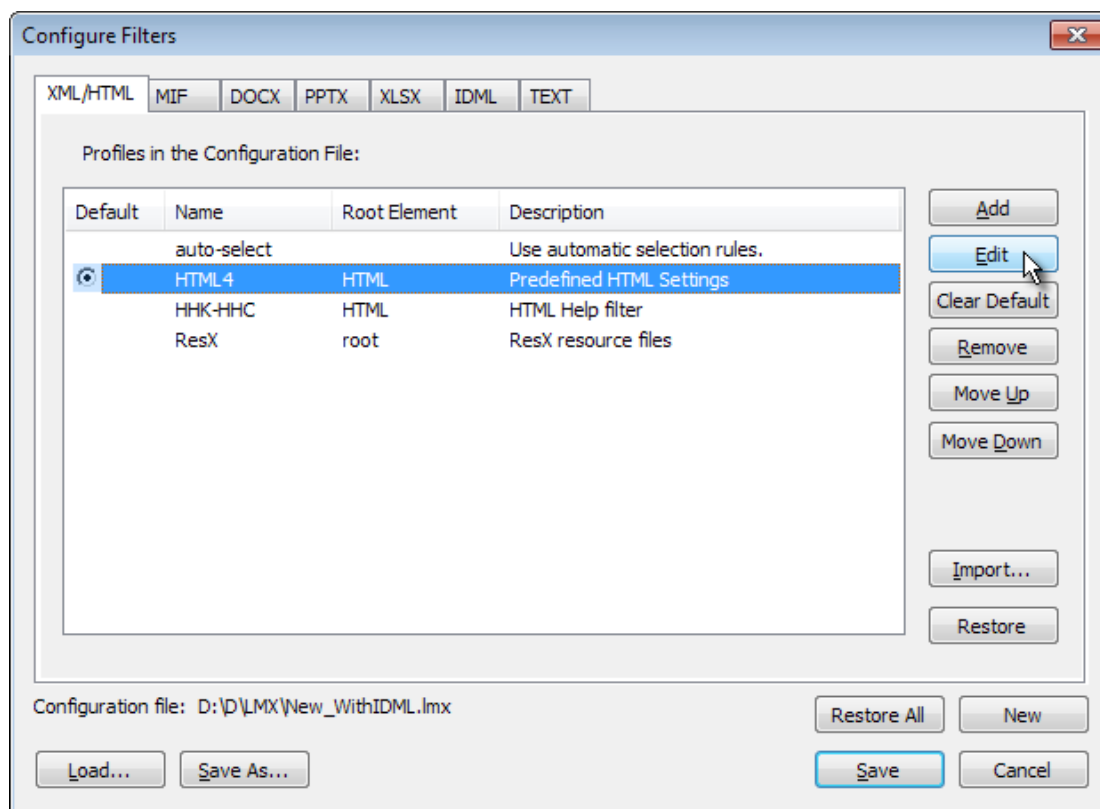
To get files in markup language format (XML, HTML, SGML, XHTML, ASP, JSP) into XLIFF, RTF, or Trados, use the **Document Conversion** dialog.



To configure the markup filters, clicking the Configuration button brings you to the **Configure Filters** dialog, which contains profiles for XML and HTML configuration (as well as for MIF, DOCX, PPTX, and XLSX formats):

A Profile is a set of parsing rules, defining what is translatable or non-translatable in the source document.

In this dialog each conversion filter has a separate tab, where you can configure the conversion options.

**Note:** Supported Document Encodings

The Markup Language conversion function supports all document encodings known today for HTML and XML documents. It detects the document character set from the respective meta tag and auto-detects the used Unicode format (if used).

All these formats are converted into correct, completely displayable, RTF or XLIFF. Depending on the settings, content entities are displayed as the characters they represent and not as XML entities (&#punc;).

For information about the settings for the [MIF](#), [DOCX](#), [PPTX](#), [XLSX](#), [IDML](#), [INX](#), or [TEXT](#) filters, see the section for the particular filter.

The **Add**, **Edit**, **Set/Clear Default**, **Remove**, **Move Up**, and **Move Down** buttons apply only to the XML/HTML and TEXT filter profiles.

Note: Translation Workspace provides a Default .lmx file with some predefined settings for each filter. For example for HTML/XML, pre-defined profiles are provided for HTML4, HHK-HHC, and Resx.

Option	Description
Add	To add a profile, click Add . A new line (labeled "new1" appears in the profiles list. With the new profile selected, click Edit to go to the Markup Language Assistant dialog, where you can configure the new profile. When you have configured the profile there, click OK to return to the Configure Filters dialog. Click Save to save your new profile, which will show up in the Document Conversion dialog as the current profile.
Edit	Click Edit to change the configuration of the selected profile. Edit takes you to the Markup Language Assistant dialog, where you perform the configuration.



Set/Clear Default	Use this button to either set the selected profile as the default, or to clear it from being the default.
Remove	Removes the profile from the list.
Move Up	Moves the profile up in the list.
Move Down	Moves the profile down in the list.

You can update the options of one single filter (restore, import) or for all filters in one go, by using the **Import...** and **Restore** buttons.

Profile Selection and Auto-select

A Profile is a set of parsing rules, defining what is translatable or non-translatable in the source document. It also sets elements to be internal or external. The Profile uses the Root element in your source document(s) as its identifier.

When running a conversion later, this Root element is then used as a shared parameter to map a document (type) and its Root Element to its corresponding Profile (and related parsing rules).

A Configuration file can include multiple Profiles.

The user has two main choices to decide which profile to apply during the conversion process:

- ❑ Select a profile from the list: using the selected profile is enforced.
- ❑ Select the auto-select option from the list: rules (see below) are used to automatically select a profile during the conversion process.

If the auto-select option is checked in the profile list, the ML Conversion tool allows parsing XML/HTML/SGML files with different document types (different DTD, different root element) together. For each file (type) to be parsed, the matching profile will be searched from the list of available profiles.

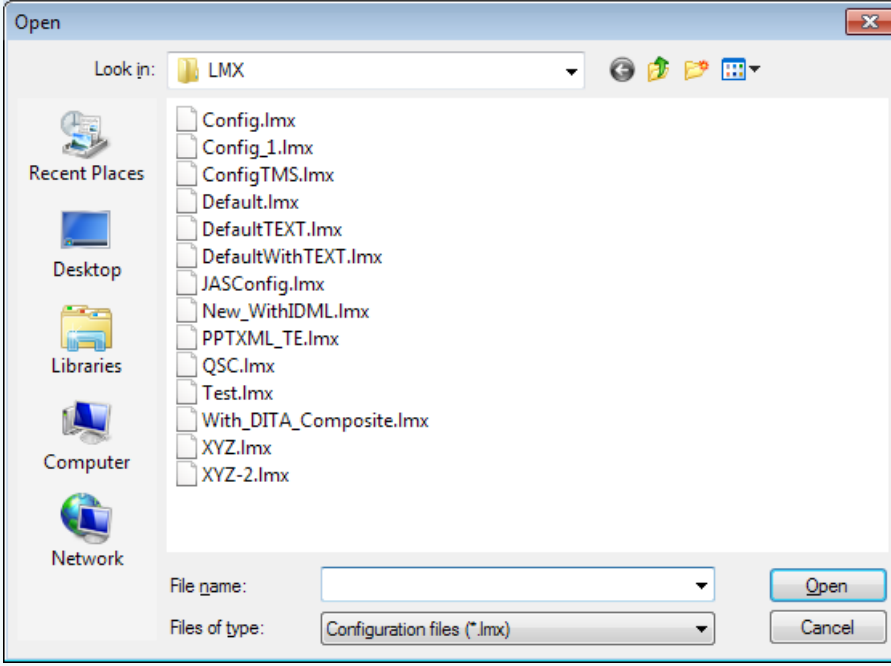
The right profile will be applied, following these auto-select rules:

1. If there is more than one matching profile for a given document type, the first matching profile found in the list is used. This profile is used to process all files that belong to the relevant document type (root element).
2. If there is no matching profile available for a given document type, Translation Workspace checks whether there is one profile defined as the "default" in the list. If so, the default profile is used to process all files that belong to the relevant document type.
3. If there is no matching profile available for a given document type, and no default profile was set, the file will be parsed with the default settings (as if no settings file was provided)



Actions Common to All File Types

The common actions for all filters are listed in the table below.

Option	Description
Configuration file:	Displays the full path of the current configuration file.
Load...	You can change the Configuration file (.lmx), by clicking Load. Note: The configuration file (.lmx) contains the configuration settings for all filters listed in the Configure Filters dialog. Translation Workspace provides a default configuration file (Default.lmx). This displays the Open dialog, where you can browse and select an existing Configuration file.  You can also drag and drop a configuration file on the Configuration window to load it. Loading a configuration file also replaces the current configuration file (whose path is displayed at the bottom of the Configure Filters dialog).
Save As...	Click Save As to save all filter settings into a new configuration file. This opens the system Save As dialog with the filter settings File type set to "Configuration file (*.lmx)".
Restore All	Click Restore All to restore default settings for all filters.
Save	Stores the new settings to the current configuration file (whose name is displayed at the bottom of the Configure Filters dialog) and closes the dialog.
New	Click New to create a new Configuration file. All loaded .lmx files will be closed. If unsaved changes already exist, the user will be prompted to save or cancel the changes. Then all settings of all filters are set to default value.
Cancel	Closes this dialog box without further action.

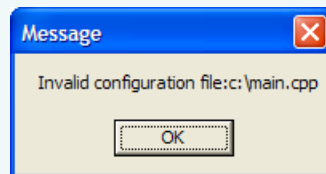
**Import...**

Click the **Import...** button to import the settings related to the current filter from an existing configuration file.

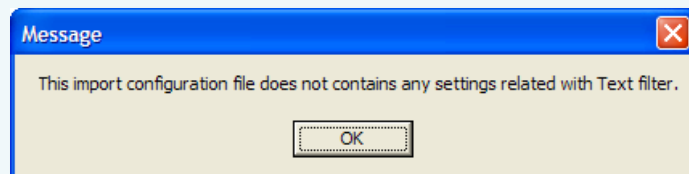
This displays the **Open** dialog, where you can browse and select an existing Configuration file.

By clicking **Open** you can retrieve and import the settings for the currently selected filter from the selected configuration file.

If the file to be imported is not a valid configuration file, an error message is displayed:



If the configuration file is valid, but it does not include any settings for the current filter, the existing profiles remain unchanged, and a message is displayed:



If the configuration file is valid, and it includes settings for the current filter, all the profiles in the current filter are updated with the profile settings in the imported settings file.

Note: The current configuration file (whose path is displayed at the bottom of the **Configure Filters** dialog) is not updated until you click **Save** to save the new settings.

Restore

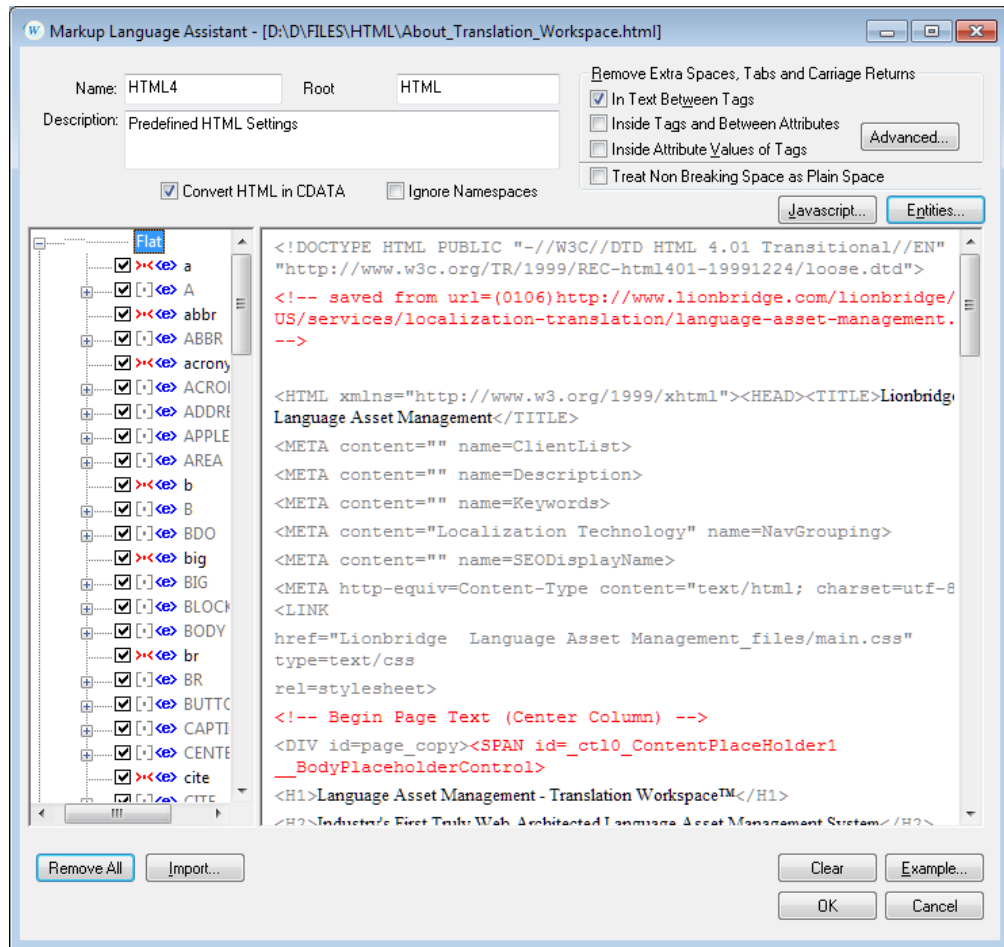
Click **Restore** to restore default settings for the current filter.

Note: Translation Workspace provides a Default .lmx file with some predefined settings for each filter. For example for HTML/XML, pre-defined profiles are provided for HTML₄, HHK-HHC, and Resx.

Clicking the **Edit** button brings you to the **Markup Language Assistant** dialog, where you can configure parsing rules for the filter.



Markup Language Assistant Dialog



This dialog contains the following layout:

- In the top left, you'll see the name of the profile, a description, and the root element.
- The top right shows you some text handling options, and buttons to configure JavaScript and Entities handling.
- The tree view on the left shows all the elements and attributes for the profile,
- The preview pane shows what a document would look like if it were converted based on the current settings. You can drag and drop a file into this area to see how the settings you choose affect it.

You can use the checkboxes in the tree view to mark that element or attribute as translatable or non-translatable text.

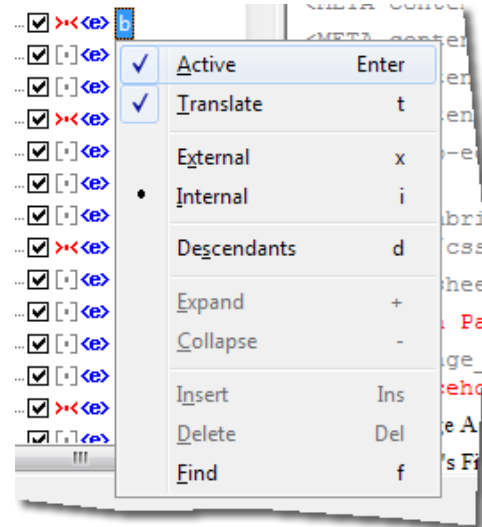
The tree view contains three types of settings that determine whether an element or attribute is translatable or not.

- Flat settings are Boolean, in that they indicate that an element or attribute is always translatable, or always non-translatable,
- Hierarchical settings define translatable or not based on the file location of the element or attribute matching the hierarchy.
- Advanced settings can be used to define more complex parsing rules using XPath expressions.



Here is the way you work in the tree view:

- You can activate or de-activate a rule by right-clicking and selecting **Active**. Activated rules show as black font, and de-activated rules show as gray font.
- Elements are indicated with an “e” icon, and attributes are indicated with an “x=” icon. These elements and attributes can be defined as translatable or non-translatable by enabling or disabling the checkbox next to it.
- You can also set an element as external or internal.



The **Markup Language Assistant** provides an intuitive and interactive way for you to identify localizable content within a tagged file (for example, XML, HTML, and SGML). The “parsing rules” are stored as a “Profile” in a Configuration file. This file is saved in XML format, which advanced users can also create or manually update in a text or XML editor.

The Markup Language Assistant allows advanced users to take benefit from the power of XPath expressions. The ML Assistant also allows you to define XPath expressions with conditional parsing rules based on attribute values for HTML and XML files.

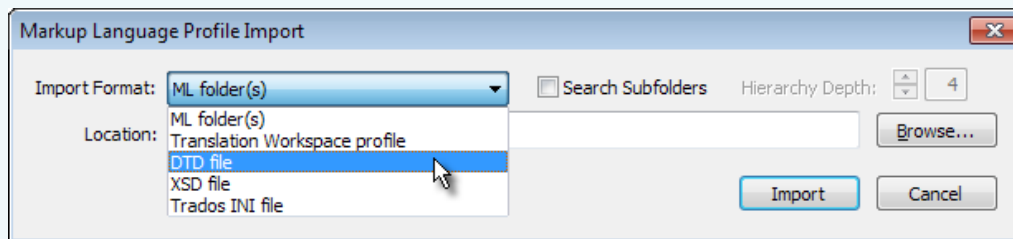
You can use the **Load...** button on the **Configure Filters** dialog or the **Import...** button on the XML/HTML tab of the **Configure Filters** dialog to load a Configuration file. You can also drag-and-drop a Configuration file in the Tree View pane of the Markup Language Assistant window to load it.

The ML Assistant dialog has the following options:

Option	Description
Name	The name of the Profile. You can change the name by typing in the text field. This Name will also display in the Configuration window.
Description	The description of the profile. You can add a descriptive text in the text field. This Description will also display in the Configuration window.
Root	The root element of a ML file is the element, which includes all other content elements of the file. If you do not identify the root element correctly, the parser may not work as it should. The root element in HTML files is usually HTML, but in XML, for example, the root element depends on the DTD to which the localizable XML files comply. If the root element is different in two XML files, usually the other element tags are also different, because of the settings in the DTD. In auto-select mode, this value will be used to do the mapping between files to convert and Profiles available. Note: This field is case sensitive.
Convert HTML in CDATA	Markup HTML tags within CDATA tags to make them internal or external.
Clear (tree views)	Clears all profile settings.

**Import**

The Import button allows importing settings from external files. Click the **Import** button to open the **Profile Import** window, where you can select the appropriate import format.



After selecting the **Import format** and **Location** of your import file, click the **Import** button. This will import the settings from your import file into the current profile.

Notes: You can use drag and drop in this Import window. The Import format will then be automatically selected.
You can also use drag and drop directly in the tree view. The Import window will then be opened and completed automatically.

The following Import formats are supported:

Import Format	Description
ML folder(s)	You can provide a folder path containing ML files (SGML, HTML, XML). The files contained in the folder (and sub-folders if the Search subfolders option is enabled) will be analyzed and settings will be imported.
Translation Workspace profile	To import a Translation Workspace Profile from an existing Configuration file. If this is an old single Profile Configuration file, it will be imported directly. If this is a Multi-profile Configuration file, the list of available profiles will be displayed and the user can select from the list.
DTD file	To import settings from a Document Type Definition (DTD) file. For this specific import, you can select the analyze depth in the DTD file. The default setting is 4 levels down in the hierarchy.
XSD file	To import settings from a Schema file (XSD).
Trados INI file (TagEditor ini file support)	It is possible to load the settings from a TagEditor INI file and convert these to an ML Assistant Profile. It is strongly recommended to double check the settings and the parsing results when importing from an INI file. Since TagEditor is using a different parser engine, the parsing result may be different in some cases. A known limitation is that Translation Workspace takes into account namespaces during parsing, while Trados does not.

Clear

There are two **Clear** buttons in this window. The left **Clear** button will remove all settings from the Tree View pane. The right **Clear** button will clear the Preview pane.

Example

Allows loading a sample file and previewing the parsing result.
You can also use drag and drop to load a sample file.

**Ignore namespaces**

This option determines whether or not a namespace declaration should be taken into account during parsing.

The option is only relevant for XML files parsing, as for HTML we will always ignore namespaces.

JavaScript (Parsing Rules)

If you're working with HTML files containing JavaScript segments, you'll also want to define whether any of the scripts should be processed for translatable content.

Clicking the **JavaScript** button in the **Markup Language Assistant** window will open this window, where you can define your parsing rules for JavaScript inside the documents you want to convert:

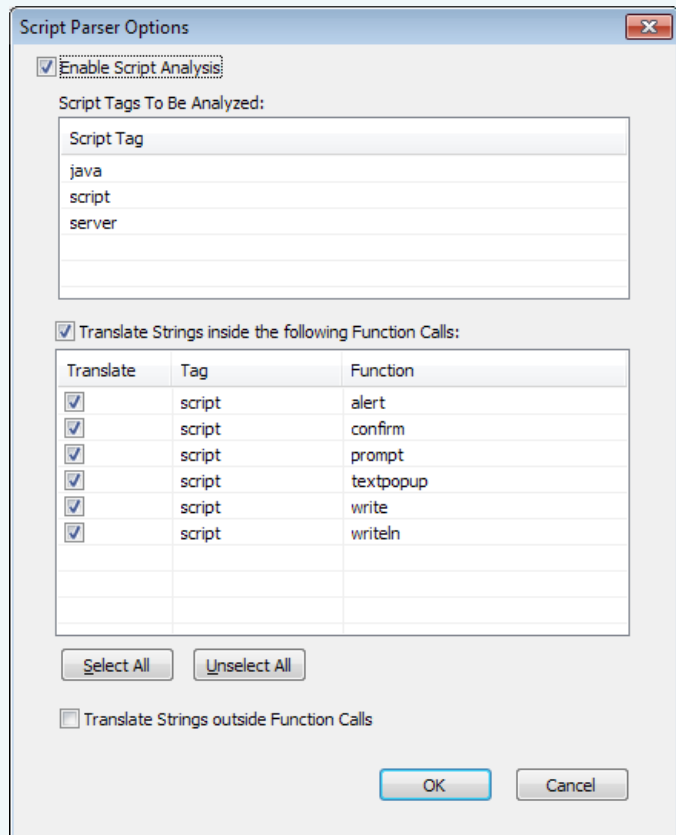
The JavaScript parser allows you to define:

- Whether or not to analyze JavaScript
- Which tags to analyze
- Which function calls inside the selected tags to analyze

Script analysis analyzes the listed JavaScript tags for the possible localizable function calls. The analysis is done during parsing. If you disable this option, no JavaScript functions are parsed as translatable, whether they are listed as such or not.

Turn on the option to **Enable Script Analysis**, then enter in the script tags that need to be analyzed. Use the **Insert** key on your keyboard to do this.

You can choose to specify the function calls that contain translatable strings (string inside double quotes), or indicate that there are translatable strings outside of the function calls. Use the **Insert** key to add new function calls.





Entities

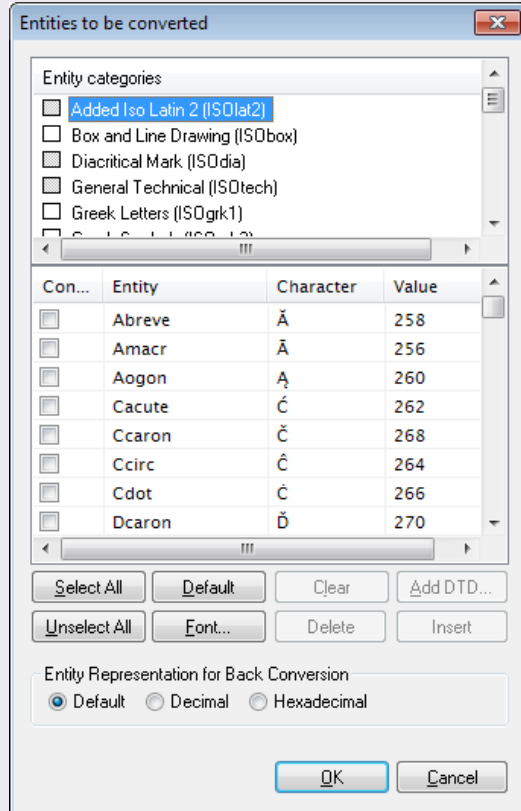
When working with HTML/XML formats, you may also need to convert entities into special characters. For example, the entity expressed by "ampersand" then the letters "l" and "t" indicate the special character "less than". By default, all such XML entities and ISO Latin 1 entities are converted into their corresponding special characters. You can control which entities you want converted.

Clicking the [Entities](#) button launches the [Entities to be converted](#) dialog.

This allows the user to define which entities should be converted to their corresponding characters in the output RTF/XLIFF.

Unconverted entities are put in Internal style.

You can also add entities that are not in the list via the User-defined category at the bottom of the Entity categories list. Use the [Insert](#) key on your keyboard to add a new entity for conversion.



**Remove extra spaces, tabs and carriage returns**

This option removes extra spaces, tabs or carriage returns from the translatable text. This ensures, for example, that a sentence will not be split into two segments.

Remove Extra Spaces, Tabs and Carriage Returns

☒ In Text Between Tags

☐ Inside Tags and Between Attributes

☐ Inside Attribute Values of Tags

☐ Treat Non Breaking Space as Plain Space

Advanced...

These are the settings for the **Remove Extra** spaces option

Note: We will abbreviate this as ESCR = Extra spaces, tabs, carriage returns:

Option**Description**

In text between tags

ESCR are removed in translatable text between tags. (To treat the non breaking spaces as normal spaces enable also the option **Treat Non Breaking Space as Plain Space**). For example:

```
<p>H E L L O  
W O R L D</p>
```

results in:

```
<p>H E L L O W O R L D</p>
```

Note: Single-spaces are not removed, but in the places where there are three consecutive spaces, two are removed. The carriage return is also removed—replaced by a space.

Inside tags and between attributes

ESCR are *not* removed from text between tags when the range of ESCR is next to a tag that will have External style. In this case, the range of ESCR, is set to External style.

ESCR are removed in tags and when occurring between attributes in tags. (To treat the non breaking spaces as normal spaces enable also the option **Treat Non Breaking Space as Plain Space**).

```
<p><b class="Indented Note" type="text">This</b> is text.</p>
```

results in:

```
<p><b class="Indented Note" type="text">This</b> is text.<p>
```

Note that the extra spaces are removed between the two attributes (class="Indented Note" and type="text").

Inside attribute values of tags

ESCR are removed inside string values of tag attributes. (To treat the non breaking spaces as normal spaces enable also the option **Treat Non Breaking Space as Plain Space**). For example:

```
<p><b class="Indented Note" type="text">This</b> is text.</p>
```

results in:

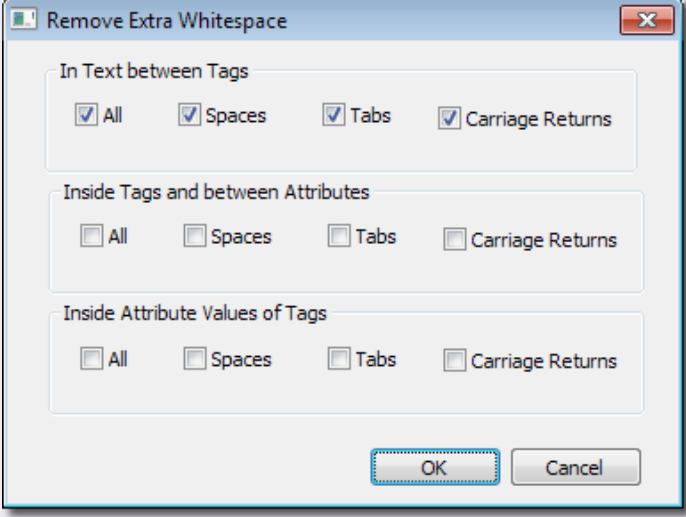
```
<p><b class="Indented Note" type="text">This</b> is text.<p>
```

Note: Note that the extra spaces have been removed in the attribute string "Indented Note".

Treat Non Breaking Space as Plain Space

When this option is enabled combined with the previous options also the non breaking space characters are handled the same way as the plain space characters in the previous examples. However, non breaking spaces outside elements are not treated as plain spaces.



Advanced	If you click the Advanced... button, to give users more flexibility for the parsing, the Remove extra spaces, tabs, and carriage returns function has been split out into three individual choice sets.	
Tree View pane	The different tree view controls present a list of all the element names and element attribute names used within the ML file(s) (or DTD/XSD/INI file). Element names can be used an infinite number of times in an ML file; the tree view control distills these occurrences and only displays the different element name once in the list for a specific context. Any attributes used with an element in the ML file(s) are displayed as child items to that element in the tree view. Each element and element attribute of the tree view can be selected. Actions are associated to the selection. These actions allow you to set different properties of the elements and element attributes, such as Translatable/Not translatable and Internal/External, thus defining the parsing rules. Right-clicking the selected item will open a contextual menu with the list of possible actions and the associated accelerator keys. Two checkboxes are associated to each item of the tree view: one will define the element or attribute as translatable, the second will define it as External/Internal.	
Sample Preview pane	The user can load one of the analyzed files in the sample preview pane by clicking on the Example... button. Any action done on the Tree view pane is reflected in the preview pane that will present the resulting parsed file (Translation Workspace RTF like). The preview will grey out untranslatable elements/attributes and also differentiate between Internal and External elements (using red/grey) color. Any other settings, such as entity conversion, removing extra spaces or carriage returns will also be reflected in the preview pane.	

Defining Parsing Rules

You can control how Translation Workspace handles specific tags. You might want the content of certain tags to *not* be presented for translation. For example, if your content included code that was tagged as `<code>`, you might want this code to be protected, so it couldn't be translated. Or you might want certain tags to be relocated if need be, but not modified. The **Markup Language Assistant** dialog lets you accomplish both of these with the following parsing options:

Translate lets you control whether the content of a tag will be presented for translation.



External/Internal lets you control whether tags themselves can be moved.

Note: If a tag is **External**, it is presented as grayed out in Word, and you cannot do anything to it.
If it is **Internal**, it displays as red, and you can only move it to another valid location.

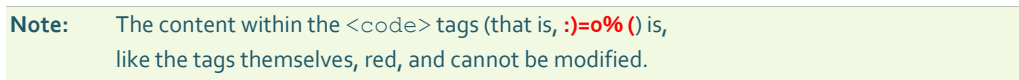
To set up parsing rules, do the following:

1. Drag or add a list of ML file(s), a DTD/XSD, or INI file into the **Document Conversion** dialog.
2. Click the **Configuration...** button.
3. In the **Configure Filters** dialog, select a Profile for the file type for which you are setting parsing rules.
4. Click **Edit**. Translation Workspace analyzes the files, extracting the tags and placing them in a list in the **Tree View** pane.
5. In the **Tree View** pane, you can make a tag untranslatable either by unchecking the checkbox next to it, or by right-clicking it and clicking **Translate**. You will see the checkbox is now unchecked. You can control whether the tag is **Internal** or **External** by clicking either **Internal** or **External**, which toggle between each other.

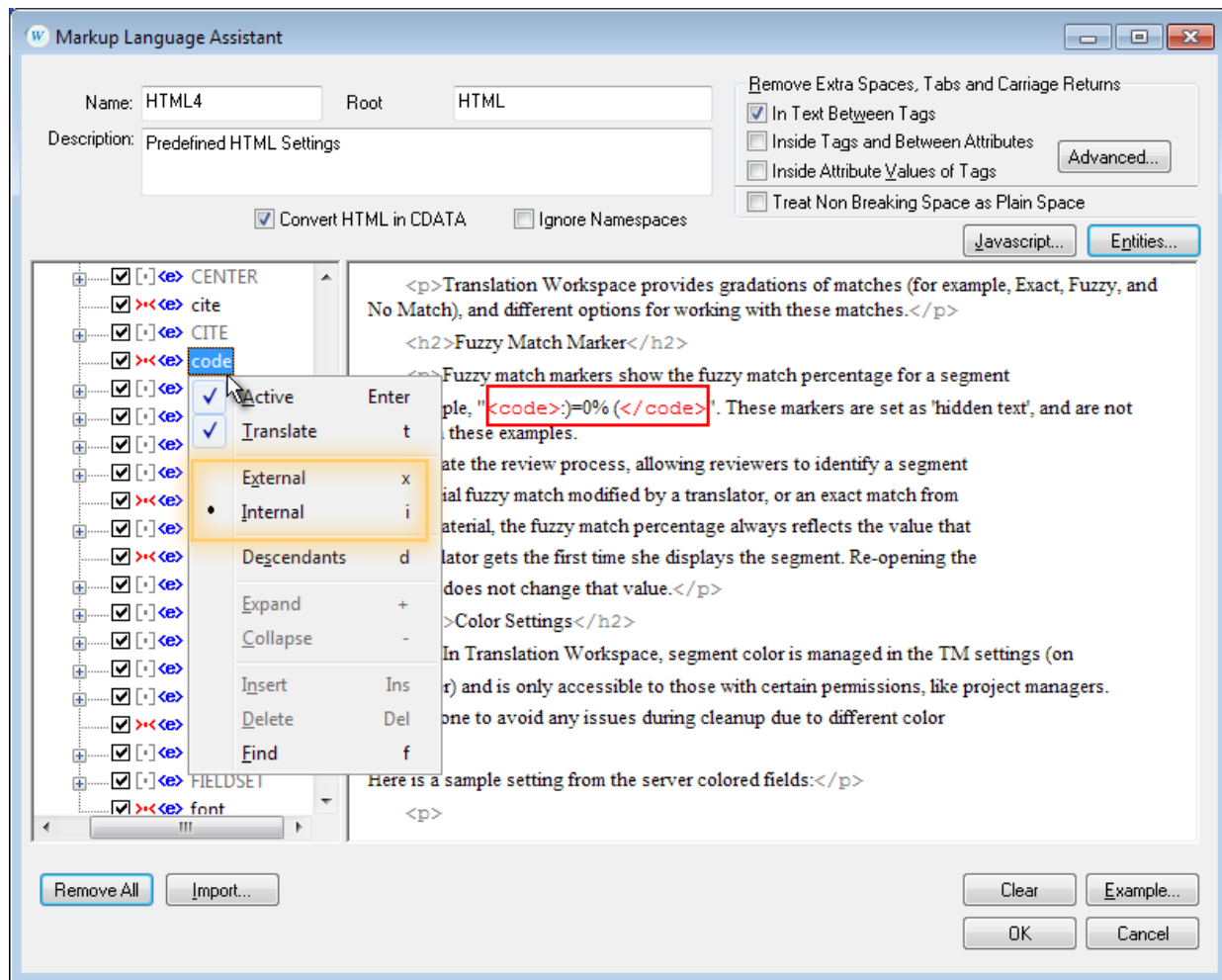
One of the analyzed files (typically, one of the more complex ones) can then be loaded in the preview window (by clicking the **Example** button) to represent the result of the parsing rules set in the tree views.

Warning: The Translation Workspace parser is case insensitive for tag name, attribute name, and attribute values in HTML files, and case sensitive for XML files.

The following screen capture shows the Markup Language Assistant previewing a file mapped to a Profile that has the `<code>` tag marked to not present for translation.



The next screen capture shows the same passage when the `<code>` tag has been marked for translation (and kept as **Internal**).



Note: Here the `<code>` tag itself is red, indicating that the translator can relocate it, but not modify it; and the text the markup tags (that is, `=0%()`) is black, indicating that the translator can modify it.

Using the Flat Tree View

Once the ML files have been loaded and analyzed, you will need to identify the content that requires translation.

Defining an element/attribute as translatable in the Flat tree view implies that all occurrences of the element/attribute name in the ML file require translation wherever the element is located.

It is good practice to start defining parsing rules using this view, as it allows defining very generic rules that apply to one element/attribute name in the whole document. These rules can then be further refined in the **Hierarchical** and **Advanced** tree views.

In the following sample, we set all occurrences of the element `<num>` as translatable and internal:

Note: All elements are set to Translatable/External and all element attributes are set to Not Translatable by default.



Sample and Legend

To help understand setting parsing rules in the **Flat Tree View**, we will use the following sample and give a legend of what the various icons are.

Sample

Consider the following sample:

```
<para text_status="ToTrans">Text to translate, sample <num>three</num></para>
```

Legend

The following table explains what the icons in the **Flat Tree View** are:

Flat Tree View	Icon	Legend
		ML element
		ML element attribute
		Active element (a rule is associated with the element)
		Inactive (grayed out) element (no associated rule)
		The <num> element is active and set to Translatable/Internal.

Possible actions

The following table shows the actions for the various parsing settings:

Contextual Menu	Icon/Checkbox	Action
Active Enter	(None)	Use this menu to activate a rule associated to an item. Activating a rule will generate the corresponding XPath expression in the Advanced view and apply the corresponding setting. Note: XPath will only work completely in the flat tree and advanced tree view.
✓ Translate t	☐ ☒	Use this menu or checkbox to set an element/attribute as Translatable/Not translatable.
• External x Internal i	☐ ☒	Use this menu or checkbox to set an element as External/Internal.
✓ Descendants d	(None)	Use this menu to include/exclude descendants of the selected element in the parsing rule.
Expand + Collapse -	 	Use this menu or icons to Expand/Collapse a tree branch.
Insert Ins Delete Del Find f	(None)	The Insert and Delete menus are disabled in flat view. The Find menu allows you to highlight the selected item in the preview pane. You can also double-click.



Using the Hierarchical Tree View

In the Hierarchical tree view, defining an element/attribute as translatable implies that all occurrences of the element/attribute name in the XML file having the same "path" (matching with the hierarchy) require translation.

This view allows defining more precise parsing rules as you can define different rules for a same element/attribute name depending in which branch of the tree it appears.

In the following example, the `<para>` element should be translated when it appears under the `<section>` element, but not when it appears under the `<title>` element.

Note: All elements are set to Translatable/External and all element attributes are set to Not Translatable by default.

Sample and Legend

To help understand the setting parsing rules in the Hierarchical Tree View we will use the following sample and give a legend of what the various icons are.

Sample

Consider the following sample:

```
<title>
<tr_unit text_status="NotToTrans">
<para text_status="ToTrans">Text to translate, sample one</para>
<para text_status="NotToTrans">Text not to translate, sample one</para>
</tr_unit>
</title>
<section>
<tr_unit text_status="ToTrans">
<para text_status="ToTrans">Text to translate, sample two</para>
<para text_status="NotToTrans">Text not to translate, sample two</para>
</tr_unit>
</section>
```

Legend

The following table explains what the icons in the Hierarchical Tree View are:

Hierarchical View	Icon	Legend
	<code><e></code>	ML element
	<code>x="</code>	ML element attribute
	<code><e> para</code>	Active element (a rule is associated with the element)
	<code><e> para</code>	Inactive (grayed out) element (no associated rule).
	☒ <code>[*]</code> <code><e> para</code>	The <code><para></code> element is not active, default rule apply (translatable/external).



Flat

Hierarchical

root

date

manual_ref

section

text_status

tr_unit

para

text_status

title

tr_unit

para

text_status

Advanced

☐ [.] <e> para

The <para> element is active and set to Not translatable/External.

Possible actions

The following table shows the actions for the various parsing settings:

Contextual Menu	Icon/Checkbox	Action
Active Enter	(None)	Use this menu to activate a rule associated to an item. Activating a rule will generate the corresponding XPath expression in the Advanced view and apply the corresponding setting.
✓ Translate t	☐ ☒	Use this menu or checkbox to set an element/attribute as Translatable/Not translatable.
• External x Internal i	[.] [.]	Use this menu or checkbox to set an element as External/Internal.
✓ Descendants d	(None)	Use this menu to include/exclude descendants of the selected element in the parsing rule.
Expand + Collapse -	+ -	Use this menu or icons to Expand/Collapse a tree branch.
Insert Ins Delete Del Find f	(None)	The Insert and Delete menus are disabled in flat view. The Find menu allows you to highlight the selected item in the preview pane. You can also double-click.

Using the Advanced Tree View and XPath expressions

The **Advanced** view can be used by advanced users to define complex parsing rules using XPath expressions.

In the Advanced view, you can see all the rules (XPath expressions and corresponding settings) that are used by the XML parser.

Note: XPath will only work completely in the flat tree and advanced tree view.

Two XPath expressions are included by default (default parsing rules):

- With the first default rule (☒ [.] <e> descendant::*) all elements are set to Translatable/External.



- With the second default rule ( `x=" descendant::*/*`) all element attributes are set to Not translatable.

You can't delete these default XPath expressions, but you can modify the associated settings, for example defining all elements as Not translatable/Internal.

When you activate an item using the Flat or Hierarchical views, a corresponding XPath expression is automatically added in the **Advanced View** and the corresponding parsing rule is applied.

Note: New expressions are always added on top of the list and will be parsed last.
See XPath Expression Order for details.

When you deactivate an element, the corresponding XPath expression is removed from the **Advanced View**.

Sample and Legend

To help understand the setting parsing rules in the **Advanced View**, we will use the following sample and give a legend of what the various icons are.

Sample

Consider the following sample:

```
<root date="01/25/2004" manual_ref="LB10X0005">
  <title>
    <tr_unit text_status="NotToTrans">
      <para text_status="ToTrans">Text to translate, sample one</para>
      <para text_status="NotToTrans">Text not to translate, sample one</para>
    </tr_unit>
  </title>
  <section text_status="ToTrans">
    <tr_unit text_status="ToTrans">
      <para text_status="ToTrans">Text to translate, sample <num>three</ num></ para>
      <para text_status="NotToTrans">Text not to translate, sample three</para>
    </tr_unit>
  </section>
</root>
```

Legend

The following table explains what the icons and XPath expressions in the **Advanced View** are:

	Flat
	Hierarchical
	Advanced
	<code>[*] X /root/section/tr_unit/para[@text_status="NotToTrans"]/descendant-or-self::*</code>
	<code>[*] <e> /root/title/tr_unit/para/descendant-or-self::*</code>
	<code>[*] <e> descendant::num/descendant-or-self::*</code>
	<code>[*] <e> descendant::*</code>
	<code>x=" descendant::*/*</code>

Icon/Rule

Legend



 <e>	ML element
 x=""	ML element attribute
 X	User-defined XPath expression
☒  <e> descendant::*	Default rule defining all elements as Translatable/External.
☐ x="" descendant::*/@*	Default rule defining all element attributes as Not Translatable.
☐  <e> /root/title/tr_unit/para/descendant-or-self::*	XPath expression created from the Hierarchical view (start by /root/) <para> element under the tree branch <title> as well as all its children are set as Not translatable/External.
☒  <e> descendant::num/descendant-or-self::*	XPath expression created from the Flat view (prefix descendant::). All <num> elements are set as Translatable/Internal.
☐  X /root/section/tr_unit/para[@text_status="NotToTrans"]/descendant-or-self::*	User defined XPath expression (X icon). All <para> elements under the tree branch <section>\<tr_unit> and for which the attribute text_status is equal to "NotToTrans" are set as Not translatable/External.

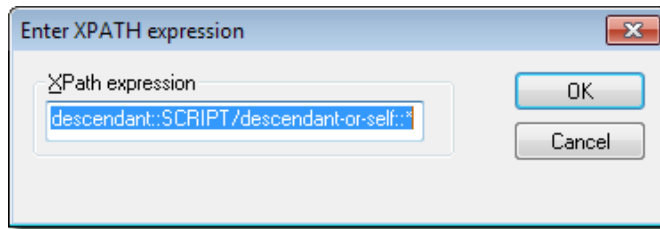
Possible actions

The following table shows the actions for the various parsing settings:

Contextual Menu	Icon/Checkbox	Action
Active Enter	(None)	This menu is disabled in Advanced view. You can use Insert/Delete instead.
✓ Translate t	☐ ☒	Use this menu or checkbox to set an element/attribute as Translatable/Not translatable.
• External x Internal i	 	Use this menu or checkbox to set an element as External/Internal.
✓ Descendants d	(None)	Includes/excludes descendants of the selected element in the parsing rule.
Expand + Collapse -	 	Use this menu or icon to Expand/Collapse a tree branch.
Insert Ins Delete Del Find f	(None)	Use the Insert menu to create a new custom XPath expression. Use the Delete menu to delete an XPath expression.
Find f	(None)	The Find menu is disabled in the Advanced tree view.

XPath Expression Editor

If you double-click an XPath expression, you can edit the expression in the Expression editor:



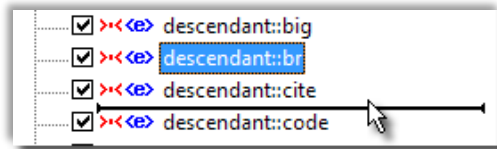
XPath Expression Order

The order in which the different XPath expressions appear in the **Advanced** view, define the order in which the expressions are evaluated by the ML parser. This can have a big impact on the resulting parsing.

By default, the latest action performed in the Flat or Hierarchical view will generate a corresponding rule (that is, XPath expression) on top of the list. The higher up the list of expressions, the later in the parsing process the rule will be applied.

In case you want a rule to take precedence over other rules (and be evaluated later) you can change the ordering of the rules.

To do so, select the rule you want to move and drag it up or down:



Overview of XPath expressions

An XML document can be thought of as having a tree structure, with nodes as its branches. XPath is a language for navigating through nodes (for example, elements and attributes) to find information in an XML document. You use expressions to select nodes or node-sets. An axis defines a node-set relative to the current node.

XPath Axes	Abbreviation	Meaning
ancestor		Selects the current node's ancestors (for example, parent or grandparent).
ancestor-or-self		Selects the current node's ancestors (for example, parent or grandparent) plus the current node.
attribute	@	Selects the current node's attributes.
child		Selects the current node's children.
descendant	/	Selects the current node's descendants (for example, children and grandchildren).
descendant-or-self	//	Selects the current node's descendants (for example, children and grandchildren) plus the current node.
following		Selects everything after the closing tag of the current node.
following-sibling		Selects elements that follow the context node in the same parent element.
namespace		Selects the current node's namespace nodes.
parent	..	Selects the current node's parent.
preceding		Selects everything before the start tag of the current node.



XPath Axes	Abbreviation	Meaning
preceding-sibling		Selects elements that precede the context node in the same parent element.
self	.	Selects the current node.
*		Selects all nodes of the specified name. For the attribute axis, attributes are selected. For the namespace axis, namespace nodes are selected. For all other axes, element nodes are selected.
comment()		Selects all comment nodes.
node()		Selects all nodes.
text()		Selects all text nodes.

Following are some examples of XPath expressions.

Example	Description
.	Selects the context node.
./para	Selects the para descendants of the context node.
..	Selects the parent of the context node.
@*	Selects all attributes of the context node.
@name	Selects the name attribute of the context node.
../@name	Selects the name attribute of the parent of the context node.
para	Selects the para children of the context node.
*	Selects all element children of the context node.
*/para	Selects all para grandchildren of the context node.
para[1]	Selects the first para child of the context node.
para[last()]	Selects the last para child of the context node.
/doc/chapter[5]/section[2]	Selects the second section of the fifth chapter of the doc child of the context node.
para[@type="warning"]	Selects para children of the context node that have type attributes with a value of warning.
para[@type="warning"][5]	Selects the fifth para child of the context node that has a type attribute with a value of warning.
para[5][@type="warning"]	Selects the fifth para child of the context node if that child has a type attribute with a value of warning.
para[@type and not(@type="warning")]	Selects para children of the context node that have that have a type attribute whose value is not "warning".
chapter[title]	Selects the chapter children of the context node that have one or more title children.
//para	Selects all para descendants of the document root.
chapter//para	Selects all para descendants of chapter children of the context node.
descendant::*	Select all descendant elements of the document root.
descendant::*/@*	Select all descendant element attributes of the document root.



Example	Description
<code>//para[@text_status="NotToTrans"]/*[descendant-or-self::*</code>	Select all para descendants of the document root that has a text_status attribute with a value of NotToTrans, and all its descendants.
<code>//tr_unit[@text_status="ToTrans"]/*[para[@text_status="ToTrans"]/*[descendant-or-self::*</code>	Select all para elements that have a text_status attribute with a value of ToTrans and that are children of the tr_unit element that has a text_status attribute with a value of ToTrans, and all its descendants.

Entity Conversions

Entities in numerical form `ñ`; or text identifier form `ñ`; can be converted to their corresponding special characters in the output file.

Every single content entity is displayed in Word as a character to the translator after document conversion, and not as an HTML escape sequence (`ñ`).

You can define which entities to convert using the **Entities to be converted** dialog, which you access by clicking the **Entities...** button in the **Markup Language Assistant** dialog.

Note: Non-converted entities will always be put in Internal style if they appear in translatable text.

The dialog displays the list of all the known **Entity categories** (groups).

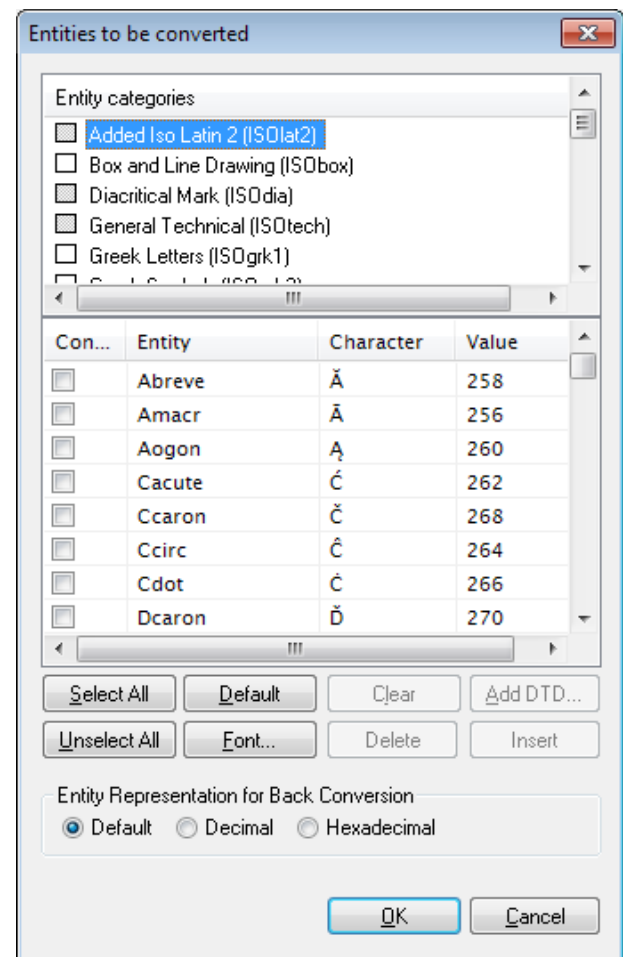
You can use the group check box to select if all or none of the entities of the group need conversion.

When a group is selected on the top list, you will see the corresponding entities in the bottom list and you can use the check boxes in front of each entity to define whether it should be converted or not.

By default, the ML Assistant will convert all `isoLatin1` entities and XML entities to their corresponding special characters.

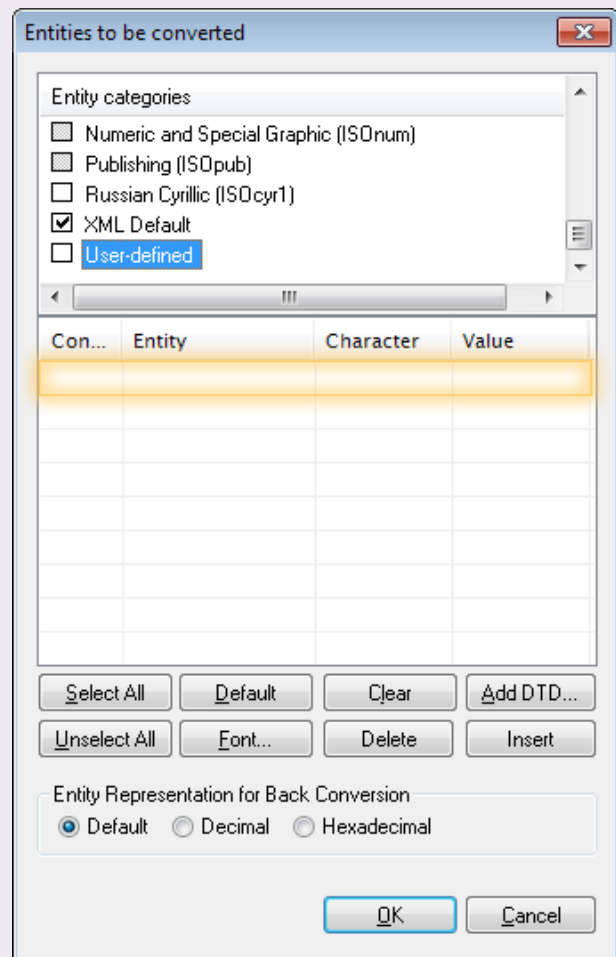
The **Entities to be converted** dialog contains the following options:

Option	Description
Select All	Select all entity categories.





Unselect All	Unselect all entity categories.
Default	Reset to the default entities conversion: IsoLatin1 entities and XML entities.
Font...	Allows you to change the font of the displayed list, which can be useful to display characters that are only supported in specific fonts.
OK	Stores the new settings and closes the dialog.
Cancel	Closes the dialog without storing new settings.
User-defined entities	At the bottom of the entity groups list, there is a user-definable group. When this entity group is selected, you can add your own list of entities to be converted. To add entities, click Insert and enter the entity itself, the character it represents, and its value. To remove them, click Delete. You can also add entities from a DTD file. To add a user-defined entity for the dollar sign, for example, you would type in <code>&#36;</code> in the Entity column and <code>\$</code> for the character.
Clear	This button is enabled when the user-defined entity group is selected. Clears the list of user-defined entities.
Add DTD...	This button is enabled when the user-defined entity group is selected. It displays a dialog allowing you to select a DTD file (.dtd or .ent) and import all the specified entity declarations into the list of user-defined entities.



**Entity representation for back conversion**

This new option allows choosing entity representation during back conversion.

Default: name value, for example: `©`

Decimal: number value, for example: `©`

Hexadecimal: hex value, for example: `©`

Entity Representation for Back Conversion

☒ Default ☐ Decimal ☐ Hexadecimal

Text Filter

Translation Workspace provides a Text conversion filter that allows you to generate tagged files (Translation Workspace RTF/XLIFF) from text files in different formats (for example, plain text, properties and INI files) using tagging rules provided in Translation Workspace .ltx configuration files. These rules are defined in a filter profile for every different input format by means of a tree of Regular Expressions.

The Text filter includes default profiles for the some text file types. The users can add new conversion profiles for other text file types to the list of current profiles and define the parsing rules for them.

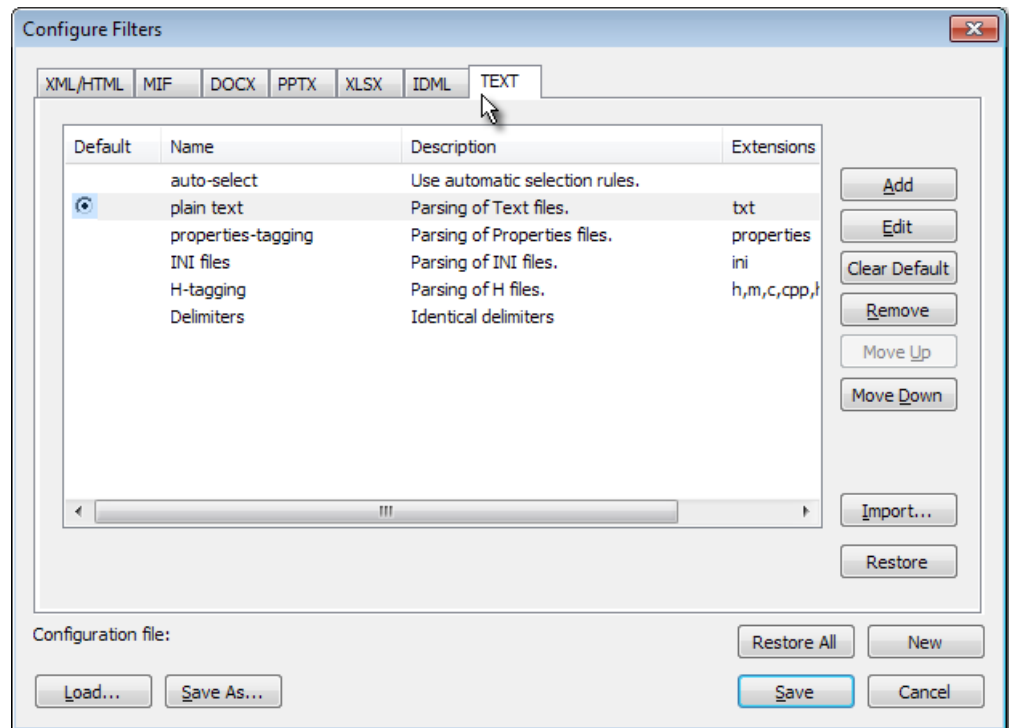
By selecting a profile and clicking **Edit** on the **TEXT** tab you can launch the **Text Filter Profile Edit** dialog, which allows you to define the parsing rules of the different profiles and provides an intuitive and interactive way to identify localizable text file content.

Supported Document Encodings

The Text conversion filter supports the most common text encodings. Files with all supported encodings are converted into correct, completely displayed, RTF or XLIFF.

Configuring the Filter

To configure the Text conversion filter, click the **Configuration...** button in the Document Conversion dialog (at **Documents > Convert** in the Word Plug-in or **Conversion tasks > Convert** in the Translation Workspace Tools) and select the **TEXT** tab.



The Configure Filters dialog **TEXT** tab lists all the available profiles (that is, parsing rule collections) in the current configuration file and allows you to add, modify, or delete each profile. The Text conversion rules are defined using (Perl syntax) regular expressions.

Profile Selection

A profile is a set of parsing rules (that include parsing settings and regular expressions) defining what is translatable and what is non-translatable in the source document. It also sets non-localizable parts of the files to be marked as internal or external in the resulting parsed RTF or XLIFF files.

A Configuration file (with the extension .lmx) usually includes multiple text filter profiles.

You can control which profile is applied during the conversion process by:

- Selecting a profile from the list.
- Selecting the **auto-select** profile option from the list. Rules (see below) are used to automatically select a profile during the conversion process

If the auto-select profile option is checked in the profile list, the Text conversion tool allows you parse text files with different document types (that is, different file extension) together. For each text file (type) to be parsed, the matching profile will be searched from the list of available profiles.

The correct profile is applied by following these auto-select rules (steps 2 and 3 are triggered only when the filter is used via the API):

1. If more than one matching profile exists for a given document type (file extension), the first matching profile found in the list is applied. This profile is used to process all files that belong to the relevant document type.
2. If no matching profile available for a given document type exists, Translation Workspace checks whether one of the profiles is set as the Default in the list, and the Default profile is used to process the document. (Through API only.)



3. If no matching profile available for a given document type exists, and no Default profile has been set, the file is parsed with the first profile from the top of the profile list.
(Through API only.)

Configure Filter Dialog Options

The following table describes the options specific to the TEXT filters. For descriptions of the options common to all filters, see the **Configure Filters Dialog** topic in the Translation Workspace Tools help.

Option	Description
New	Click New to create a new Configuration file. This will remove all existing profiles from the Configuration file and add default profiles with default settings to the list.
Select Profile	To select a profile click its line. When the line is highlighted the profile is selected.
Add Profile	Click the Add button to add a new profile to your Configuration file.
Edit Profile	The Edit button opens up the Text Filter Profile Edit (Profile editor) dialog, where you can define your parsing rules and get a real-time preview of a sample file with the defined settings. You can also double-click on a profile to start editing a profile.

Option	Description
Set Default	Set the profile as the default profile. The default profile is indicated with a radio button in the <i>Default</i> column. Click the Set Default button to define the default profile. Click the Clear Default button to remove the selection.



Remove	To remove a Profile from the list, select it from the list and click this button.
Move up	To move a Profile up in the list, select it from the list and click this button. The higher a profile appears in the list, the higher its priority (provided the auto-select option was chosen).
Move down	To move a Profile down in the list, select it from the list and click this button.

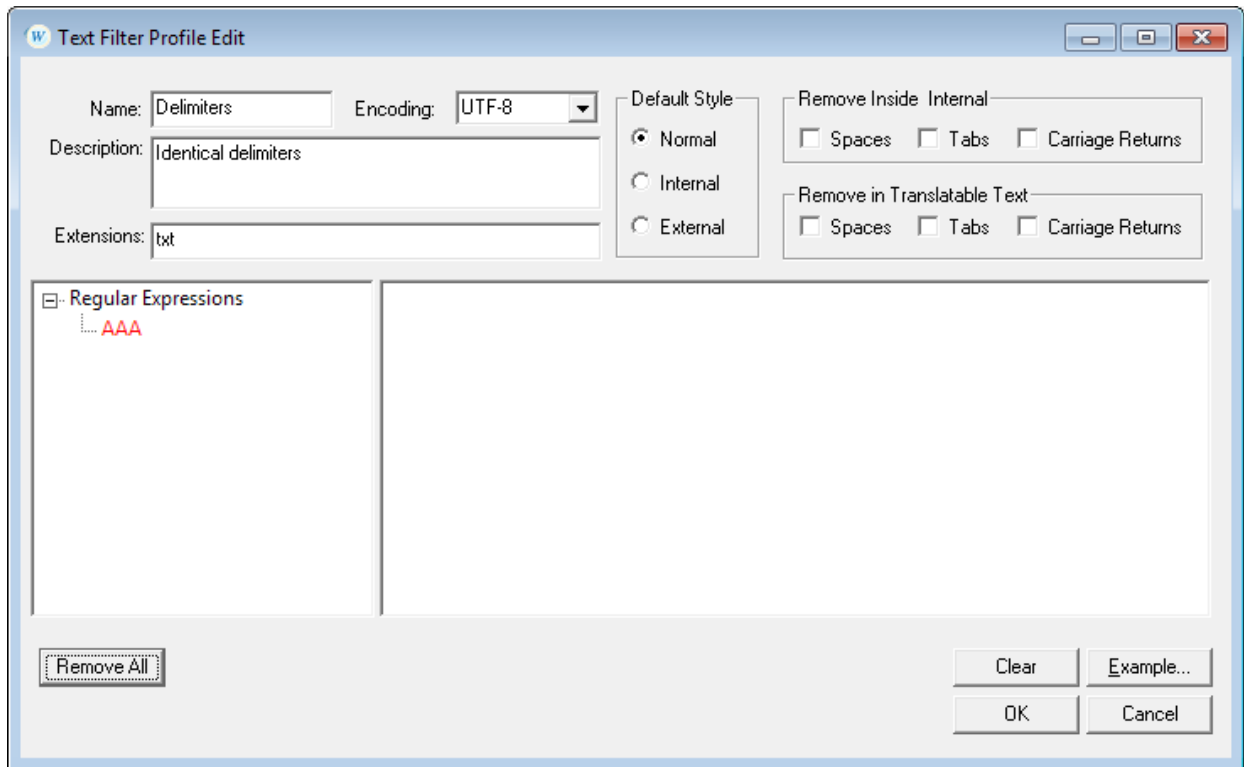
Loading a .Imx File

Click **Load...** to load a new configuration .Imx file from your local computer. If the file does not include a "Text" filter profile, or has no other profiles, click **TEXT** tab and an empty list control is displayed. To add a new profile, click **Add**.

Text Filter Profile Edit Dialog

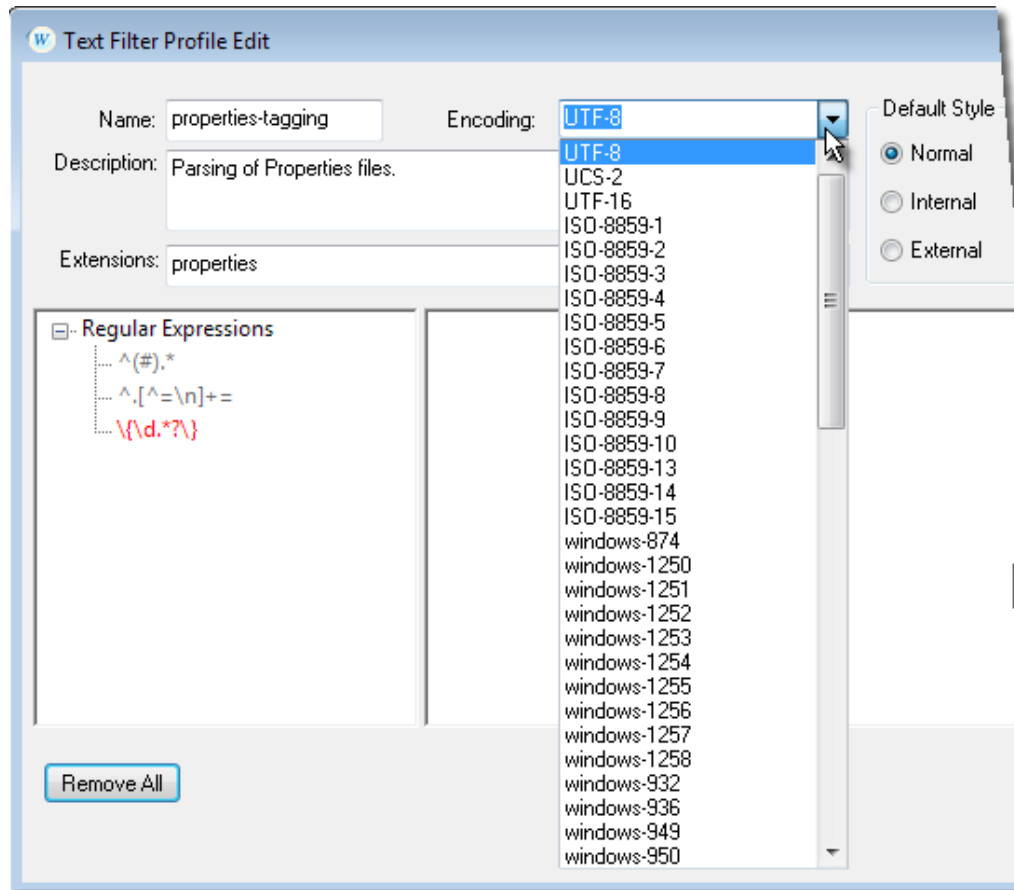
The **Text Filter Profile Edit** dialog includes three panels:

- The header panel is at the top of the dialog.
- The middle panel is split to the **Tree view** for rules and a **Preview** pane for an example file.
- The bottom panel includes buttons for rules and example file functions.



The Header Panel of the Profile Edit Dialog

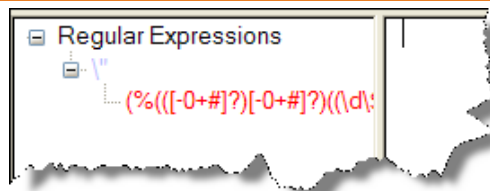
The Header panel includes the topmost part of the dialog with the following fields and selections.



- The **Name**, **Description**, and **Extensions** fields, where users can input strings. The contents of these fields are displayed also on the **TEXT** tab of the **Configure Filters** dialog. The **Extensions** field can include several file extensions, each separated by a comma.
- The **Encoding** drop-down list, where the encoding setting for the text can be selected. This setting should reflect the encoding of the original text files. The setting affects the way the files are parsed and also how the enabled parsing rules affect the sample file contents in the Preview field.
- The **Default Style** group, where one of the possible styles can be checked as the default parsing style for the current profile. All text in the files is then parsed with this style, except those parts of text that are parsed by the specific rules.

The encoding setting for the original files can be selected in the Encoding drop-down list, which lists all the supported encodings.

The Tree view in the Profile Edit Dialog



The **Tree view** displays rules under the Regular Expressions root item, which includes each rule as a child item.

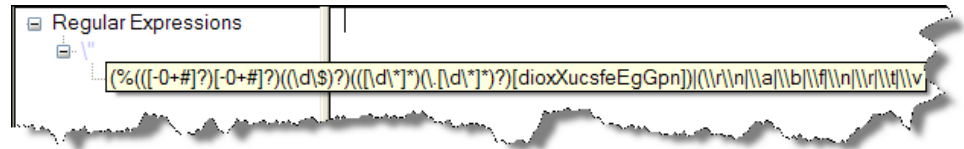
Rules can be nested. Each nested rule is displayed as a child element of the rule it is nested in. A rule that has children rules has a square box on its left side with the x/- signs to

display/hide the children rules. Click on the square box to expand or collapse the tree view.



When the expression is too long to be fully displayed in the tree view, there are several ways to view the whole expression.

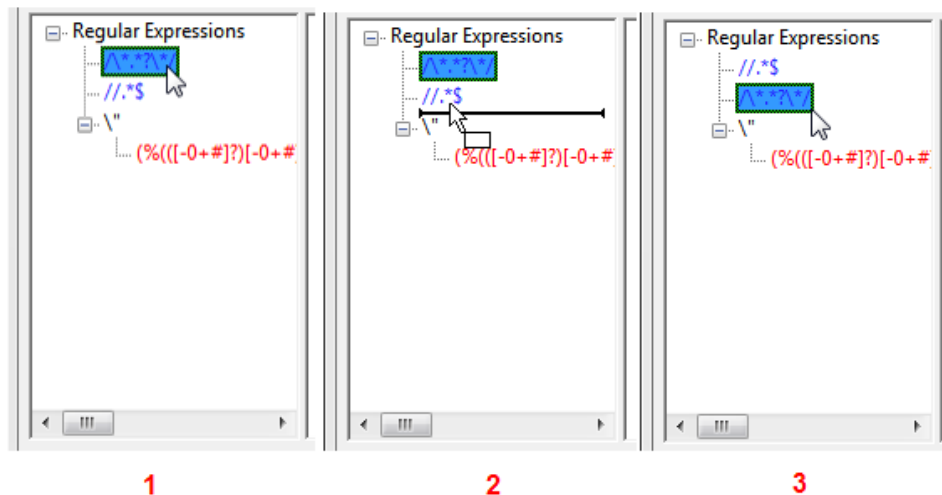
1. You can drag the field separator between the Tree view field and the Preview field to the right to fit the expression into the Tree view.
2. You can select (click) the rule, and point at the expression. The whole expression is displayed on the dialog across the field separator and over the Preview field as a tooltip.



3. You can double-click the expression (or right-click the line and select Edit in the context menu) to open the **Rule Edit Dialog**.

Visual Edit of the Tree

You can rearrange the sibling rules in the profile by selecting a rule line, dragging the rule to another place in the sibling rules list and dropping it to its new place (marked by the vertical hairline):

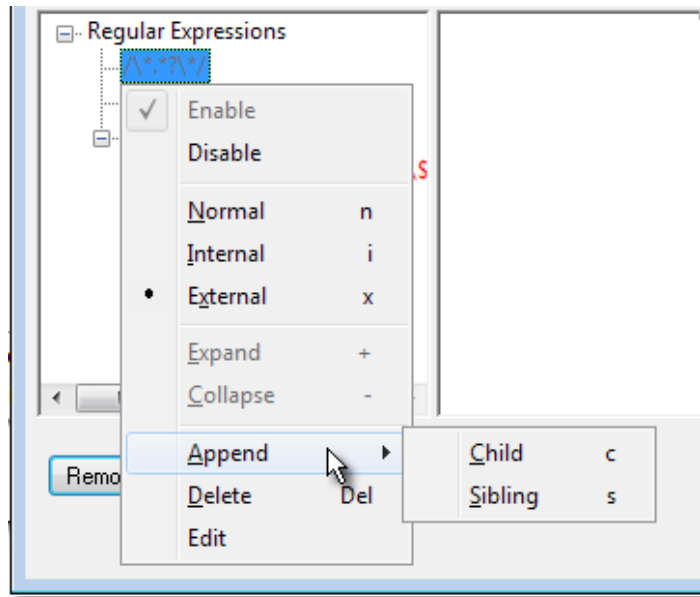


The relocated rule is displayed in the new place in the Regular Expressions Tree view.

Note: You cannot transform a sibling rule to a child rule by dragging it under another sibling rule, and you cannot transform a child rule to an upper-level sibling rule by dragging it to a level above.

Context Menu

Click on an item (that is, a rule line) to select it. Right click on a rule line to select the rule and open the context menu.

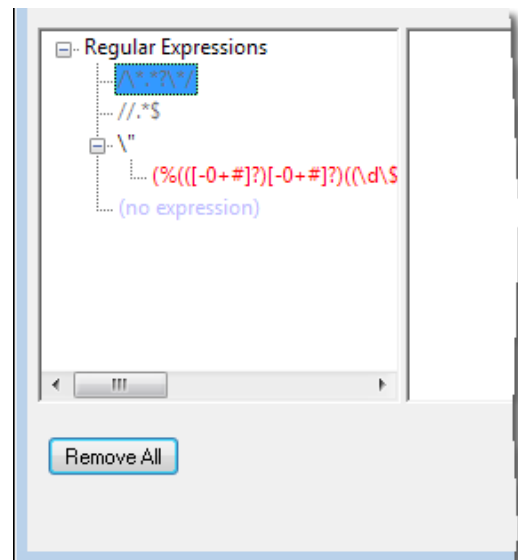


In the context menu you can:

1. Enable or disable the selected rule.
2. Select which style the text parsed with the rule is formatted to by selecting **Normal**, **Internal**, or **External** (Normal is translatable text; Internal is movable, non-translatable text; External is static non-translatable text).
3. Expand or collapse the selected nested rule by selecting **Expand** or **Collapse**. (If the selected rule does not have any children, these context menu commands are displayed in gray text).
4. Append a child (a nested rule, applied only inside text strings defined in the nesting parent rule) or a sibling (non-nested rule) to the currently selected item (rule) by selecting **Append|Child** or **Append|Sibling**. The **Append** command adds a new item to the rules tree with the name "(no expression)".

To define the details of the new rule and enable it, double-click the "(no expression)" line or right-click the line and select **Edit** in the context menu, to open the **Rule Edit Dialog**.

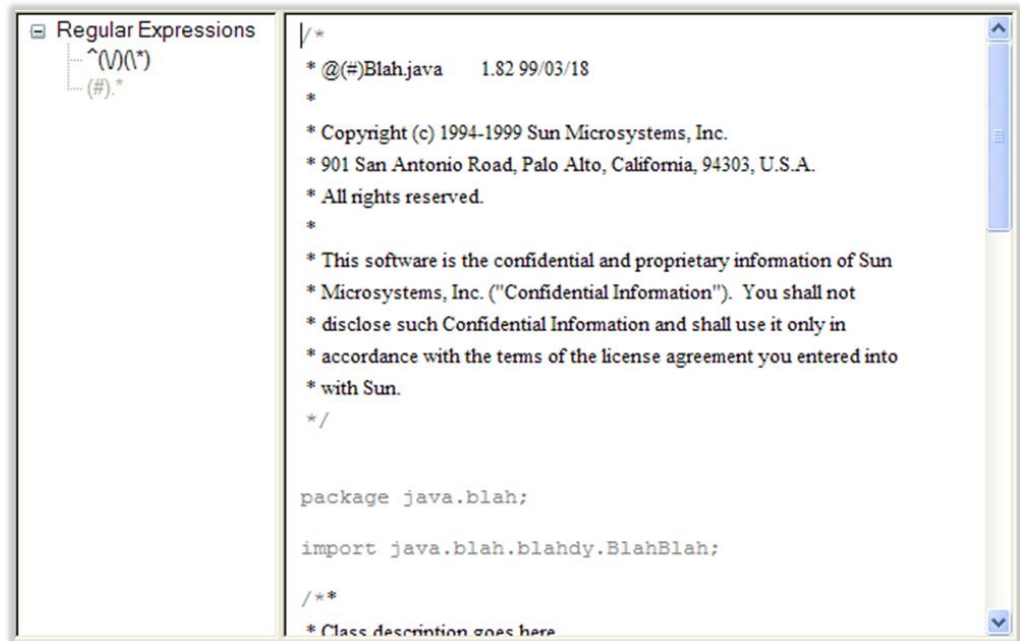
5. Delete the current selected item (rule) by selecting **Delete**.
6. Edit the selected item (rule) by selecting **Edit**. The Edit command opens the **Rule Edit Dialog**.





Preview of a Sample File

The Preview window is for using a sample file to display in real time how the enabled regular expressions and other settings affect the parsing of the file. All the additions and changes to the expressions and settings of the **Text Filter Profile Edit** dialog are displayed immediately in the Preview window. Enabling and disabling a rule, or any change of regular expressions, changes in the **Encoding** setting, in the **Default Style** field selection, have immediate effects to the Preview window display.



Drag and drop a file into the Preview window or click **Example** on the bottom panel to browse to and load a sample file to the Preview window.

The sample file name persists. Its content will appear again if the **Text Filter Profile Edit** dialogs are closed and reopened, or next time start Word client and edit any profile.

The Text Filter Profile Edit Bottom Panel

The bottom panel of the **Text Filter Profile Edit** dialog includes five buttons with commands affecting the Tree view, Preview windows and the current Text Filter profile.

Button	Description
Remove all	Click Remove All to remove all regular expression rules on tree view. The <i>Preview</i> window is immediately updated.
Clear	Click Clear to remove the current example file from the <i>Preview</i> window. After this the <i>Preview</i> window is empty.
Example	Click Example... to load an example file for preview. You can also drop a file on the <i>Preview</i> window.
OK	Click OK to save all changes and close the <i>Text Filter Profile Edit</i> dialog.
Cancel	Click Cancel to cancel all changes made while the <i>Profile Edit Dialog</i> has been open and close the dialog.



Rule Edit Dialog

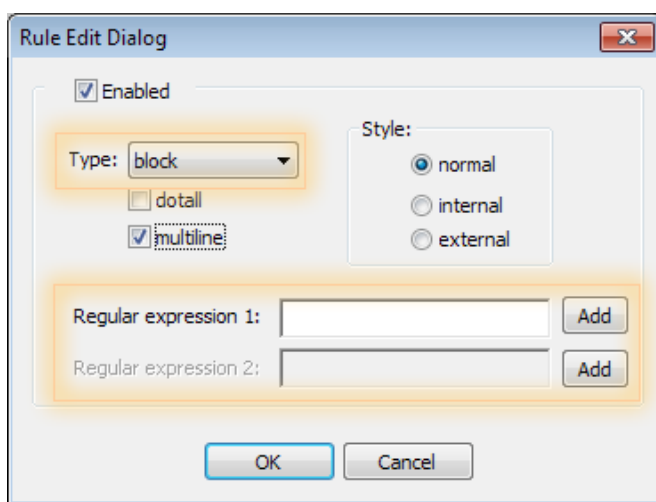
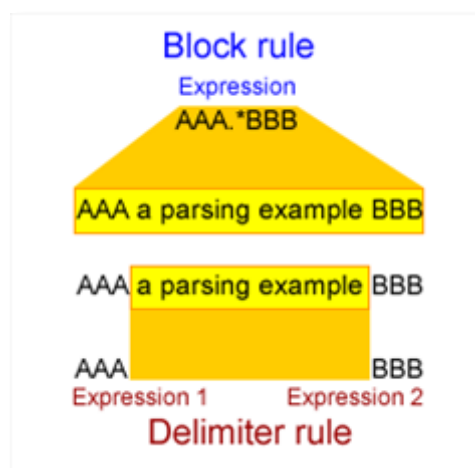
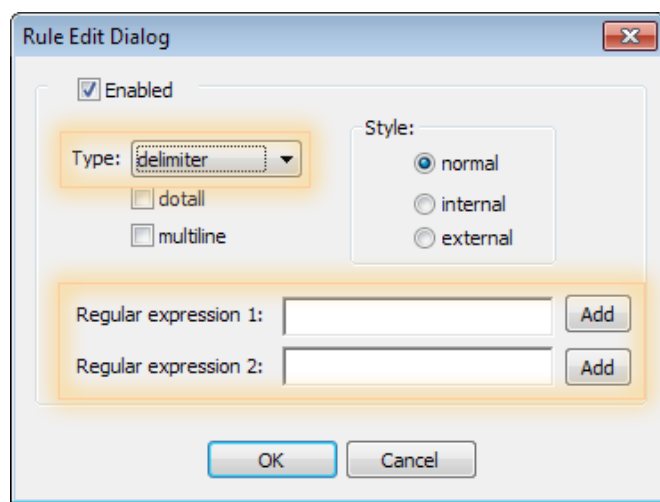
Use the **Rule Edit Dialog** to create a new rule or modify the details of an existing rule. You can also enable and disable a rule by checking or de-checking the **Enabled** option in the **Rule Edit Dialog**.

Double click the rule or the "(no expression)" line in the Tree view to display the **Rule Edit Dialog** (or right-click the line and select the **Edit** command in the context menu).

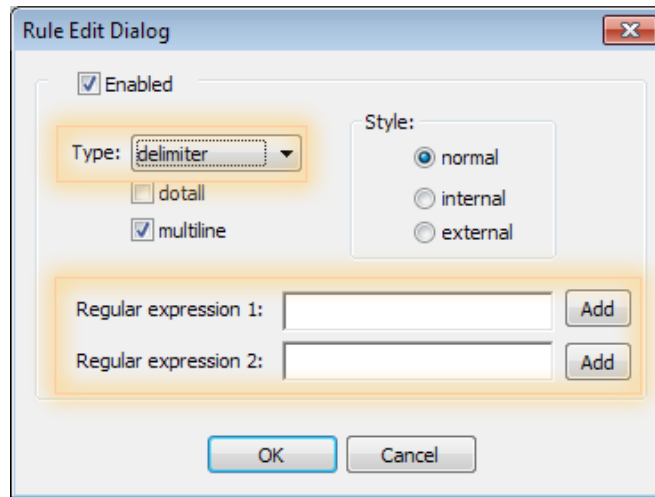
There are two types of parsing rules, block and delimiter:

1. The **block** type includes only one regular expression, which identifies a block of text.
2. The **delimiter** type includes two regular expressions. The first expression defines the starting delimiter of a text block to be parsed, and the second expression the ending delimiter of the same text block. In this type the delimiters are not parts of the text block identified by the parsing rule.

The difference between the two types is illustrated in the picture on the right:



The **Rule Edit Dialog** includes two Regular expression fields. When **block** is selected in the **Type** field drop-down list, only the **Regular expression 1** field is active.



When **delimiter** is selected in the **Type** field drop-down list, the **Regular expression 2** field becomes active also.

The starting delimiter of the text block to be parsed is inserted to the **Regular expression 1** field. The ending delimiter of the text block is inserted to the **Regular expression 2** field.

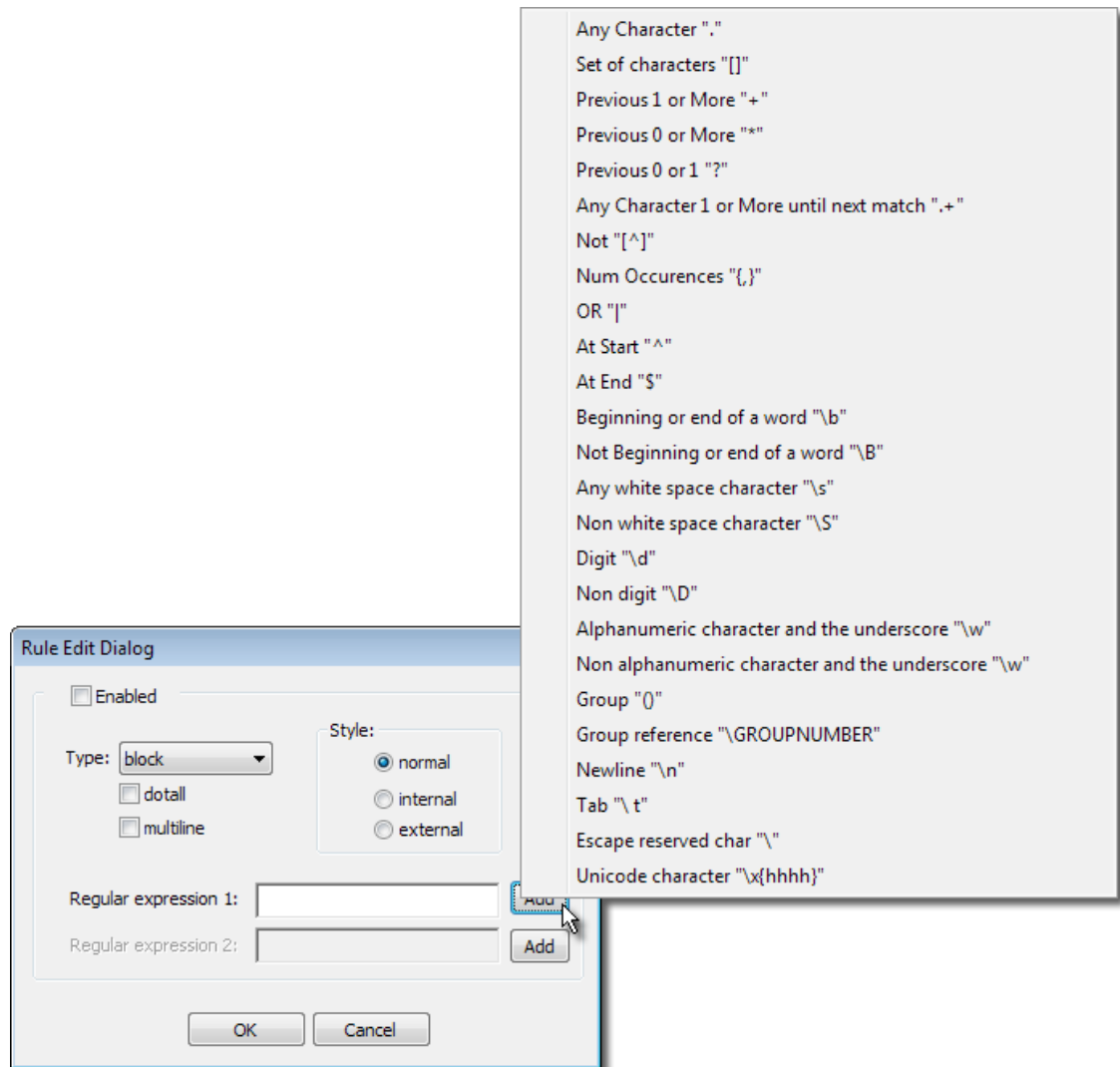
The following table describes the options and selections specific to the **Rule Edit Dialog**.

Button or selection	Description
Enabled	When checked, this option enables the rule to be used when the current profile is used for parsing files. If this option is not checked, the rule is not applied during parsing. This option is unchecked by default.
Type	The <i>Type</i> option is used for setting the rule type either to block or to delimiter . <i>Block</i> includes only one regular expression and <i>delimiter</i> includes two. This option is set by default to delimiter .
DotAll	This option controls the meaning of the <i>any character</i> sign, which is the period character "." in the regular expressions. This option is unchecked by default. When this option is unchecked, the period character means <i>any character</i> except the <i>newline character</i> , which is "\n". When this option is checked, the period character covers also the meaning of the <i>newline character</i> "\n".
Multiline	When checked, the <i>At start</i> "^" and <i>At end</i> "\$" characters in the regular expressions match the beginning and the end of any line, not just the beginning and the end of the whole regular expression string. When unchecked, the rule the <i>At start</i> "^" and <i>At end</i> "\$" characters in the regular expressions mean only the start and end of the text part the expression parses. This option is unchecked by default.
Style	When the <i>Style</i> is set as Normal , the parsed part of the file defined by the current rule is set as normal localizable text. When the <i>Style</i> is set as Internal , the parsed part of the file defined by the current rule is set as internal text, which is movable non-translatable text that can be included in the translatable segments.



Button or selection	Description
	When the <i>Style</i> is set as External , the parsed part of the file defined by the current rule is set as external text, which is static non-translatable text, which cannot be included in the translatable segments. This option is set by default to Normal .

Adding Regular Expression Items



You can add a regular expression in the **Rule Edit Dialog** to the **Regular expression 1** or to the **Regular expression 2** fields by clicking **Add** next to the field. The regular expressions popup list is displayed for you to select a regular expression item. You can add as many regular expression items to the field as necessary to create a working expression.

When a new expression item is added, the **Rule Edit Dialog** recognizes the current cursor position in the active regular expression field and current selection start and end positions if a part of the expression is selected.



- ❑ If just the cursor is in the field, but no part of the expression is selected, the regular expression item selected from the list is inserted to the current cursor position.
- ❑ If the **Block** type expression is selected and a part of the expression is selected, the selected part is replaced with the regular expression item selected from the list.
- ❑ If the **Delimiter** type expression is selected and a part of the expression is selected, the new expression item (such as "()", or "[]") is inserted before and after the selected part.

The expression strings in the **Regular expression 1** and **Regular expression 2** fields are also directly editable. You can copy and paste expressions and their parts as text strings to and from them by using Copy and Paste commands. You can also empty the field contents partly or fully by selecting the string or its part and pressing **Del** or using the **Delete** command.

Applying Rules during Parsing

If the configuration file provided for the filter is not valid (for example if the filter configuration has been imported from a .lmx file), or any other error occurs in reading the configuration file, the application logs the error and stops processing the files to be parsed.

Text conversion filter rules support Perl Regular Expression syntax. For detailed information see [Perl Regular Expression Syntax](#).

The Text Filter splits all text to blocks (strings), corresponding to matching rules. Each text block is marked if some rule applies to it.

Both **Block** and **Delimiter** types of rules are applied following these principles:

1. All enabled rules of the profile are applied.
2. The filter interprets the rules and tries to apply each rule to all text blocks which have not been already marked by any previous rule.
3. When a rule matches the file contents, the filter divides the text to blocks, separating the text which matches the rule from the rest of the text.
4. The sibling rules are applied in the order of their appearance in the configuration file, from top to bottom.
5. Children rules are applied to the text blocks parsed by the parent rule. They are applied in the order of their appearance in the configuration file, from top to bottom.
6. Overlapping sibling rules do not work.
7. Delimiters in the Delimiter type rules are not interpreted as a part of the matching segment. A **delimiter** can be an ending delimiter in a rule and a starting delimiter in another rule.

Examples of Applying Profiles with Different Parsing Rules

To demonstrate how the different rule types and rule combinations work, we give some examples. All examples show how the same text is parsed during conversion.

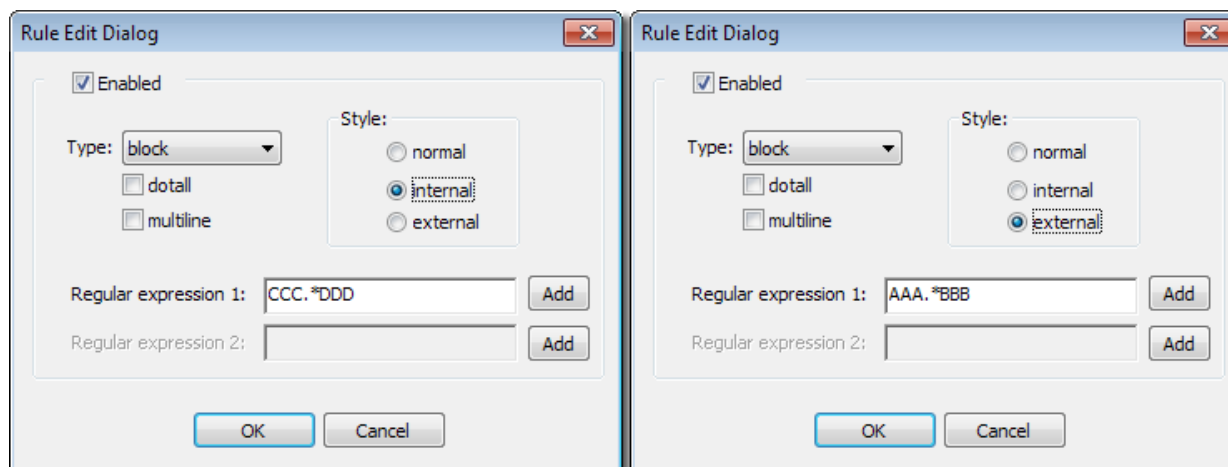
The text in the text file is in all examples this:

This is AAA a block test BBB with some CCC test strings DDD

The Default style is set to **Normal** in all examples.

Example 1: Two Block Sibling Rules

The profile includes two **block** rules:



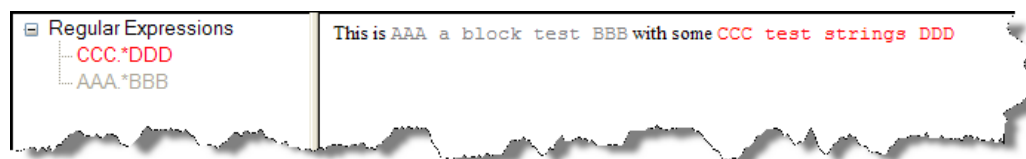
Applying the rules:

1. The first rule is applied and its expression matches the segment "CCC test strings DDD", which is parsed as internal text.
2. The second rule is applied to the remaining text "This is AAA a block test BBB with some".
3. The expression in the second rule matches the string "AAA a block test BBB", which is parsed as external text.
4. There are no more rules to apply.

Result:

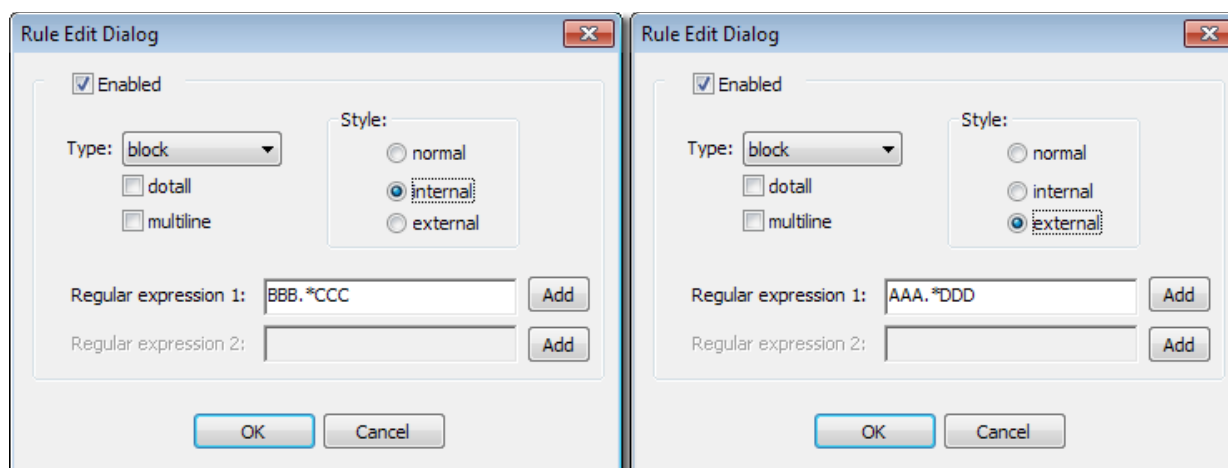
This is AAA a block test BBB with some CCC test strings DDD

The Tree view and Preview field display:



Example 2: Two Block Sibling Rules

The profile includes two **block** rules:





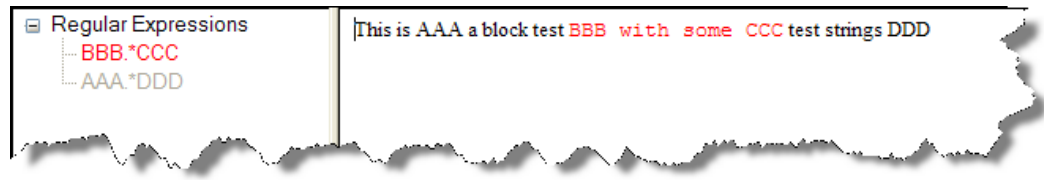
Applying the rules:

1. The first rule is applied, and its expression matches the string "BBB with nested CCC", which is parsed as internal text.
2. The second rule is applied to the remaining text "This is AAA a block test " and "test strings DDD".
3. No text matches the expression in the second rule.
4. There are no more rules to apply.

Result:

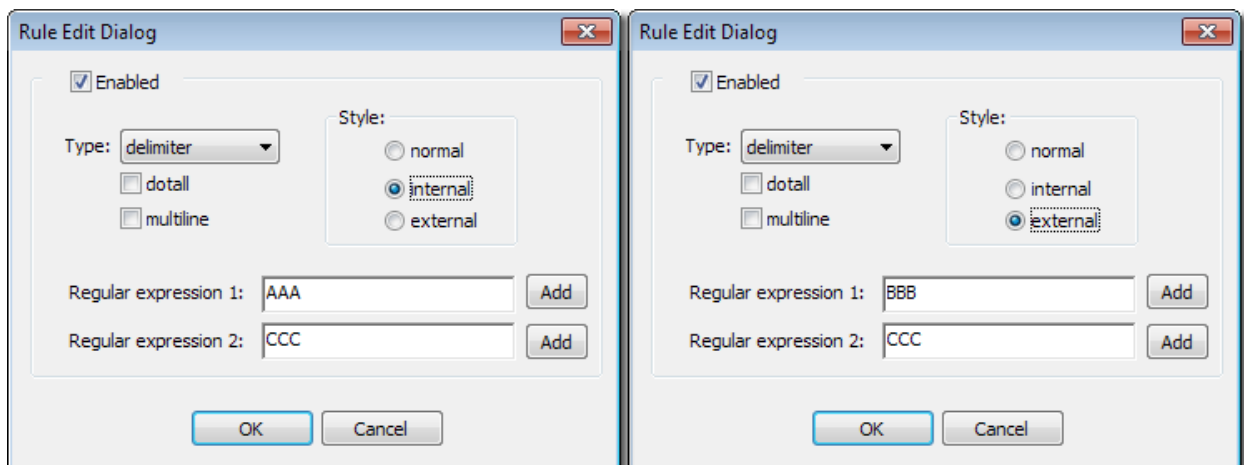
This is AAA a block test BBB with some CCC test strings DDD

The Tree view and Preview field display:



Example 3: Two Delimiter Sibling Rules

The profile includes two **delimiter** sibling rules:



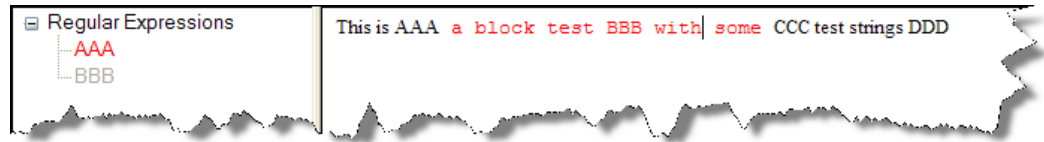
Applying the rules:

1. The first rule is applied, and its expression matches the string " a block test BBB with some ", which is parsed as internal text.
2. The second rule is applied to the remaining text "This is AAA" and "CCC test strings DDD".
3. No text matching the second rule is found.
4. There are no more rules to apply.

Result:

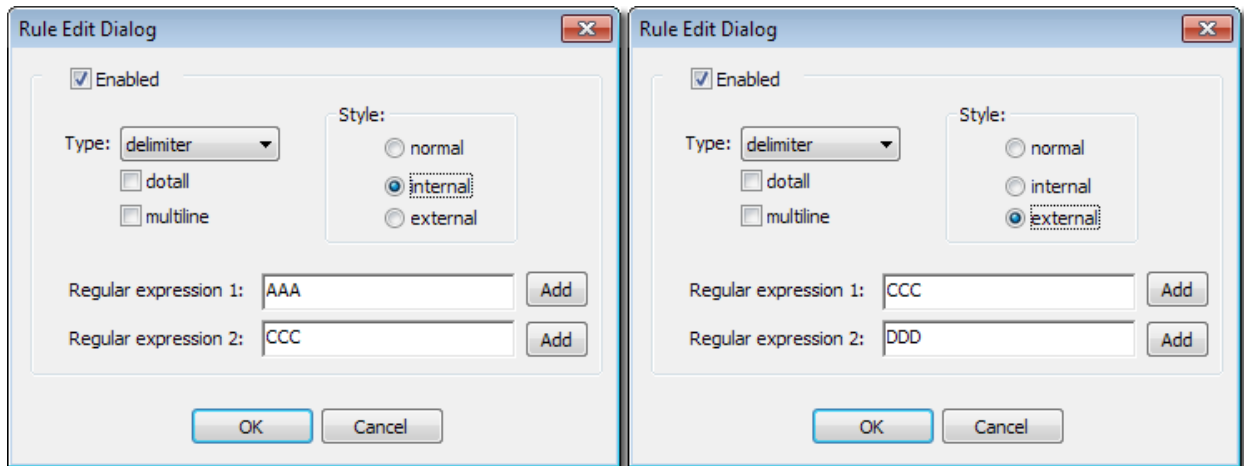
This is AAA a block test BBB with some CCC test strings DDD

The Tree view and Preview field display:



Example 4: Two Delimiter Sibling Rules

The profile includes two **delimiter** rules (the ending delimiter of the first rule is the starting delimiter of the second rule):



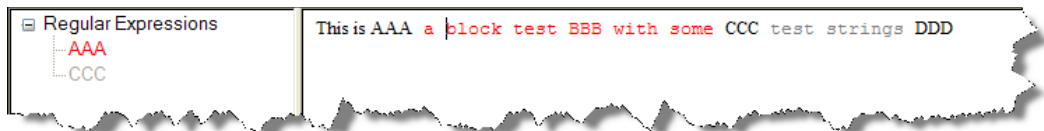
Applying the rules:

1. The first rule is applied, and its expression matches the string " a block test BBB with some ", which is parsed as internal text.
2. The second rule is applied to the remaining text "This is AAA" and "CCC test strings DDD".
3. The expression of the second rule matches the string " test strings ", which is parsed as external text.
4. There are no more rules to apply.

Result:

This is AAA a block test BBB with some CCC test strings DDD

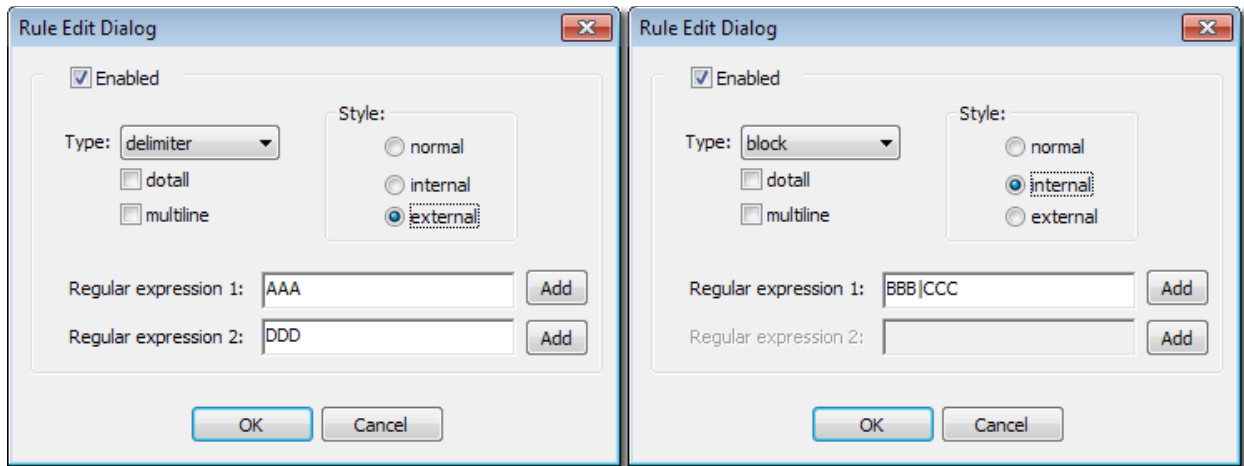
The Tree view and Preview field display:





Example 5: A Delimiter Rule Including a Nested (Child) Block Rule

The profile includes two rules (a **block** rule is nested in a **delimiter** rule):



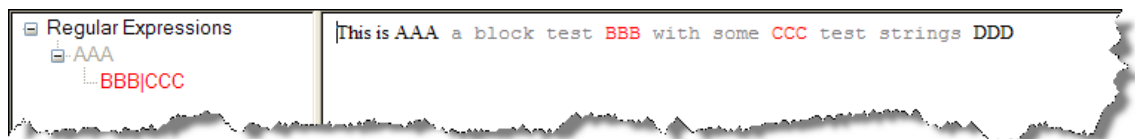
Applying the rules:

1. The first rule is applied; its expression matches the string "a block test BBB with some CCC test strings", which is parsed as external text.
2. The second rule (nested) is applied to the text matching the first rule "a block test BBB with some CCC test strings".
3. Two sub-strings match the expression in the second rule "BBB" and "CCC", which are parsed as internal text.
4. There are no more rules to apply.

Result:

This is AAA a block test BBB with some CCC test strings DDD

The Tree view and Preview field display:

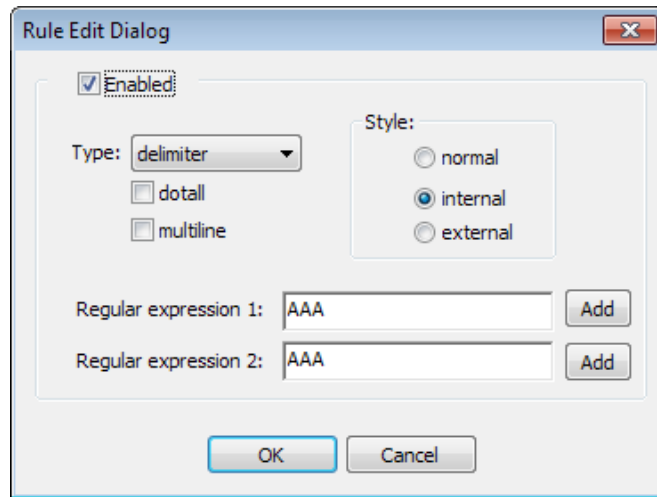


Example 6: One delimiter rule with identical delimiters

Because of the nature of this delimiter rule, in this example the text in the text file differs from the other examples.

This is AAA a block test AAA with some AAA test strings AAA

In this example the start and the end delimiter are identical.



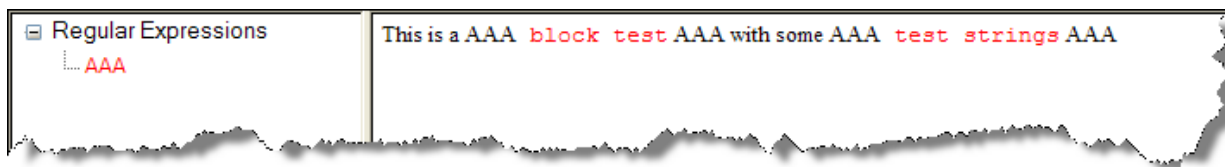
Applying the rule:

1. The rule is applied; its expression could match three strings:
"a block test",
"with some", and
"test strings".
2. When the beginning delimiter of the rule expression is the same as the end delimiter, the rule is applied only to the first string, when two consecutive strings share a delimiter.
3. Because the beginning delimiter of the second string is the same as the end delimiter of the first string, the rule is not applied to the second string.
4. The rule is applied only to the first "a block test" and the third expression in the text "test strings" matching the rule. Both expressions are parsed as internal text.
5. There are no more rules to apply.

Result:

This is AAA a block test AAA with some AAA test strings AAA

The Tree view and Preview field display:



Example 7: Making Comments Translatable in a Java Source File

Let's say you want to have the comments in a Java source file presented for translation (but, of course, leave the code itself alone). You could extract for translation everything delimited by "/" and "*".

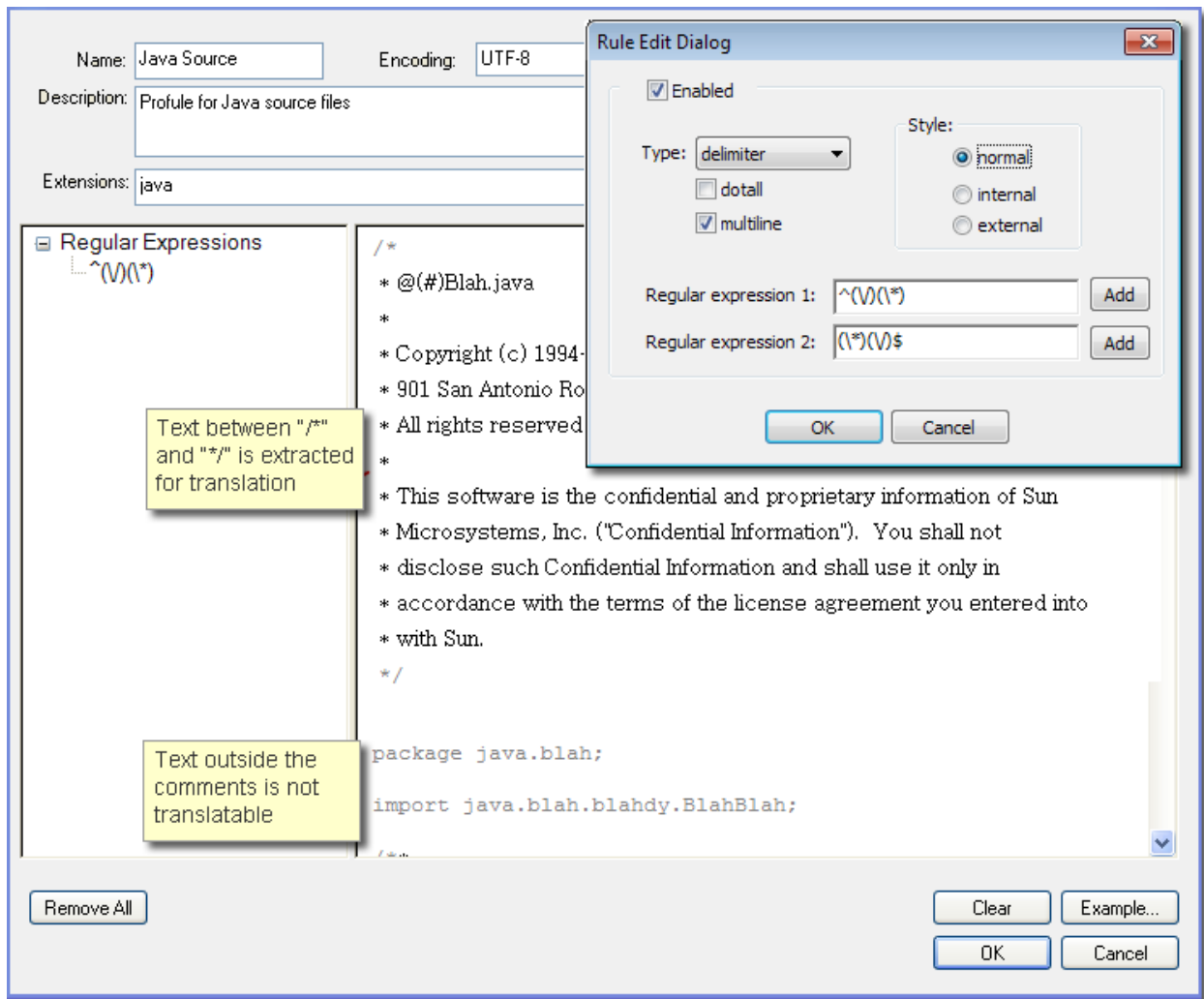
To accomplish this, do the following:

1. In the **Rule Edit Dialog**, select "delimiter" and "multiline" for **Type**.



2. Select "normal" for **Style** (normal text is presented for translation).
3. For **Regular expression 1**, specify `^(\\)(\\*)`. Note that the characters in the delimiter are escaped with `"\"` and the delimiter itself is in parentheses.
4. For **Regular expression 2**, specify `(\\*)(\\)$`. Note that the expression comes before the end indicator (`$`).
5. Be sure "Enabled" is selected at the top of the dialog.
6. Click **OK** to quit this dialog, and check "External" for **Default Style** from top of the frame window.
7. Click **Example** to load your Java file.

The result will look like this:



Previewing TXT documents converted with TXT Filter

You can run preview from XLIFF Editor for the XLZ files created from source TXT, HTML, and XML files rendered in your browser.

You can also specify explicitly the program in which you want to open the generated preview files:

```
C:\Program Files (x86)\Notepad++\notepad++.exe "${file}"
```



- The XLZ file is back converted using the embedded source document
- The generated final document is opened in the web browser or in the specified program.

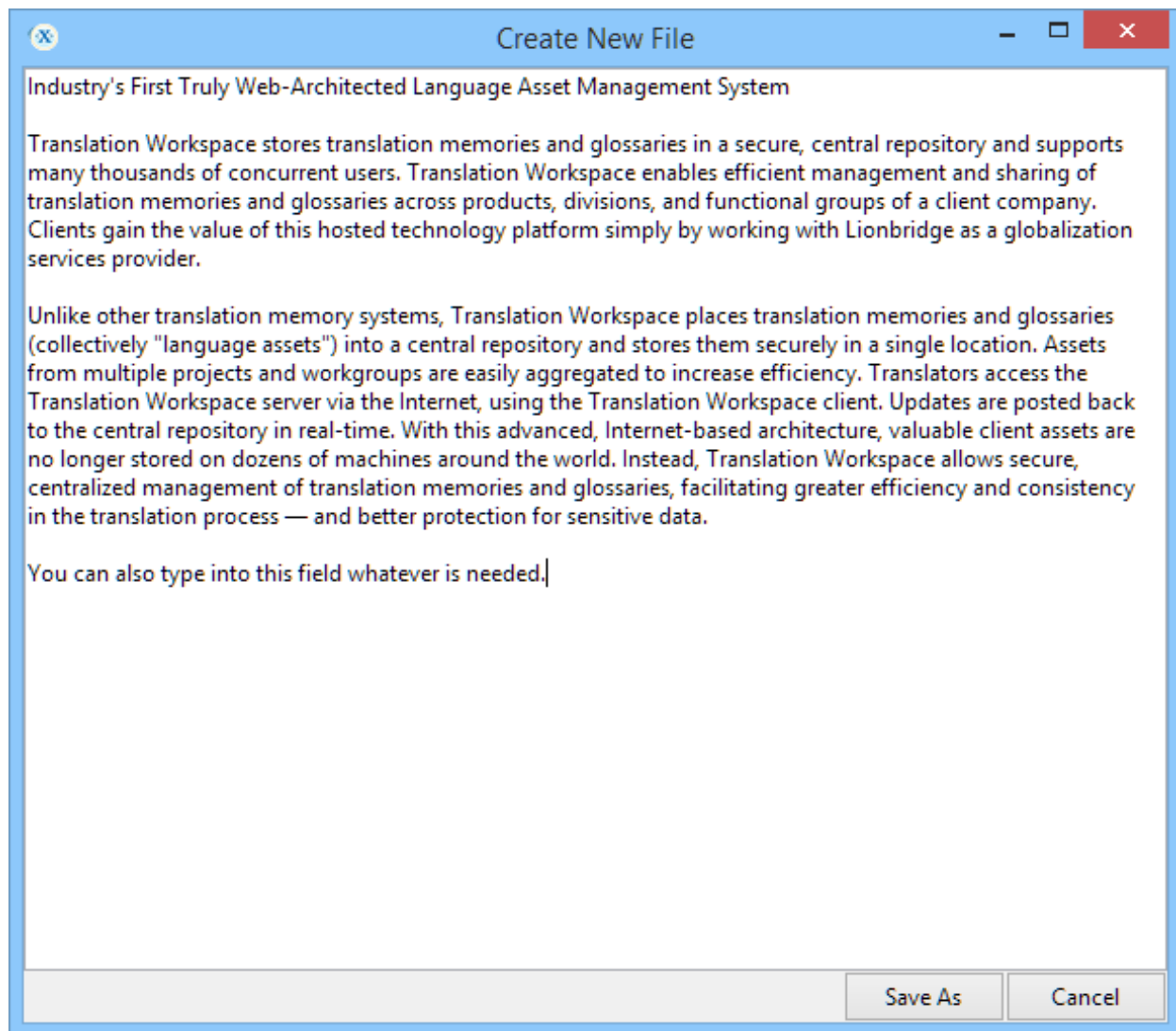
An error message is displayed (or returned),

- If any problem occurs during the back conversion
- If the file cannot be opened in the associated application
- If back conversion fails

You can generate a new preview when a previous preview file is still open.

Using the Create New File dialog to Create XLZ files from text

You can create a new file by typing or pasting plain text to the Create New File dialog in XLIFF Editor and saving it as an XLZ file.

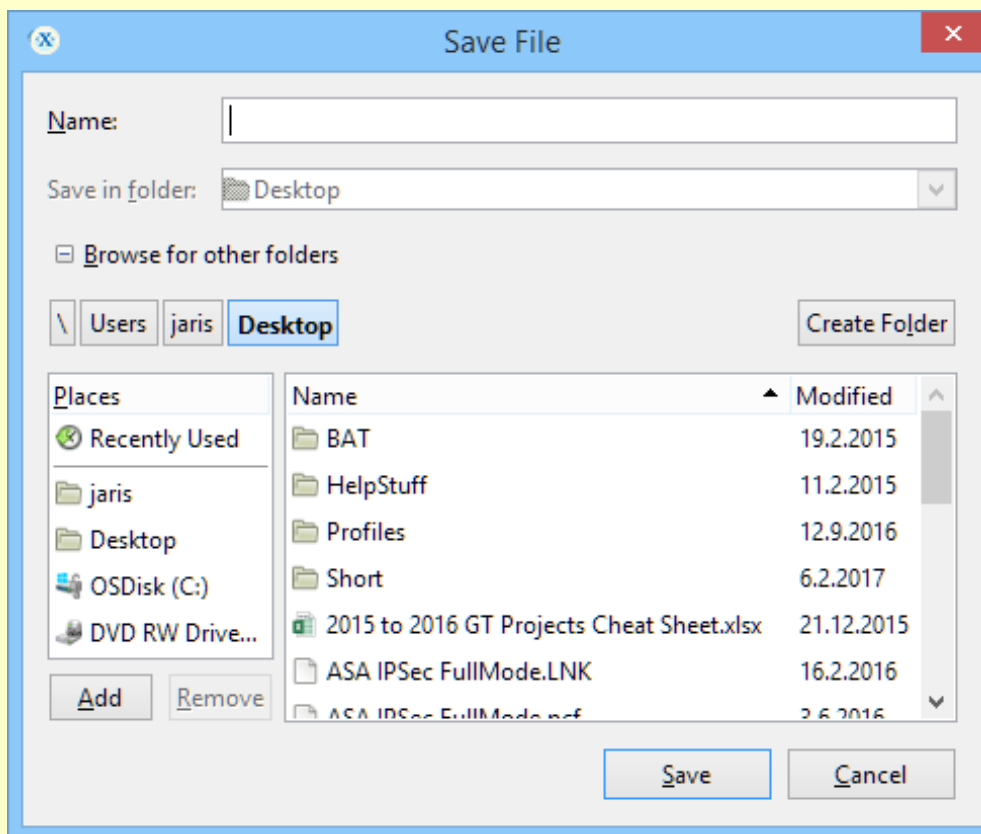




When you select the **File > New** menu command, a Create New File dialog is displayed, containing:

- A Unicode plain text editor to which you can type or paste any text (plain text).
- A Save As button (grayed out until some text is added to the editor field).
- A Cancel button.

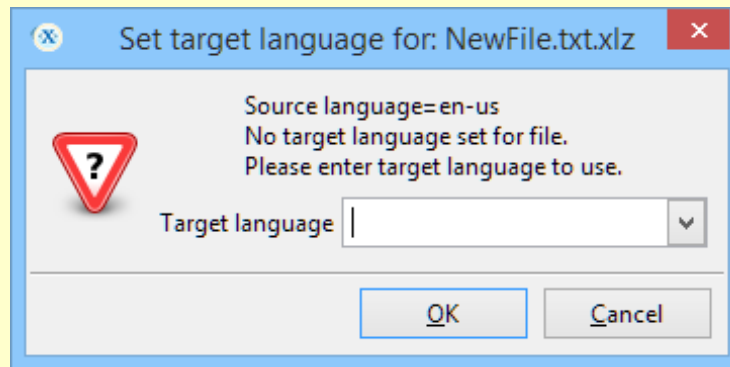
Clicking **Save As** in the **Create New File** dialog displays a **Save As** dialog where you can choose where to store the XLZ file and provide a name for the file.



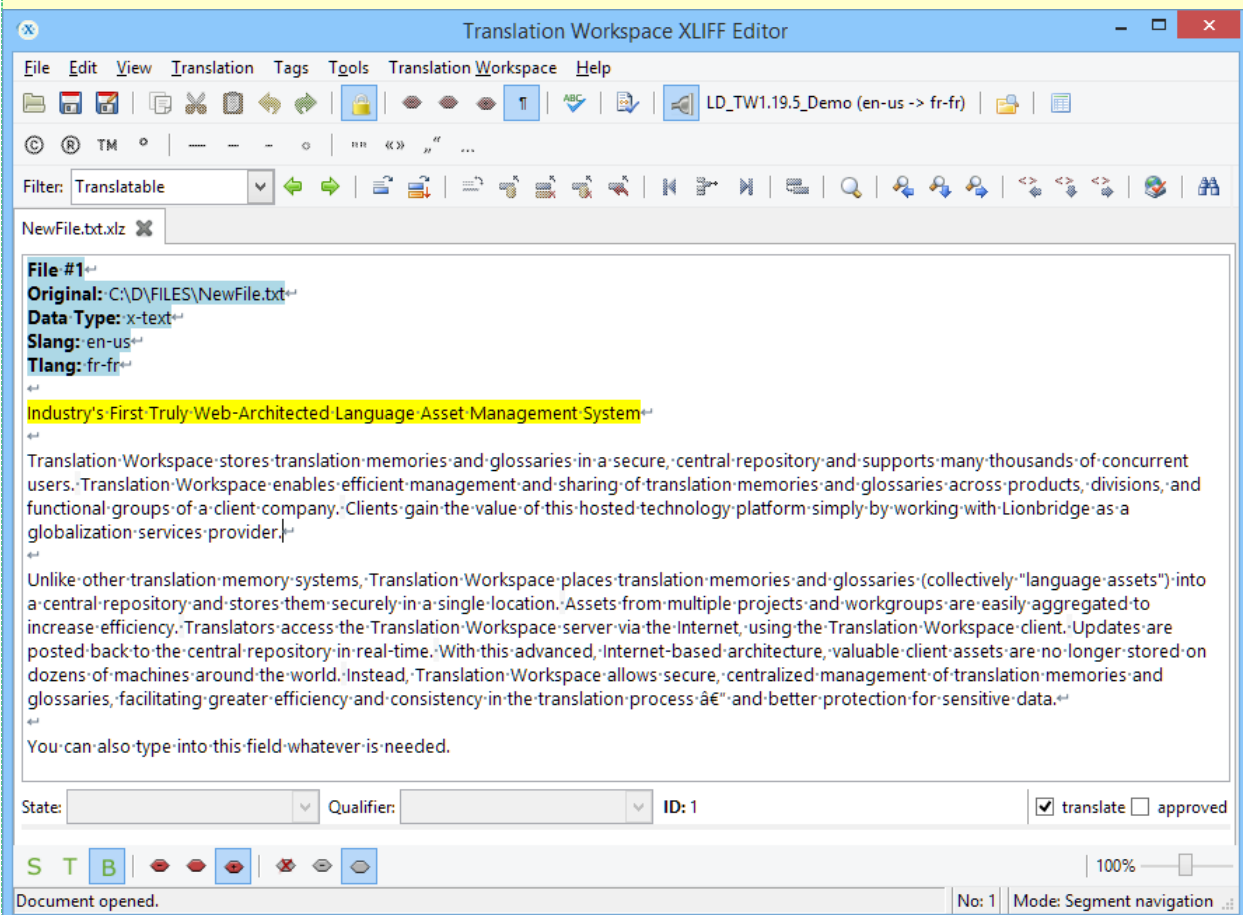
When you click **Save** on the **Save File** dialog, the TEXT Filter is used in background to convert the text added in the **Create New File** dialog into XLZ using the Plain Text profile of the `Default.lmx` file.

An XLZ file is then created into the selected path with the provided name.

The created file is opened in the XLIFF Editor and the **Set Target Language For** dialog is displayed.



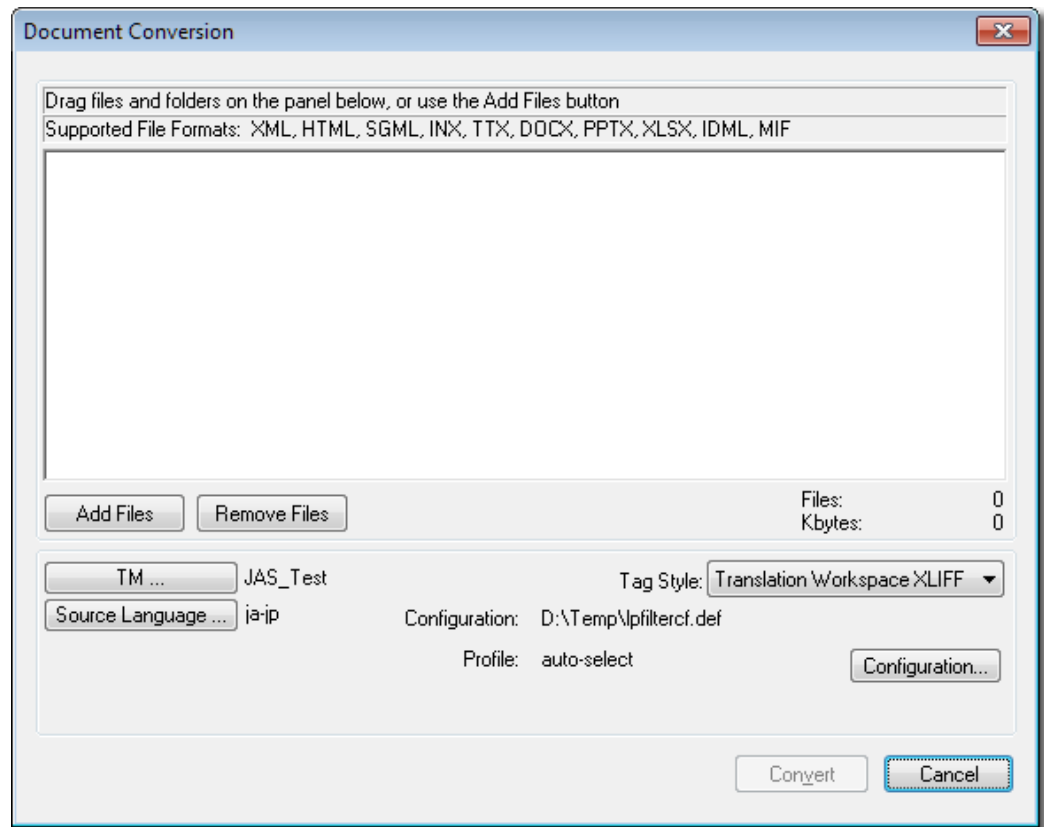
After you have selected a target language and clicked OK, the file is opened in XLIFF Editor. The file is now translatable in XLIFF Editor.





Microsoft Office Files (DOC, DOCX, RTF, PPT, PPTX, XLS, XLSX)

To get Microsoft Office files into XLIFF, RTF, or Trados, use the **Document Conversion** dialog, which has filters for these formats.



DOCX Filter

Microsoft Office 2007 contains a new format, Office Open XML, which is now widely used in the industry for Desktop Publishing (DTP). Office Open XML also includes word-processing (WordprocessingML, docx). The Translation Workspace DOCX filter supports the following native Word 2007-2013 formats: DOCX, DOCM, DOTM, and DOTX.

Defining text as Untranslatable

When some part of text of the source document should not be translated, you can modify the original DOCX file so this specific text will be treated in a specific way by the Translation Workspace DOCX Filter.

To mark text as untranslatable in DOCX files, open the file in Microsoft Word 2007-2013 and apply the specific Translation Workspace styles **LogoportMarkup** or **LogoportDoNotTranslate**. During the filtering, any text marked with these styles is recognized and the text is marked accordingly.

DOCX TagML

The tags for the TagML that you get when you apply the convert to Translation Workspace RTF or XLIFF format filter to a DOCX file is shown in the following table.



Tag name	Parent	Type	Explanation
TagML	(N/A)	External	Root element of tagged XML file.
part	TagML	External	Package part which has translatable contents. The name attribute contains path and file name inside the package.
p	document, comment, dataModel, endnote, ftr, hdr	External	The p (Paragraph) tag specifies a paragraph of content in the document, a footer, a header, footnotes, endnotes, or diagrams.
comment	comments	External	The comment (Comment Content) tag specifies the content of a single comment stored in the Comments part of a WordprocessingML document.
comments	part	External	The comments (Comments Collection) tag specifies all of the comments defined in the current document.
dataModel	part	External	The dataModel (Data Model) tag contains data for instance of a diagram.
document	part	External	This element specifies the contents of a main document part in a WordprocessingML document.
endnote	endnotes	External	The endnote (Endnote Content) tag specifies the content of a single endnote within a WordprocessingML document.
endnotes	part	External	The endnotes (Document Endnotes) tag specifies the set of all endnotes in the document, including endnote separators and continuation notices.
footnote	footnotes	External	The footnote (Footnote Content) tag specifies the content of a single footnote within a WordprocessingML document.
footnotes	part	External	The footnotes (Document Footnotes) tag specifies the set of all footnotes in the document, including footnote separators and continuation notices.
ftr	part	External	The ftr (Footer) tag specifies the content for a single footer for use within one or more sections of a WordprocessingML document.
hdr	part	External	The hdr (Header) tag specifies the content for a single header for use within one or more sections of a WordprocessingML document.
fieldproperties	document	External	Field property collection which holds all placeholder information of a nearby paragraph.
field	fieldproperties	External	The field tag holds additional information (translatable and non-translatable) extracted from an inline placeholder.
fieldcode	field	External	The fieldcode (Field Code) tag shows the raw field code name, format, and so on.
fielddisplay	field	External	The fielddisplay (Field display) tag shows non-translatable auto text.
fieldtext	field	External	The fieldtext (Field text) tag specifies translatable content.
sdt		External	The sdt (Inline-Level Structured Document Tag) tag specifies the presence of a structured document tag around one or more inline-level structures (runs, DrawingML objects, fields, and so on) in the current paragraph.



Tag name	Parent	Type	Explanation
alias	sdt	External	The alias (Friendly Name) tag specifies the friendly name associated with the current structured document tag.
sdtContent	sdt	External	The sdtContent (Cell-Level Structured Document Tag Content) tag specifies the last known contents of a structured document tag around a single table cell.
alt	pict, picture	External	The alt (Alternative Text description) tag of a picture specifies alternative text for the current DrawingML object, for use by assisting technologies or applications which will not display the current object.
table	document, td	External	Specifies a table structure. A table can be embedded in a table cell of another table.
tr	table	External	The tr (Table Row) tag contains a table row.
td	tr	External	The tc (Table Cell Data) tag contains a table cell.
textbox	pict, picture	External	The textbox (Rich Text Box Content Container) tag specifies that its contents shall be any rich WordprocessingML content, and that this content is the rich contents of a drawing object defined using the Vector Markup Language (VML) syntax.
textpath	pict, picture	External	The textpath (Text Layoutout Path) tag is used to define a vector path based on the text data, font and font styles supplied.
cf	field, fieldtext, hyperlink, p, sdtContent, smartTag	Internal	The cf (Characters format) tag is the most common and the most important tag, holding, in attributes, all formatting information, such as font name, size, color, and all kinds of styles, such as bold, italic, underline, and strikethrough.
symbol	cf, filed, p	Internal	The symbol tag specifies symbols which usually are not displayable in tagged format.
br	p, hyperlink, smartTag	External Internal	The br (Line Break) tag depends on a user option, which can be internal or external style, resulting in a segmentation difference.
bookmarkStart	P	Internal	A placeholder that indicates the start of a bookmark.
bookmarkEnd	P	Internal	A placeholder that indicates the end of a bookmark.
IndexMarker	P	Internal	A placeholder that indicates the existence of an index marker.
commentrangestart	P	Internal	A placeholder that indicates a comment range start.
commentrangeend	P	Internal	A placeholder that indicates a comment range end.
commentreference	P	Internal	A placeholder that indicates the existence of a Comment Content Reference Mark.
drawing	P	Internal	A placeholder that indicates an inline drawing object.
endnoteref	P	Internal	The endnoteref (Endnote Reference Mark) tag specifies the presence of an endnote reference mark in endnotes context.
endnotereference	P	Internal	An inline placeholder that specifies the presence of an endnote reference. An endnote reference is a run of automatically numbered text which references a



Tag name	Parent	Type	Explanation
			particular endnote within the parent document, and inherits the endnote reference mark's numbering.
equationreference	P	Internal	A placeholder that specifies the presence of an equation.
field	fieldproperties, p, sdtContent	Internal	A placeholder that specifies the presence of a field, where more information is extracted and placed after this paragraph in the fieldproperties section.
footnoteref	P	Internal	The footnoteref (Footnote Reference Mark) tag specifies the presence of a footnote reference mark in footnotes context.
foornotereference	P	Internal	The footnoteReference (Footnote Reference) tag is an inline placeholder which specifies the presence of a footnote reference. A footnote reference is a run of automatically numbered text which references a particular footnote within the parent document, and inherits the footnote reference mark's numbering.
hyperlink	P	Internal	The hyperlink (Hyperlink) tag specifies the presence of a hyperlink at the current location in the document. The display text is translatable inline, while its tooltip and URL address are extracted after this paragraph, which are also translatable contents.
oMathPara	P	Internal	The oMathPara (Math Paragraph) tag specifies a math paragraph, one or more display equations within a single paragraph.
oleobject	P	Internal	A placeholder that specifies the presence of an oleobject (Embedded OLE Object).
pict	P	Internal	The pict tag specifies a placeholder.
picture	p	Internal	The picture tag indicates that an object specified in VML is located at the current position in the run (for example, a floating embedded control). It may contain translatable content.
ptab	p	Internal	The ptab (Absolute Position Tab Character) tag specifies that an absolute position tab character shall be placed at the current location.
continuationSeparator	p	Internal	The continuationSeparator (Continuation Separator Mark) tag specifies the presence of a continuation separator mark within the current run. A continuation separator mark is a horizontal line which spans the width of the main story's text extents.
separator	p	Internal	The separator (Footnote/Endnote Separator Mark) tag specifies the presence of a separator mark within the current run. A separator mark is a horizontal line which spans part of the width text extents.
smartTag	p, smartTag	Internal	The smartTag (Inline-Level Smart Tag) tag specifies the presence of a smart tag around one or more inline structures (runs, images, fields, and so on) within a paragraph.



Tag name	Parent	Type	Explanation
wordart	p	External	The wordart (Word Art) tag is an inline placeholder, where its text path is a translatable content, and is extracted after this paragraph.

Note:

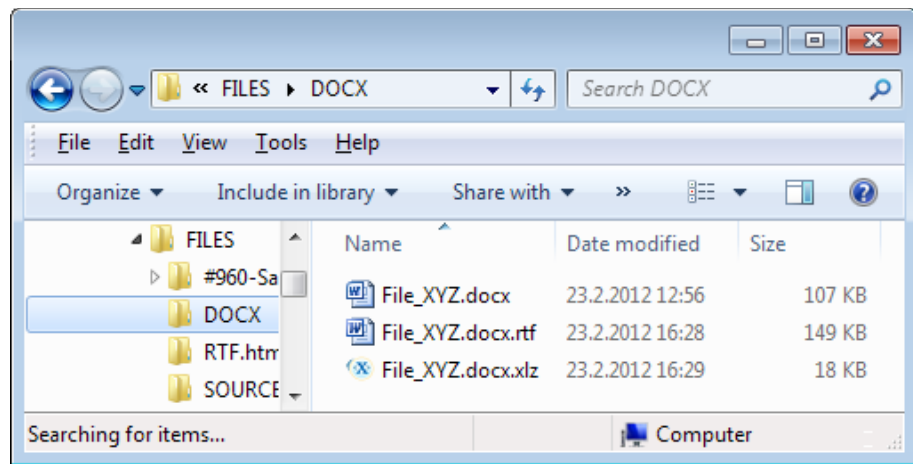
1. In general, all external tags are not editable.
2. One exception is the **<part>** tag, which can be deleted, moved, or reordered. A part can be the main story document, a footer or a header, a footer notes or an endnotes. Deleting one part means this part is skipped from localization process.
3. Internal tags are editable. A common editable tag is the **<cf>** tag, where its attributes are editable, removable, movable, or reordered. As a tag itself, a **<cf>** tag can be inserted in pair, editable, removable, movable, or reordered. Refereeing to text content, internal tags are movable. Refereeing to other internal tags, usually, they are not reordered.

Converting DOCX to RTF or XLIFF

To convert a DOCX file, drag and drop the DOCX files to the **Document Conversion** dialog field and select your configuration file (.ltx) for the **Configure Filters** dialog.

You can choose the conversion to RTF or XLIFF from the Tag Style drop down list. A default DOCX configuration is used for the conversion, even if a configuration file has been selected.

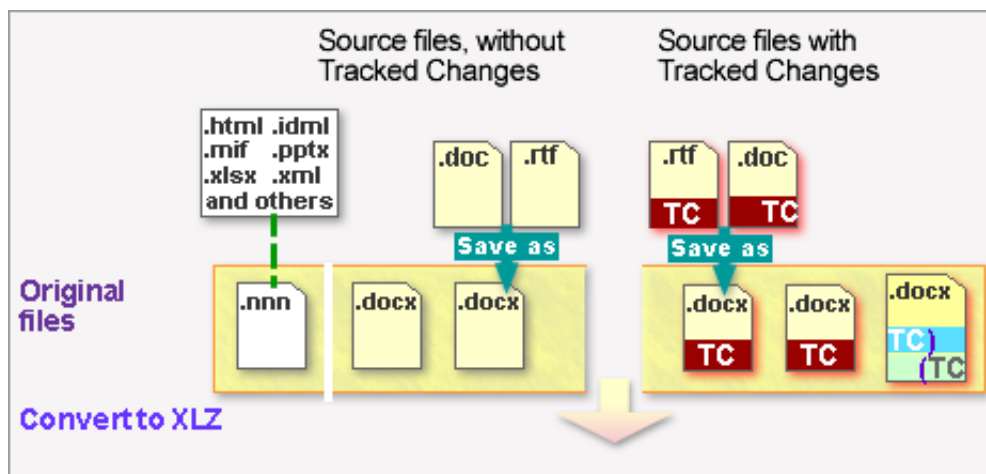
The converted file is stored in the same folder as the original DOCX file.



DOCX Files with Tracked Changes

DOCX files with track changes can be translated in Translation Workspace.

Original RTF files need to be saved to DOCX. You should save any original RTF files with or without track changes to DOCX format with MS Word before you start processing the files. Direct conversion of RTF files to XLZ accepts all tracked changes in the files and disables tracking during further processing of such files.



Files converted to RTF are not supported. You cannot process RTF files with track changes process without losing the tracking. Files have to be converted to XLZ with the DOCX filter from DOCX to enable the tracking and retain the tracked changes during processing.

Original DOCX and DOC files can include tracked changes that are retained during further processing when converted to XLZ. The DOC files have to be saved to DOCX files before converting to XLZ with the DOCX Filter.

Original DOC and DOCX files can also be already (fully or partially) segmented and they can include tracked changes in both, the source and the target segments.

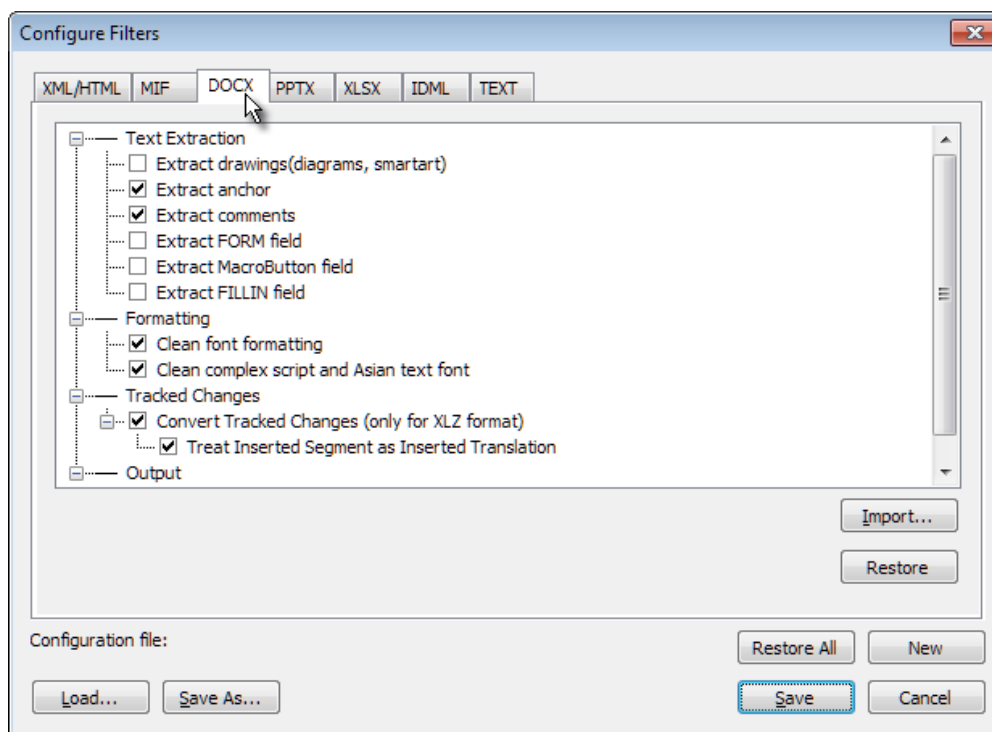
All other original file types (PPTX, XLSX, IDML, MIF...), (.nnn) are converted with the appropriate filter for those files. Tracked changes can be included first when the XLZ file is edited in XLIFF Editor with Tracked Changes ON, but all changes are accepted and changes are not tracked in the final document generated from the XLZ.

Regardless of the original file type, new changes created in XLIFF Editor can be tracked in the target segments in all XLZ files.

Using the **Convert Tracked Changes (only for XLZ format)** option in the DOCX Filter supports tracked changes while converting DOCX files to XLZ. This option is disabled by default.

Enabling this option displays and **enables** the **Treat Inserted Segment as Inserted Translation** sub-option. When this sub-option is enabled, the convert process inserts an empty target segment and target segment revision data to the XLZ for source segments consisting only of a tracked insertion. This helps the translator to mimic the tracked changes of the source segment in the target segment. In XLIFF Editor such segments have an orange highlight (see below).

```
style="Normal" font="Corbel" asiantextfont="Corbel" complexscriptfont
(<:Lorem ipsum dolor sit amet, consectetur adipiscing elit.)=0%(:)</p>
</textbox>1
```



Warning: Using the **Convert Tracked Changes (only for XLZ format)** option to convert files to **RTF** *does not support tracked changes* (all tracked changes are automatically accepted in the files before the conversion).

We recommend enabling two additional options for reducing the number of tags in the generated XLZ files:

Clean font formatting and

Clean complex script and Asian text font.

You can of course convert files with other than DOCX (DOC or RTF) as the original format with the other filters. At this phase those files cannot include track changes.

Note: You can convert DOCX files with tracked changes and DOCX files without tracked changes to XLZ in the same run. You must *enable* the **Convert Tracked Changes (only for XLZ format)** option on the DOCX tab of the **Configure Filters** dialog to retain the tracked changes in the files.

Note: Tracked formatting changes are in all cases accepted during conversion.

Pretranslation options

The Translate Documents dialog has two **Track Changes (XLZ only)** options for handling the tracked changes in the documents to be pretranslated. Any of these options can be selected when XLZ files with tracked changes have been added to the dialog's files panel.

Accepted bases the pretranslation on the final segment (*as if* all changes had been accepted).

Rejected (default option) bases the pretranslation on the original segment (*as if* all changes had been rejected). Segments pretranslated while this option is selected are marked as **0%** matches in the files and reported by the TW Tools analysis as "untranslated". In XLIFF Editor such segments have an orange highlight (see the screen capture below).

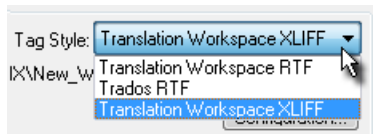


```
<p style="pBodytxt" font="Segoe UI" asiantextfont="Segoe UI" complexscriptsfont="Segoe UI" size="9" c  
(-:This filesegment includes tracked changes.)=0%(Ce fichier comprend le suivi des modifications.:) </p>  
<p style="pBodytxt" font="Segoe UI" asiantextfont="Segoe UI" complexscriptsfont="Segoe UI" size="9" c
```

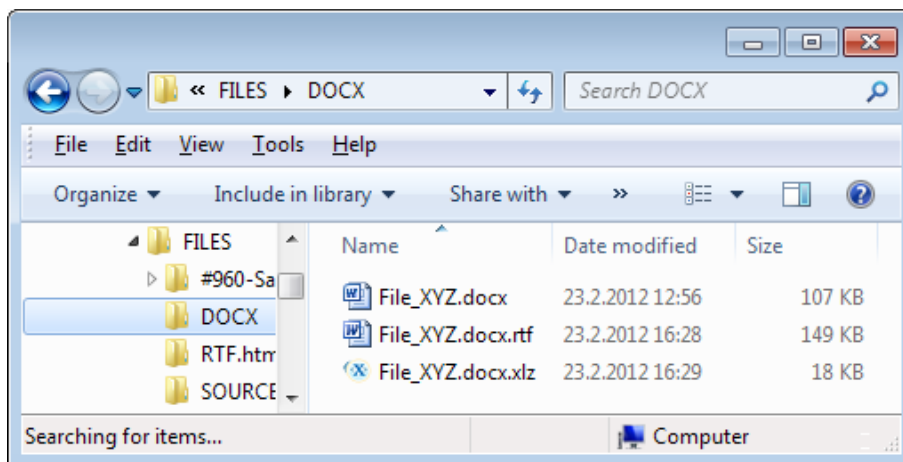
Converting DOCX to RTF or XLIFF

To convert a DOCX file, select **Documents > Convert** and drag and drop the DOCX files to the Document Conversion dialog field file box (or click **Add Files** and browse to the file).

You can select conversion to XLIFF (default) or RTF from the **Tag Style** drop down list. You can select a configuration file (.lmx) with desired options. If no configuration file is specified, the default settings file that is stored on the server is automatically used.



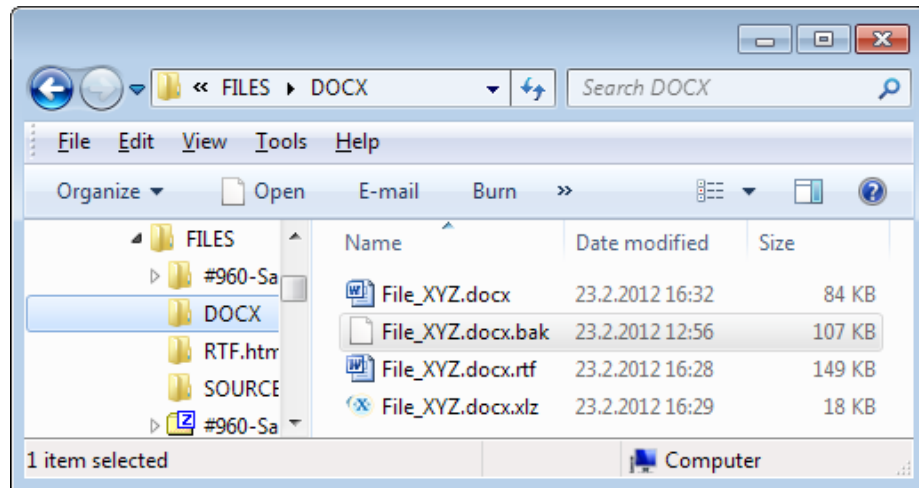
The converted file is stored in the same folder as the original DOCX file.



Converting back to DOCX

You can use the **Back Convert** function to generate a translated version of a DOCX file. When translating bidirectional languages, you must select the correct target bidirectional language in the back conversion window.

The original DOCX file should be in the same folder as the file to be back converted. Make sure *not* to modify the original file extension of the translated file, because the filter checks it to recognize the file as converted from a DOCX file.



Translation Workspace creates a new DOCX file and renames the source DOCX file with a .bak extension.

DOCX files are always UTF-8 XML files, and they are therefore not affected by the selected **Codepage** or the **Character to entities** setting.

Files with Tracked Changes

When generating a translated document using the Back Convert function (TW Tools, Xliff Editor) or **File > Save Target** in the Xliff Editor from a XLZ file containing tracked changes:

If the original source document is a Microsoft Word 2007-2010 document (.docx), then tracked changes in target segments are preserved and appear in the final DOCX file.

For all other original file types (PPTX, XLSX, IDML, MIF...), target segments tracked changes are automatically accepted in the translated XLZ file and then the final document generated.

If a segment does not contain a target (translation), the source segment with all the tracked changes is retained in the back converted file.

Back conversion of XLZ files with tracked changes to DOCX format retains the tracked changes existing in the XLZ files in the resulting DOCX files when **Convert Tracked Changes (only for XLZ format)** is enabled in the DOCX Filter.

If a segment does not contain a target (translation), the source segment with all the tracked changes is retained in the back converted file.

Known Limitations:

Back conversion attempts to add track changes to fields such as comments and alternate text that are not supported by Word. In case of comments, the changes are joined together and in case of alternate text, the translation is not shown.

Track changes to field codes are accepted during back conversion.

Back conversion changes deletions to insertions when the content is inside a deleted text box.

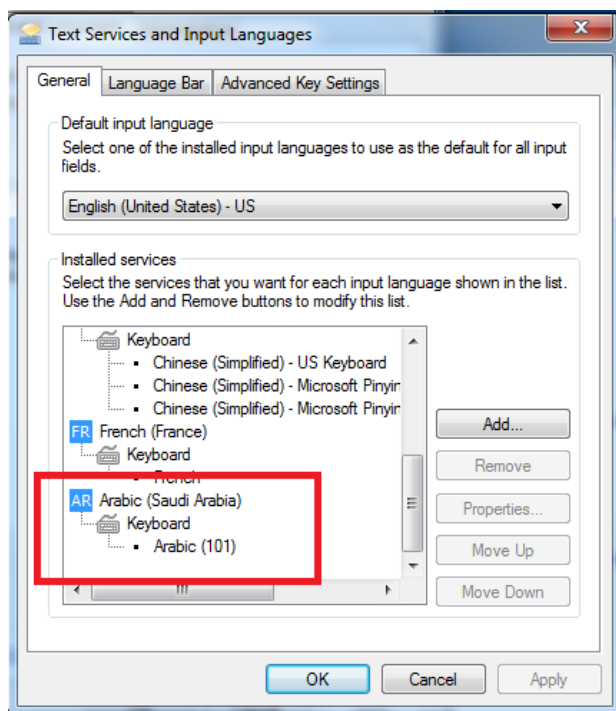
Track changes in index entries are not supported during forward and back conversion. Changes are accepted.

Track changes on paragraph marks are not accepted during back conversion, when the option **Convert Track Changes** is OFF.



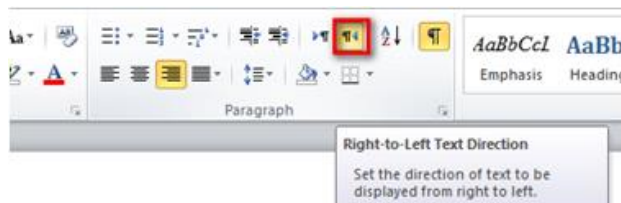
Right-to-left rendering on the DOCX file

In order to have the correct rendering of bidirectional languages on Microsoft Word, you need to have the following:



Arabic, Farsi, Hebrew, and Urdu languages defined under the *Choose Editing Languages* section of the Microsoft Office Language Preferences tool, or

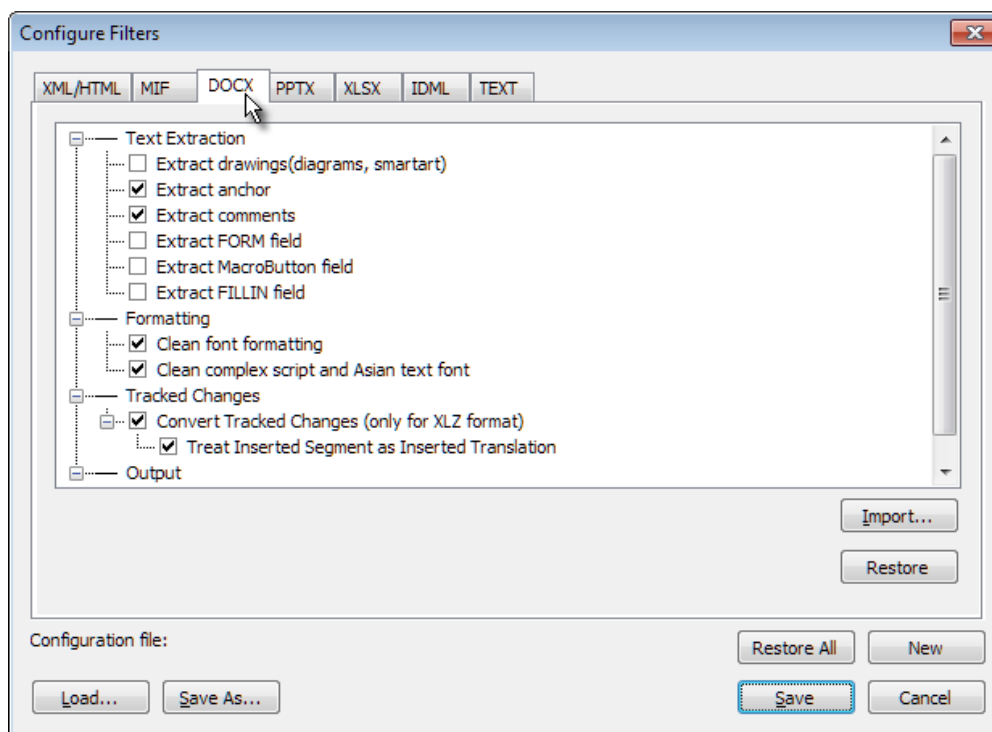
- An Arabic, Farsi, Hebrew or Urdu keyboard defined on the machine, or
- Regional settings set to Arabic, Farsi, Hebrew or Urdu on the machine
- To generate a DOCX file with a correct right-to-left alignment, do the following:
 1. Switch the regional settings of the machine to Arabic, Farsi, Hebrew or Urdu, or just add/set an Arabic, Farsi, Hebrew or Urdu keyboard:



2. Open the back converted DOCX file, select all the text, and click on the Right-to-Left Text Direction button, available on the Home tab of the Word ribbon.

Additional DOCX filter information

To configure the DOCX filter, go to the **Document Conversion** dialog and click **Configuration...** to get to the **Configure Filters** dialog. Then select the DOCX tab.



What is extracted for translation?

By checking options in the **Text Extraction** option list you can select some additional objects to be extracted for translation are extracted:

Option	Description
Extract drawings (diagrams, Smart Art)	When enabled , DrawingML objects like diagrams or Smart Art are extracted for translation.
Extract anchor	When enabled , anchors are extracted for translation.
Extract comments	When enabled , comments are extracted for translation.
Extract FORM field	When enabled , text from the FORM field is extracted for translation. The FORM field code prompts you to enter text. Your response is printed in place of the field.
Extract MacroButton field	When enabled , text from the MacroButton field is extracted for translation.
Extract FILLIN field	When enabled , text from the FILLIN field is extracted for translation. The FILLIN field code prompts you to enter text. Your response is printed in place of the field.

Formatting objects extracted for translation

By checking options in the **Formatting** option list you can select some formatting to be extracted for translation are extracted:

Option	Description
Clean font formatting	When enabled , tags corresponding to extra font formatting information are not extracted fro translation: Spacing information



	▪ Scale information ▪ Kerning information ▪ Position information ▪ Complex case with different formatting.
Clean complex script and Asian text font	When enabled, complex script tags and Asian font tags are not extracted for translation. This option should not be selected if complex script or Asian font is in use in the document. Information not extracted during forward conversion is also not included in the back converted document..

Objects extracted for translation

Text run	Picture alternative text
Text Box	Comments
Footnotes and Endnotes	Wordart text
Header and Footer	SDT text
Index entries	FORM Fields text (optional)
Hyperlink	MacroButton Fields text (optional)
Hyperlink URL	FILLIN Fields text (optional)
Hyperlink Alternate text	Smartart text (optional)
Chart text	Diagram text (optional)

Objects not extracted for translation

Citation	More generically All field code (except Index entries)
Equation	Embedded object (docx, pptx, xlsx,...)

Note: TOCs and Indexes are *not* included. These can be re-generated after the body text has been localized.

Output options

To enable the back conversion to run without the source file being available separately, set the conversion to embed the source DOCX file within the XLZ file:

Option	Description
Include the Source File (only for XLZ format)	When converting to XLZ this option disabled, the conversion creates an XLZ file without embedding the DOCX source file within it. When converting to XLZ with this option enabled, the DOCX source file is embedded within the XLZ output file..

Previewing XLZ documents converted with DOCX Filter

To run preview from XLIFF Editor for the XLZ files created from source MS Word files in MS Word, and from TW Online Review Client rendered as HTML in your browser by an open office library, enable the *Include the Source File (only for XLZ format)* option in the DOCX Filter when converting the original file to XLZ.

When the **Use Associated Application** option is checked in the **Preview Settings** dialog of the XLIFF Editor and you request a preview:



- The XLZ files is back converted using the embedded source document
- The generated final document is opened in MS Word.

An error message is displayed (or returned),

- If any problem occurs during the back conversion
- If the file cannot be opened in the associated application
- If back conversion fails

You can generate a new preview when a previous preview file is still open.

Paragraph and Line Breaks Handling

Line breaks (soft return) are converted to `<w:br>` tags and therefore not considered as end of segment markers.

Paragraph breaks are converted to `<p>` tags and therefore considered as end of segment markers. This causes dividing to segments at all Paragraph breaks.

Special Characters and Symbols

There are some symbol characters that are treated in a specific way.

Symbols characters are identified, as they are represented with few special Microsoft Word Fonts. These fonts are identified as using to represent symbol:

Webdings	Greek symbols
Wingdings	MT Extra
Wingdings 2	Marlett
Wingdings 3	VisualUI

These characters are not output as Unicode characters as there is no representation for them. Instead, we output them as placeholders, for example:

```
<symbol character="00f063"/><symbol character="00f064"/>
```

The character code corresponds to the RTF character code. The back conversion replaces the placeholder with the corresponding symbol.

PPTX Filter

The Translation Workspace PPTX filter supports following native PowerPoint 2007 formats: .pptx, .pptm, .potx, .potm, .ppsx, and .ppsm.

Defining Text as Untranslatable

When some part of a slide of the source document should not be translated, you can modify the original PPTX file so that this specific text will be treated in a specific way by the Translation Workspace PPTX Filter.

To mark text as *not* to be presented for translation in PPTX files, open the file in Microsoft PowerPoint (2007-2013) and apply a **strikethrough** formatting.

During the filtering, if the **Extract strikethrough text** option is not selected, then strikethrough text will be extracted in `LogoportDoNotTranslate` format.

PPTX TagML

The tags for the TagML that you get when you apply the convert to Translation Workspace RTF or XLIFF format filter to a PPTX file is shown in the following table.



Tag name	Parent	Type	Explanation
TagML	(N/A)	External	Root element of tagged XML file.
part	TagML	External	Package part which has translatable contents. The name attribute contains path and file name inside the package.
p	cmLst, dataModel	External	The p (Paragraph) tag specifies a paragraph of content in comments and, or, diagrams.
cmLst	part	External	The cmLst (Comment list) tag specifies all of the comments defined in the current document.
dataModel	part	External	The dataModel (Data Model) tag specifies data for instance of a diagram.
notes	part	External	The notes (Notes Slide) tag specifies the existence of a notes slide along with its corresponding data.
slide	part	External	The slide (Presentation Slide) tag specifies a slide within a slide list. The slide list is used to specify an ordering of slides.
fieldproperties	group, notes, slide, td	External	Field property collection which holds all placeholder information of a nearby group, notes, slide, or a table cell.
field	textbox	External	A placeholder for the presence of an inline auto-text field, for example, a slide number.
fieldtext	hyperlink	External	The fieldtext (Field text) tag specifies a translatable content of a hyperlink address in field property section.
alt	picture	External	The alt (Alternative Text description) tag of a picture specifies alternative text for the current DrawingML object, for use by assisting technologies or applications which will not display the current object.
table	slide	External	Specifies a table structure in a slide. PowerPoint tables are not embedded each other, which is simpler than table in Word.
tr	table	External	The tr (Table Row) tag specifies a table row.
td	tr	External	The tc (Table Cell Data) tag specifies a table cell.
textbox	group, notes, slide, td	External	The textbox tag is an important tag in PowerPoint, holding translatable contents in a group, a notes, a slide, or a table cell.
cf	p, textbox	Internal	The cf (Characters format) tag is the most common and the most important tag, holding, in attributes, all formatting information, such as font name, size, color, and all kinds of styles, such as bold, italic, underline, and strikethrough.
symbol	cf		The symbol tag specifies symbols which usually are not displayable in tagged format.
br	textbox	External Internal	The br (Line Break) tag depends on a user option, which can be internal or external style, resulting in a segmentation difference.
sr	cf	Internal	The sr (Soft Break) tag indicates a soft break.
cxnSp	group, slide	External	The cxnSp (Connection Shape) tag is a placeholder that specifies a connection shape that is used to connect two <sp> elements.
graphicFrame	group, slide	External	The graphicFrame (Graphic Frame) tag specifies the existence of a graphics frame. This frame contains a graphic that was generated by an external source and needs a container in which to be displayed on the slide surface.



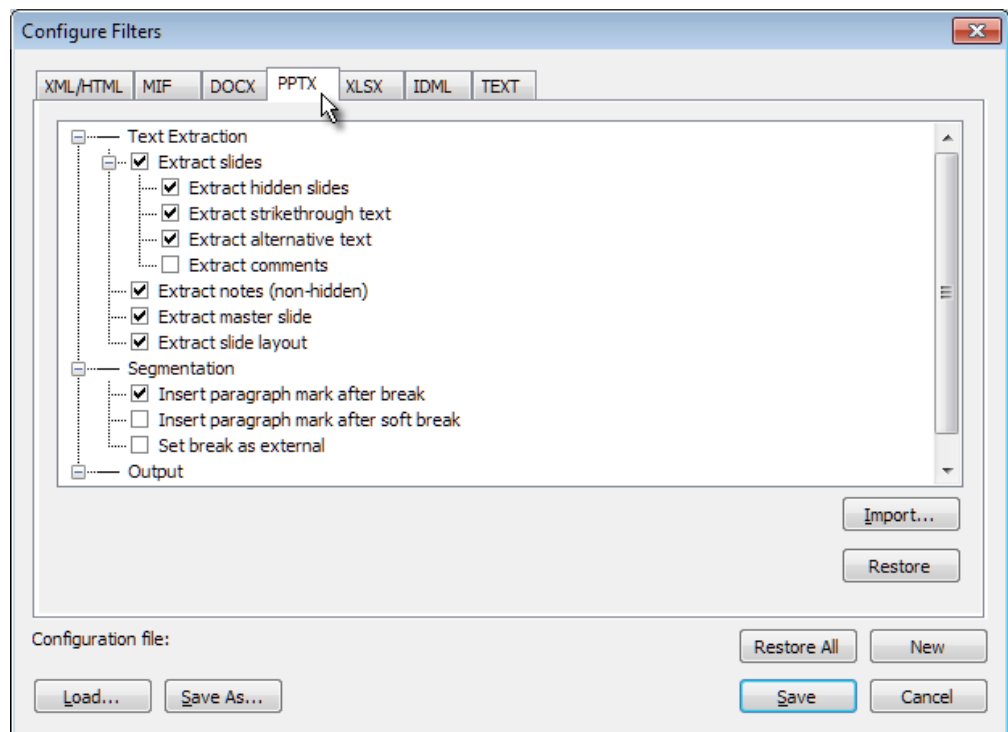
Tag name	Parent	Type	Explanation
group	group, slide	External	The group(Group Shape) tag specifies a group shape that represents many shapes grouped together. This shape is to be treated just as if it were a regular shape but instead of being described by a single geometry it is made up of all the shape geometries encompassed within it. Within a group shape each of the shapes that make up the group are specified just as they normally would.
oleobject	group, slide	Internal	A placeholder specifying the presence of an oleobject (Embedded OLE Object).
picture	group, slide	Internal	The picture tag indicates that an object specified in VML is located at the current position in the run (for example, a floating embedded control). It may contain translatable content.
hyperlink	fieldproperties, notes, p, slide, td, textbox	External Internal	The hyperlink (Hyperlink) tag specifies the presence of a hyperlink at the current location. As an inline placeholder it is internal, but in the property section it is external and its tooltip and URL address are extracted after the paragraph and are also translatable contents.

Note:

1. In general, all external tags are not editable.
2. One exception is the <part> tag, which can be deleted, moved, or reordered.
3. Internal tags within a textbox are editable, removable, movable, and can be reordered.

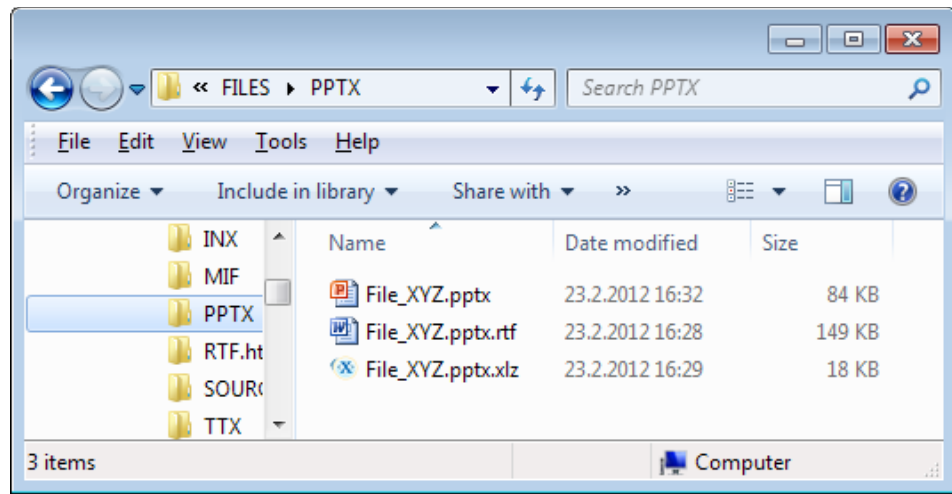
Converting PPTX to RTF or XLIFF

To convert a PPTX file, drag and drop the PPTX files to the **Document Conversion** dialog field, and select your configuration file (.ltx) for the **Configure Filters** dialog.





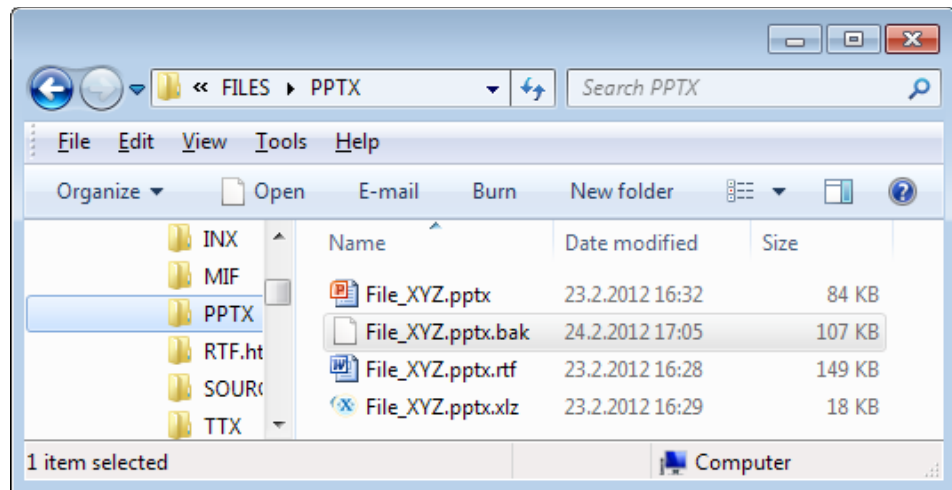
You can choose the conversion to RTF or XLIFF from the **Tag Style** drop down list.



The converted file is stored in the same folder as the original PPTX file.

Converting back to PPTX

You can use the **Back Convert** function to generate a translated version of a PPTX file.



The original PPTX file should be in the same folder as the file to be back converted. Make sure *not* to modify the original file extension of the translated file, because the filter checks it to recognize the file as converted from a PPTX file.

Translation Workspace creates a new PPTX file and renames the source PPTX file with a `.bak` extension.

PPTX files are always UTF-8 XML files and they are therefore not affected by the selected **Codepage** or the **Character to entities** setting.

Additional PPTX filter information

To configure the PPTX filter, select **Documents > Convert > Configuration...** and click the **PPTX** tab.

Text Extraction

The **Text Extraction** option lets you to select following options when extracting for translation:



Option	Description
Extract slides	When enabled, the first three slides options are checked.
Extract hidden slides	When enabled, hidden slides are extracted for translation.
Extract strikethrough text	When not enabled, strikethrough text is extracted in <code>LogoportDoNotTranslate</code> style.
Extract alternative text	When enabled, alternative text for graphics is extracted for translation.
Extract comments	When enabled, comments are extracted for translation.
Extract notes	When enabled, speaker notes are extracted for translation.
Extract master slide	When enabled, text from the master slide is extracted for translation.
Extract slide layout	When enabled, text from the slide layouts is extracted for translation.

Segmentation

You can control the segmentation with following options.

Option	Description
Insert paragraph mark after break	When enabled, break tag (<code>
</code>) is considered as end of segment markers.
Insert paragraph mark after soft break	When enabled, soft break (<code><sr></code>) is considered as end of segment markers.
Set break as external	When enabled, break tag (<code>
</code>) is set in <code>LogoportDoNotTranslate</code> style, so is considered as end of segment markers.

Objects extracted for translation

The following objects are extracted for translation:

Master slides	Notes
Slides layout	Headers and Footers
Slides	Smart Art
Charts	

Objects not extracted for translation

The following objects are not extracted for translation:

Embedded object (for example, docx, pptx, xlsx)

Output options

To enable the back conversion to run without the source file being available separately, set the conversion to embed the source DOCX file within the XLZ file:

Option	Description
--------	-------------

**Include the Source File
(only for XLZ format)**

When converting to XLZ this option disabled, the conversion creates an XLZ file without embedding the PPTX source file within it.

When converting to XLZ with this option enabled, the PPTX source file is embedded within the XLZ output file.

Previewing XLZ documents converted with PPTX Filter

To run preview from XLIFF Editor for the XLZ files created from source MS PowerPoint files in MS PowerPoint, and from TW Online Review Client rendered as HTML in your browser by an open office library, enable the *Include the Source File (only for XLZ format)* option in the PPTX Filter when converting the original file to XLZ.

When the **Use Associated Application** option is checked in the **Preview Settings** dialog of the XLIFF Editor and you request a preview:

- The XLZ files is back converted using the embedded source document
- The generated final document is opened in MS PowerPoint.

An error message is displayed (or returned),

- If any problem occurs during the back conversion
- If the file cannot be opened in the associated application
- If back conversion fails

You can generate a new preview when a previous preview file is still open.

Known Limitations

Support for Bidirectional text (Arabic, Hebrew) is limited.

Currently translation is supported, but not Right to Left rendering during back conversion to PPTX.

XLSX Filter

The Translation Workspace XLSX filter supports following native Microsoft Excel formats: .xlsx, .xlam, .xlsm, .xltm, and .xltx.

Defining Text as Untranslatable

When some part of the source Excel document should not be translated, you can modify the original XLSX file so that this specific text will be treated in a specific way by the Translation Workspace XLSX Filter. You can do the following:

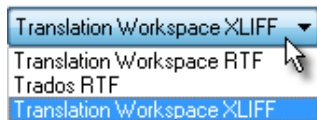
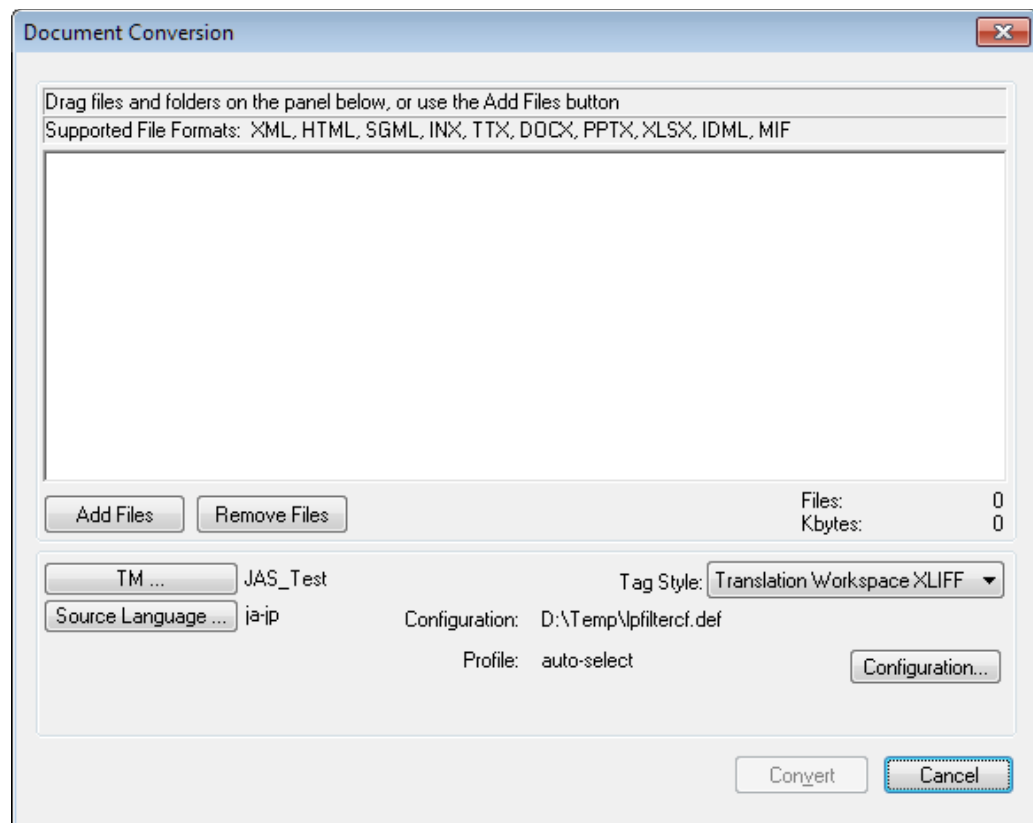
1. Open the source file.
2. Hide rows, columns, or sheets that contain non-translatable text.
3. Save the file.
4. Use Translation Workspace Tools, with "Translate hidden rows/columns" and/or "Extract hidden sheets" configuration options disabled.

Alternatively, you may specify a range of cells that contain non-translatable text, in the "Non-translatable cells ranges:" text field. For example, typing in "Sheet 1!A6:B7" will mark cells **A6**, **A7**, **B6**, and **B7** on **Sheet1** as non-translatable.



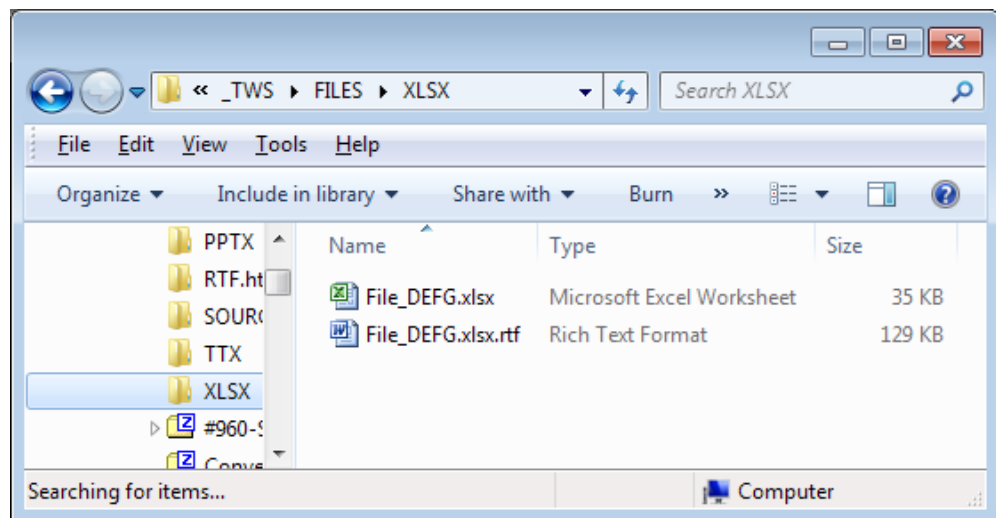
Converting XLSX to RTF or XLIFF

To convert an XLSX file, select **Documents > Convert** and drag and drop the XLSX files to the **Document Conversion** dialog field (or click **Add Files** and browse to the files).



You can select conversion to RTF or XLIFF from the **Tag Style** drop down list. You can select a configuration file (.lrx) with desired options. If no configuration file is specified, the default settings file that is stored on the server is automatically used.

The converted file is stored in the same folder as the original XLSX file.

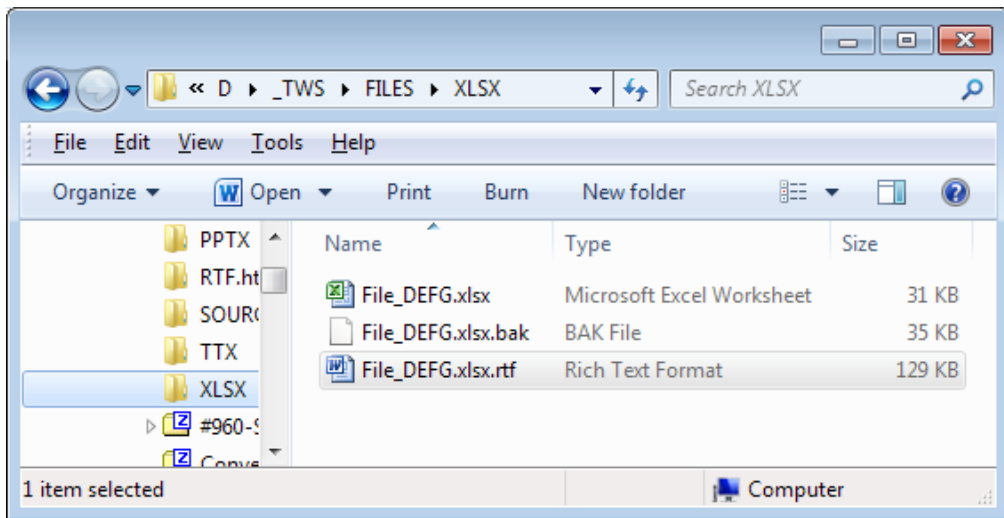




Converting Back to XLSX

You can use the **Back Convert** function to generate a translated version of a XLSX file.

The original XLSX file should be in the same folder as the file to be back converted. Make sure not to modify the original file extension of the translated file, because the filter checks it to recognize the file as converted from a XLSX file.



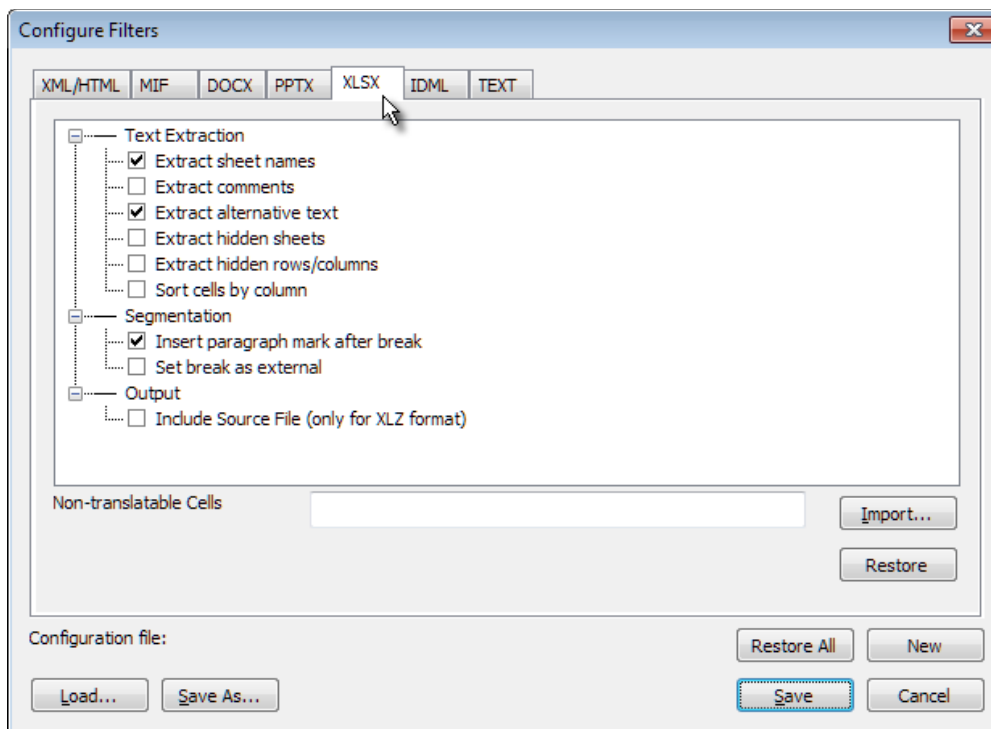
Translation Workspace creates a new XLSX file and renames the source XLSX file with a .bak extension.

XLSX files are always UTF-8 XML files and they are therefore not affected by the selected Codepage or the Character to Entities setting.



Additional XLSX Filter Information

To configure the XLSX filter, select **Documents > Convert > Configuration...** and click the XLSX tab.



Text Extraction

The Text Extraction option lets you to select following options when extracting for translation:

Option	Description
Extract sheet names	When enabled, worksheet names are extracted for translation.
Extract comments	When enabled, comments are extracted for translation.
Translate hidden rows/columns	When enabled, text from hidden rows and columns is extracted for translation.
Sort cells by column	When enabled, text from cells is extracted from the top down and then from left to right. When disabled, text from cells is extracted from left to right and then from the top down.
Extract alternate text	When enabled, alternate text for graphics (for example, shape, picture, chart, and SmartArt graphic) is extracted for translation.
Extract hidden sheets	When enabled, hidden text on hidden worksheets is extracted for translation.



Segmentation

You can control the segmentation with following options.

Option	Description
Insert paragraph mark after line break	When enabled, the break tags (and <sr>) are considered as end of segment markers.
Set line break as external	When enabled, the break tag () is set in LogoportDoNotTranslate style, and is considered as end of segment markers.

Specifying an Excluded Range

Alternatively, you may specify a range of cells that contain non-translatable text, in the **Non-translatable cells ranges** text field. The "non-translatable" cells will be present in the tagged format but will be in external style, allowing the preservation of the full context.

Following are examples of using this feature:

The range to be excluded	Input to the text field
One cell, say, B4 (location at Column B, Row 4)	B4
One row, for example, the second entire row	Row 2
One column, for example, Column C	Column C

To exclude more specific cells, you list them and must use ',',';' or ':' to separate them.

The range to be excluded	Input to the text field
Two cells, say B4 and D7	B4, D7
Several columns and several rows	Row 2, Row 5; Col 3, Col 5
A range of cells, say, cells starting from B2 and ending at D7. (It is acceptable to put the right bottom cell identifier first; the result is same.)	B2:D7 [or, alternatively] D7:B2
You can specify a range by an "off-diagonal" manner. For example, you could specify (left, bottom) to (right, top), or any combination among left, right, top and bottom.	A10:E2 [or, alternatively] E2:A10 [or, alternatively] A2:E10 [or, alternatively] E10:A2
Without specifying a worksheet number, all cells/rows/columns apply to the first sheet.	A1:C34, Row 20, D:3
To exclude a cell on the second sheet, you must input ranges and use '!' as a sheet separator.	Sheet 2!B3:D7
One cell on the first sheet, two cells on the third sheet.	Sheet 1!A1; Sheet 3!B2, D7
One row on the second sheet and two columns on the third sheet (all letters and words are case-insensitive).	Sheet 2!row 3; Sheet 3!col B, col F
The list is order-free: an entire sheet list can be swapped, and within one sheet, the list of cells, list of rows, and columns are order-free.	Sheet 3!col F, col B; Sheet 2!row 3

Objects Extracted for Translation

The following objects are extracted for translation:



Cell text	Sheet names (optional)
Formula text	Comments (optional)
Table text	Hidden rows/columns (optional)
Headers and Footers (optional)	Alternative text of graphics (optional)
Hyperlinks	Hidden sheets (optional)
SmartArt text	Chart text

Objects Not Extracted for Translation

The following objects are not extracted for translation:

Embedded Objects (DOC, XLS, PPT, and so on)

Output options

Option	Description
Include the Source File (only for XLZ format)	When converting to XLZ this option disabled, the conversion creates an XLZ file without embedding the XLSX source file within it. When converting to XLZ with this option enabled, the XLSX source file is embedded within the XLZ output file..

Previewing XLZ documents converted with XLSX Filter

To run preview from XLIFF Editor for the XLZ files created from source MS Excel files in MS Excel, and from TW Online Review Client rendered as HTML in your browser by an open office library, enable the *Include the Source File (only for XLZ format)* option in the XLSX Filter when converting the original file to XLZ.

When the **Use Associated Application** option is checked in the **Preview Settings** dialog of the XLIFF Editor and you request a preview:

- The XLZ files is back converted using the embedded source document
- The generated final document is opened in MS Excel.

An error message is displayed (or returned),

- If any problem occurs during the back conversion
- If the file cannot be opened in the associated application
- If back conversion fails

You can generate a new preview when a previous preview file is still open.

XLSX TagML

The tags for the TagML that you get when you apply the convert to Translation Workspace RTF or XLIFF format filter to a XLSX file is shown in the following table.

Tag name	Parent	Type	Explanation
TagML	(N/A)	External	Root element of tagged XML file.
part	TagML	External	Package part which has translatable contents. The name attribute contains the path and file name inside the package (for example, <part name="xl/worksheets/sheet1.xml">).
name	part	External	Sheet tab name, which is an optional extracted translatable content.
p	dataModel	External	The p (Paragraph) tag specifies a paragraph of content in diagrams.



Tag name	Parent	Type	Explanation
comment	comments	External	The comment (Comment Content) tag specifies the content of a single comment stored in the Comments part of a WordprocessingML document.
comments	part	External	The comments (Comments Collection) tag specifies all of the comments defined in the current document.
dataModel	part	External	The dataModel (Data Model) tag contains data for an instance of a diagram.
links	part	External	The links tag specifies a container for specific types of external links.
header	worksheet	External	The header tag specifies presence of a header or footer, where its attribute distinguishes its type. There are six types: evenFooter (Even Page Footer), evenHeader (Even Page Header), firstFooter (First Page Footer), firstHeader (First Page Header), oddFooter (Odd Page Footer), and oddHeader (Odd Header).
section	header		Header section. Each header collects one, two, or three sections (left, center, or right sections).
worksheet	Part	External	The worksheet tag contains cells.
cell	links, worksheet	External	A cell in a worksheet or external link. This element contains an "r" (Reference) attribute and font name, size, and color (three general styles).
fieldproperties	worksheet	External	Field property collection which holds all placeholder information of a nearby cell.
field	cf, header, section, fieldproperties	External Internal	A field in header, section, or cf isplaceholder, which specified the presence an auto-field, such as, page (Page number), pages (Total number of pages), time (Time), date (Date), path (Path of current file), file (filename of current file), tab (Tab name of current worksheet), and pic (Background image). These internal tags can be added/deleted/reordered. In the field property section, a field holds additional information (translatable and non-translatable) extracted from of an inline placeholder.
fieldtext	field	External	The fieldtext (Field text) tag specifies translatable content.
picture	textbox	External	A picture in a textbox. "picture" may have alt (alternative text) which is translatable.
alt	picture	External	The alt (Alternative Text description) tag of a picture specifies alternative text for the current DrawingML object, for use by assistive technologies or applications which will not display the current object.
group	group, part	External	The group tag contains textboxes or other groups.
textbox	group	External	The textbox (Rich Text Box Content Container) tag specifies that its contents shall be any rich WordprocessingML content, and that this content is the rich contents of a drawing object defined using the Vector Markup Language (VML) syntax.
cf	cell,comment, p, section,	Internal	The cf (Characters format) tag is the most common and the most important tag. It holds, in attributes, all formatting



Tag name	Parent	Type	Explanation
	textbox		information, such as font name, size, color, and all kinds of styles, such as bold, italic, underline, and strikethrough.
br	alt, cell, cf, p, section, textbox	External Internal	The br (Line Break) tag depends on a user option, which can be internal or external style, resulting in a segmentation difference.

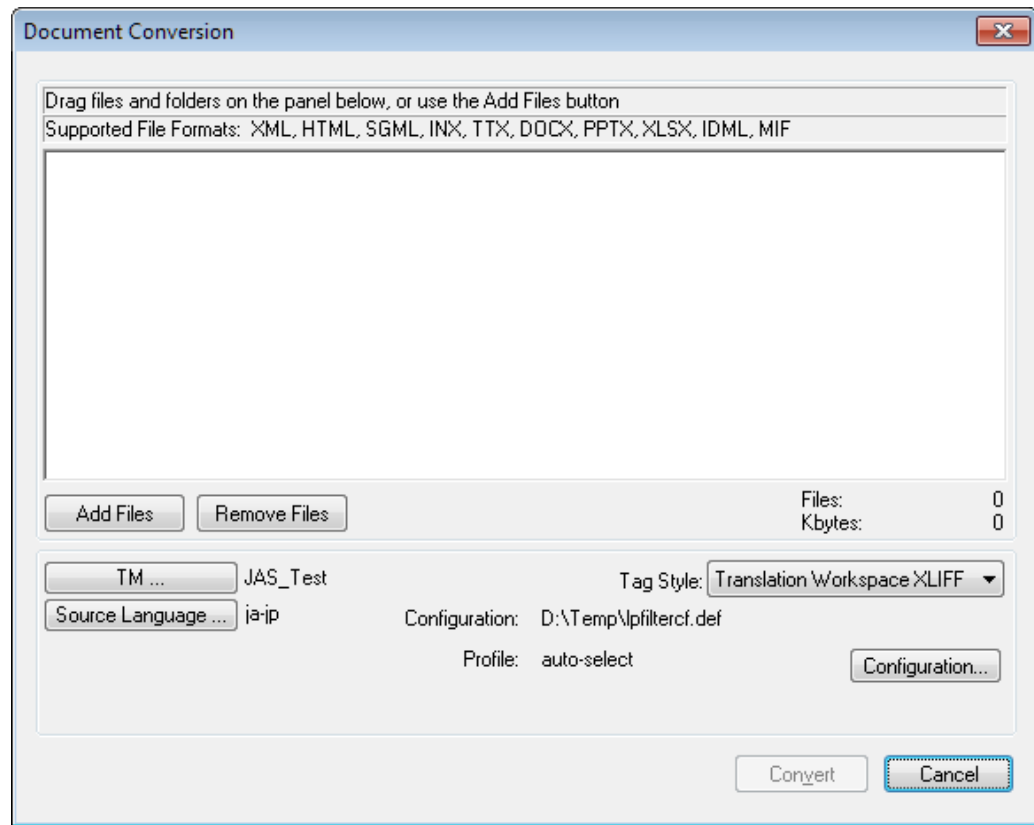
Note:

1. In general, all external tags are not editable.
2. The <part> tag, except the worksheet <part>, can be deleted, moved, or reordered. A part can be header, comments, external links, diagrams. Deleting one part means this part is skipped from localization process.
3. Internal tags are editable.
4. All external fields in header section are editable, removable, added, reordered.



FrameMaker and InDesign Files (MIF, IDML, INX)

To get MIF, IDML and INX into XLIFF, RTF, or Trados, use the **Translation Workspace Document conversion** dialog, which has a MIF, an IDML, and an INX filter.



MIF Filter

The Translation Workspace FrameMaker filter supports the Maker Interchange Format files (MIF) generated with FrameMaker version 14 and earlier. Structured FrameMaker MIF files are also supported.

Warning: Use MIF 14 files conversion with care as it has not yet been tested thoroughly.

Note: Changes marked in MIF files with Change Bars are accepted prior to conversion.

To save the FM files to MIF format, select **File > Save as...** in FrameMaker and select the MIF format in the Save as type drop-down list.

Defining text as Untranslatable

When any part of the source document should not be translated, you can modify the original FrameMaker file, to let the Translation Workspace MIF Filter mark the specific text as untranslatable.

To mark text as untranslatable, create in FrameMaker a character or paragraph tag (format) called LogoportMarkup or LogoportDoNotTranslate.

Apply the created tag (format) to the untranslatable parts of texts.

During the filtering, any text marked with this tag is set as untranslatable.



You can also take any character or paragraph Tag name existing in the FM files and add it as "untranslatable text" into the Translation Workspace TM settings.

MIF TagML

The tags for the TagML that you get when you apply the convert to Translation Workspace RTF or XLIFF format filter to a MIF file is shown in the following table.

Color coding:

Tag – Can be moved, deleted, and added

Tag – Can be moved and reordered

Tag – Can be moved, but not reordered

Adding new tags is supported only for the formatting tags. All other are synchronized with the source MIF. Therefore, new tags will cause a "tag missed" error.

All internal tags can be moved inside black text.

Those that do not have the 'id' property should not change their sequence (order) because Translation Workspace expects the initial tags order as in the source MIF during the backward conversion. The following elements tags can't be reordered: elb, ele, Suffix, and Prefix.

Tag Name	Parent	Type	Explanation
TagML		External	
page	TagML	External	A page
textrect	TagML,page	External	A frame
Table	p	External	Table
TableTitle	Table	External	Table Title
TableHeader	Table	External	Table Header
TableBody	Table	External	Table Body
TableFooter	Table	External	Table Footer
tr	TableBody, TableHeader, TableFooter	External	Table Row
td	tr	External	Table Cell
p	Table, Page, textrect, TagML	External	Paragraph
fieldproperties	Page, textrect, TagML	External	Field properties collection
field	fieldproperties	External	Field
fieldtext	field	External	Field display text
fieldcode	field	External	Field code
footnotes	Page	External	Footnotes collection
footnote	footnotes	External	Footnote Text
ti	P	External	TextInset (start)
tie	P	External	TextInset (end)
footnotereference	p,cf	Internal	Footnote Reference
afr	p,cf	Internal	Anchored frame
cns	cf,p	Internal	Conditional Text style start



Tag Name	Parent	Type	Explanation
ecns	cf,p	Internal	Conditional Text style end
CondMarker	cf,p	Internal	Conditional Text marker
hyperlink	cf,p	Internal	Hyperlink
alert	cf,p	Internal	Alert message
alerttitle	cf,p	Internal	Alert title
HypertextMarker	cf,p	Internal	Hypertext Marker
IndexMarker	cf,p	Internal	Index Marker
commentreference	cf,p	Internal	Comment Reference
Marker	cf,p	Internal	Marker
HeadFootMarker1	cf,p	Internal	Header Footer Maker
HeadFootMarker2	cf,p	Internal	Header Footer Maker
XrefMarker	cf,p	Internal	Cross Reference Marker
xref	cf,p	Internal	Cross reference
var	cf,p	Internal	Variable (ex: date)
bb	vartext	Internal	Format Building blocks
cf	p	Internal	cf (Characters format) is the most common which holds, in attributes, all formatting information, such as font name, size, color, and all kinds of styles, such as bold, italic, underline, and strikethrough.
b	p	Internal	Bold
i	p	Internal	Italic
s	p	Internal	Small Caps
so	p	Internal	Sort Order
bi	p	Internal	Bold Italic
flq	p,cf	Internal	French-style left double quote
frq	p,cf	Internal	French-style right double quote
glq	p,cf	Internal	German-style left double quote
grq	p,cf	Internal	German-style right double quote
t	p,cf	Internal	Tab
gr	p,cf	Internal	
qs	p,cf	Internal	
lq	p,cf	Internal	Left single curly quote
rq	p,cf	Internal	Right single curly quote
ldq	p,cf	Internal	Left double curly quote
rdq	p,cf	Internal	Right double curly quote
ts	p,cf	Internal	Thin space
ens	p,cf	Internal	En space
ems	p,cf	Internal	Em space
sh	p,cf	Internal	Hyphen (no option for soft hyphen)
hs	p,cf	Internal	Hard (non breaking) space



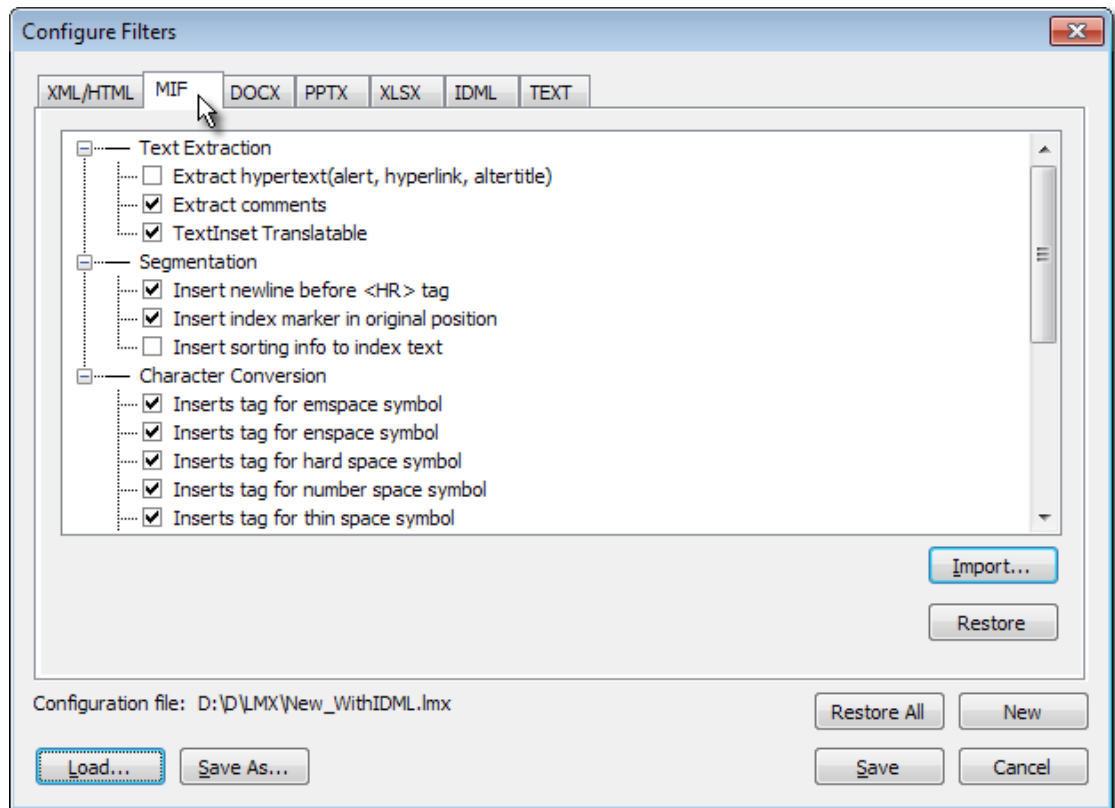
Tag Name	Parent	Type	Explanation
hh	p,cf	Internal	Hard (non-breaking) hyphen
dh	p,cf	Internal	Discretionary hyphen
nh	p,cf	Internal	No hyphen
hr	p,cf	Internal	Hard (non breaking) return
ns	p,cf	Internal	Number space
gt	p,cf	Internal	Greater than symbol
lt	p,cf	Internal	Smaller than sign
rc	p,cf	Internal	
rt	rc	Internal	
CrossReferenceFormat	TagML	External	Cross reference Format section
CrossReference	CrossReferenceFormat	External	Cross reference format text
FootnoteFormat	TagML	External	Footnote Format section
FootnoteAnchorPrefix	FootnoteFormat	External	Footnote anchor prefix text
FootnoteAnchorSuffix	FootnoteFormat	External	Footnote anchor suffix text
FootnoteNumberPrefix	FootnoteFormat	External	Footnote number prefix text
FootnoteNumberSuffix	FootnoteFormat	External	Footnote number suffix text
TableFootnoteFormat	TagML	External	Table Footnote Format section
TableFootnoteAnchorPrefix	TableFootnoteFormat	External	Table footnote anchor prefix text
TableFootnoteAnchorSuffix	TableFootnoteFormat	External	Table footnote anchor suffix text
TableFootnoteNumberPrefix	TableFootnoteFormat	External	Table footnote number prefix text
TableFootnoteNumberSuffix	TableFootnoteFormat	External	Table footnote number suffix text
DocumentNumberingFormat	TagML	External	Document numbering format section
DocumentFootnoteLabel	DocumentNumberingFormat	External	Document footnote label
DocumentTableFootnoteLabel	DocumentNumberingFormat	External	Document Table footnote label
DocumentVolume	DocumentNumberingFormat	External	Document Volume
DocumentChapter	DocumentNumberingFormat	External	Document Chapter
variableFormat	TagML	External	Variable Format section
var	variableFormat	External	Variable name
Vartext	var	External	Variable Text
elb	P,cf	External Internal	Structured SGML Element begin
ele	P,cf	External Internal	Structured SGML Element end
Suffix	P,cf	Internal	Structured SGML Element suffix
Prefix	P,cf	Internal	Structured SGML Element prefix
ElementPrefixSuffixFormats	TagML	External	Structured SGML Element Prefix suffix format
ElementPrefix	ElementPrefixSuffixFormats	External	Structured SGML Element Prefix format
EPText	ElementPrefix	External	Structured SGML Element Prefix Text
ElementSuffix	ElementPrefixSuffixFormats	External	Structured SGML Element suffix format



Tag Name	Parent	Type	Explanation
ESText	ElementSuffix	External	Structured SGML Element suffix text

Converting MIF to RTF or XLIFF

Drag and drop the MIF files to the **Document Conversion** dialog and click the **Configuration...** button.



Specify a configuration file (.lmx) and make any adjustments to the character conversion, text extraction, or segmentation fields.

Select the conversion to RTF or XLIFF in the Tag Style drop down list. The conversion uses the default MIF configuration, even if a configuration file has been selected.

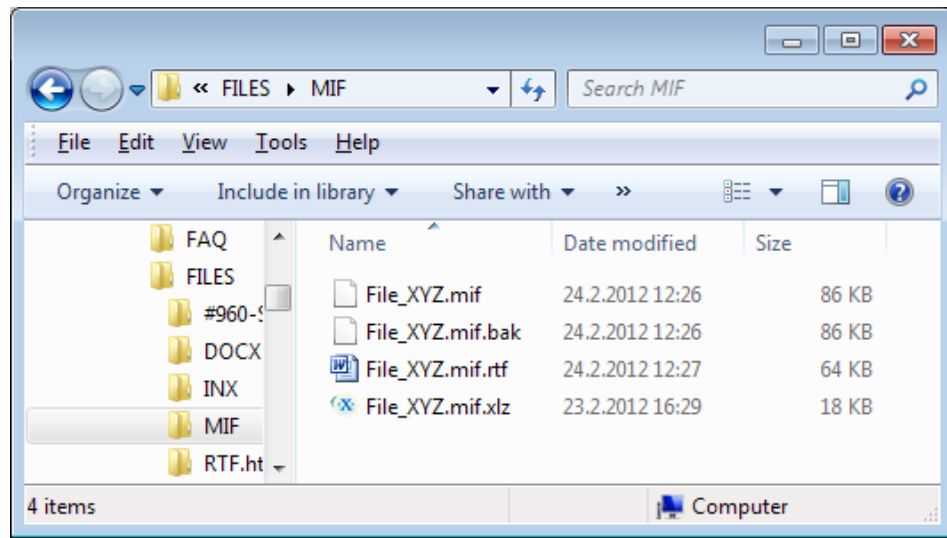
The converted file is stored in the same folder as the original MIF file.

Converting back to MIF

Use the **Back convert** dialog to generate a translated MIF file.

Put the original MIF file in the same folder with the file you are converting back to MIF. Make sure *not* to modify the original file extension of your translated file, because the filter checks it to recognize the file as converted from a MIF file.

Running the back conversion creates a new MIF file and renames the source MIF file with a .bak extension.



The selected **Codepage** or the **Character to entities** setting of the **Back Convert** dialog has no effect on the MIF files back conversion.

The **Target Language...** selection, however, does affect the back conversion. Depending on the selected language, some language settings are applied in the generated MIF file.

Additional MIF filter information

To configure the MIF filter, select the MIF tab in the **Configure Filters** dialog. The MIF filter has several options.

Character Conversion

You can decide how some special characters should be represented in the converted files.

Option	Description
Inserts tag for emspace symbol	If enabled, inserts <ems/>
Inserts tag for enspace symbol	If enabled, inserts <ens/>
Inserts tag for hard space symbol	If enabled, inserts <hs/>
Inserts tag for number space symbol	If enabled, inserts <ns/>
Inserts tag for thin symbol	If enabled, inserts <ts/>
Inserts tag for tab symbol	If enabled, inserts <t/>
Inserts tag for lower than symbol	If enabled, inserts <lt/>
Inserts tag for greater than symbol	If enabled, inserts <gt/>
Inserts tag for soft hyphen symbol	If enabled, inserts <sh/>
Inserts tag for hard hyphen symbol	If enabled, inserts <hh/>
Inserts tag for discretonal hyphen symbol	If enabled, inserts <dh/>
Inserts tag for discretonal hyphen symbol	If enabled, inserts <nh/>



Translation Workspace has a particular option to manage Soft hyphen characters:

Option	Description
Keep soft hyphen symbols	When disabled, the soft hyphen symbol characters are removed. When enabled, the soft hyphen symbol characters are kept. The way it will be represented in the output file depends also on option Inserts tag for soft hyphen symbol.

Text Extraction

These parts of the MIF file are extracted:

Text run	Text insets
Master Pages (those used In files)	Cross references (including Format definitions used in files)
Normal Pages	Variables (including Format definitions which are used in files)
Tables	Numbering Formats (which are used in files)
Anchored Frames	Inline Graphics, Links to graphics
Headers and Footers	Conditional texts
Footnotes, Endnotes	Index, Index Markers, Index Entries
Grouped Objects	Index sort order (optional)
Nested Objects	Comments (optional)
Text Boxes	Hypertext (optional)

Note: The MathML objects (available in MIF 12) are not extracted.

The **Text Extraction** option lets you to select following options when extracting for translation:

Option	Description
Extract Hypertext (alert, hyperlink, alertitle):	When enabled, extracts FrameMaker Hypertext definitions for translation.
Extract comments	When enabled, extracts FrameMaker comments for translation.
TextInset Translatable	When enabled, extracts FrameMaker text insets for translation.

Note: The MathML objects (available in MIF 12) are not extracted.

Segmentation

You can control the segmentation with following options:

Option	Description
Insert newline before <HR> tag	Hard Returns are represented as internal <hr> tags. They are typically considered as ending the segment. In some situations however a hard return is not preferred as the end of segment character. Therefore there is an option to insert a newline before <hr> tag (when the option is enabled).



Insert index marker in original position	When enabled, index markers stay at their original position in the sentence, marked with Internal style, and represented by the tag <IndexMarker>. When disabled, index markers are output at the end of the current paragraph and marked with External style, represented by the tag <IndexMarker>.
Insert sorting info to index text	In FrameMaker you can change where an entry appears in the index by specifying a sort order. In the MIF filter there is an option to manage index entries sort order. When disabled, no index information is generated. for example: <pre><fieldtext type="XE">my index</fieldtext></pre> When enabled, an extra tag with sort index translatable text is output: <pre><fieldtext type="XE">my index</fieldtext> <fieldtext type="Sort-XE">my index</fieldtext></pre> Note: If sort index information already exists in the original MIF file, it is always extracted for translation, and this option has no effect.

Output options

Option	Description
Include the Source File (only for XLZ format)	When converting to XLZ this option disabled, the conversion creates an XLZ file without embedding the MIF source file within it. When converting to XLZ with this option enabled, the MIF source file is embedded within the XLZ output file..

Page Breaks

The conversion process removes page breaks to avoid splitting sentences.

Default setting

Using the default settings of the MIF filter will ensure the best compatibility with "Trados S-tagger" filtering.

IDML Filter

With the Translation Workspace IDML Filter you can convert Adobe InDesign Markup files to RTF and XLIFF for localization with Translation Workspace.

Translation Workspace IDML Filter supports the InDesign Markup Language format (IDML), available from Adobe InDesign CS4 onwards. Translation Workspace currently supports InDesign CS4 and CS5.

To save the InDesign files to IDML format, select **File > Export...** in Adobe InDesign and then select the **InDesign Markup (IDML)** format in the **Save as type** drop-down list.

Defining Text as Untranslatable

For any part of the source IDML document which should not to be translated, you can modify the original InDesign file so that the Translation Workspace IDML Filter marks the specific text as untranslatable.



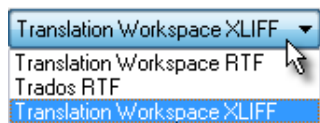
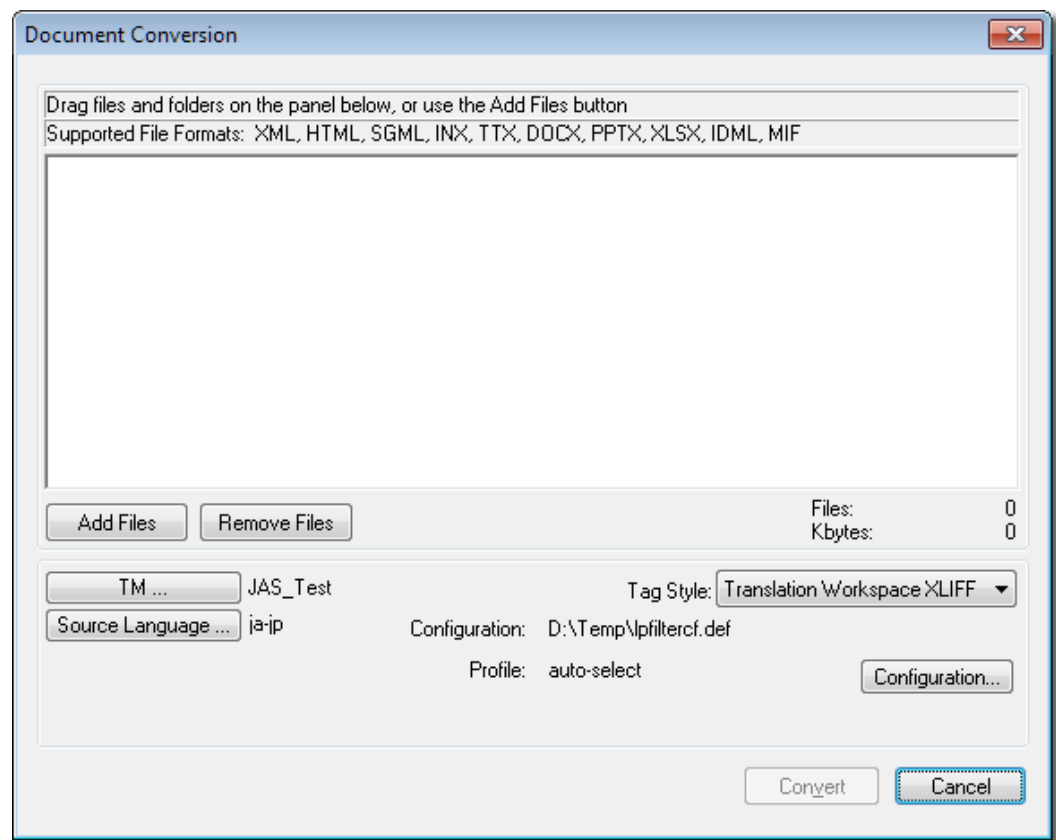
To mark text as untranslatable, create a tag in InDesign named *lionbridgeuntranslatable*, and apply it to the untranslatable parts of text.

During filtering, any text marked with this tag is set as untranslatable.

Converting IDML to RTF or XLIFF

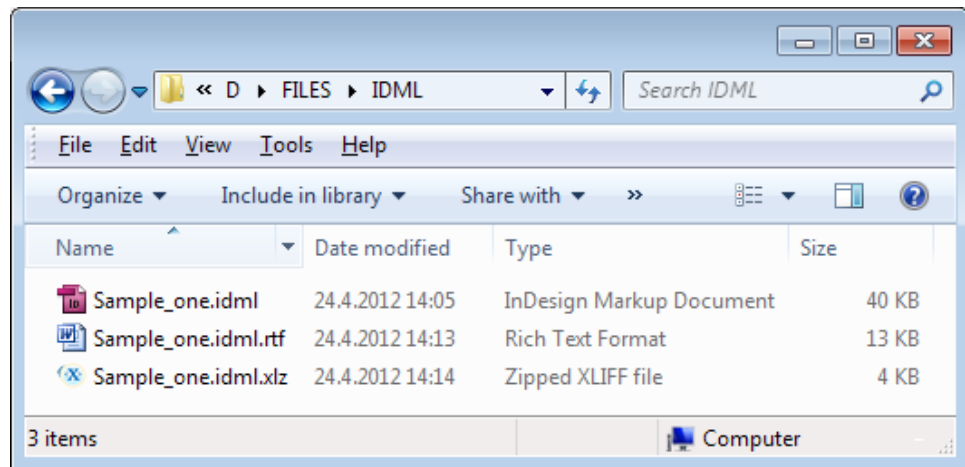
Launch the Translation Workspace Converter by selecting **Documents > Convert**.

Drag and drop the IDML files to the **Document Conversion** dialog (or click **Add Files** and browse to the files).



Select the conversion to RTF or XLIFF in the **Tag Style** drop down list. You can select a configuration file (.lmcx) with desired options. If no configuration file is specified, the default settings file that is stored on the server is automatically used.

The converted file is stored in the same folder as the original IDML file.

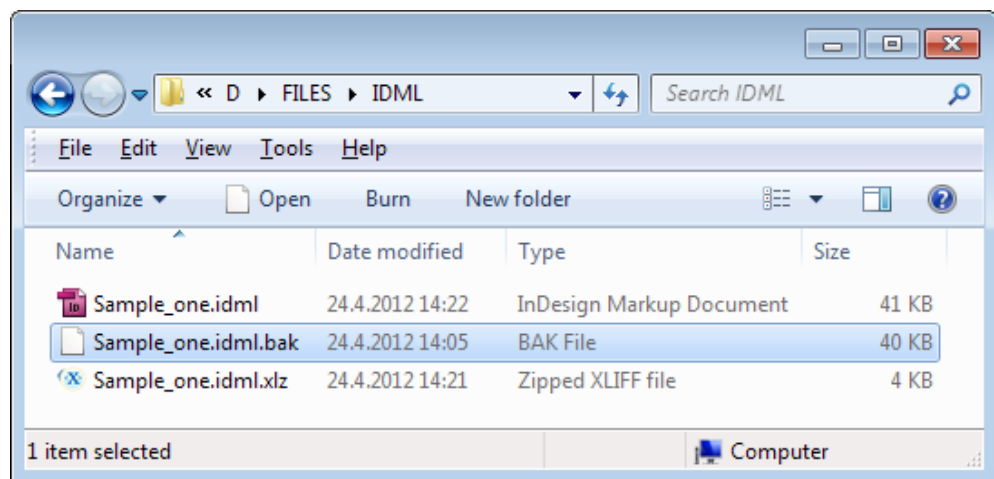


Converting Back to IDML

You can use the **Back Convert** function to generate a translated version of an IDML file.

The original IDML file should be in the same folder as the RTF or XLZ file to be back converted. Make sure to keep the original file extension of the translated file unchanged, because the filter checks it to recognize the file as converted from an IDML file.

Translation Workspace creates a new IDML file and renames the source IDML file with a `.bak` extension.



The selected **Codepage** or the **Character to entities** setting of the Back Convert dialog has no effect on the back conversion to IDML files.

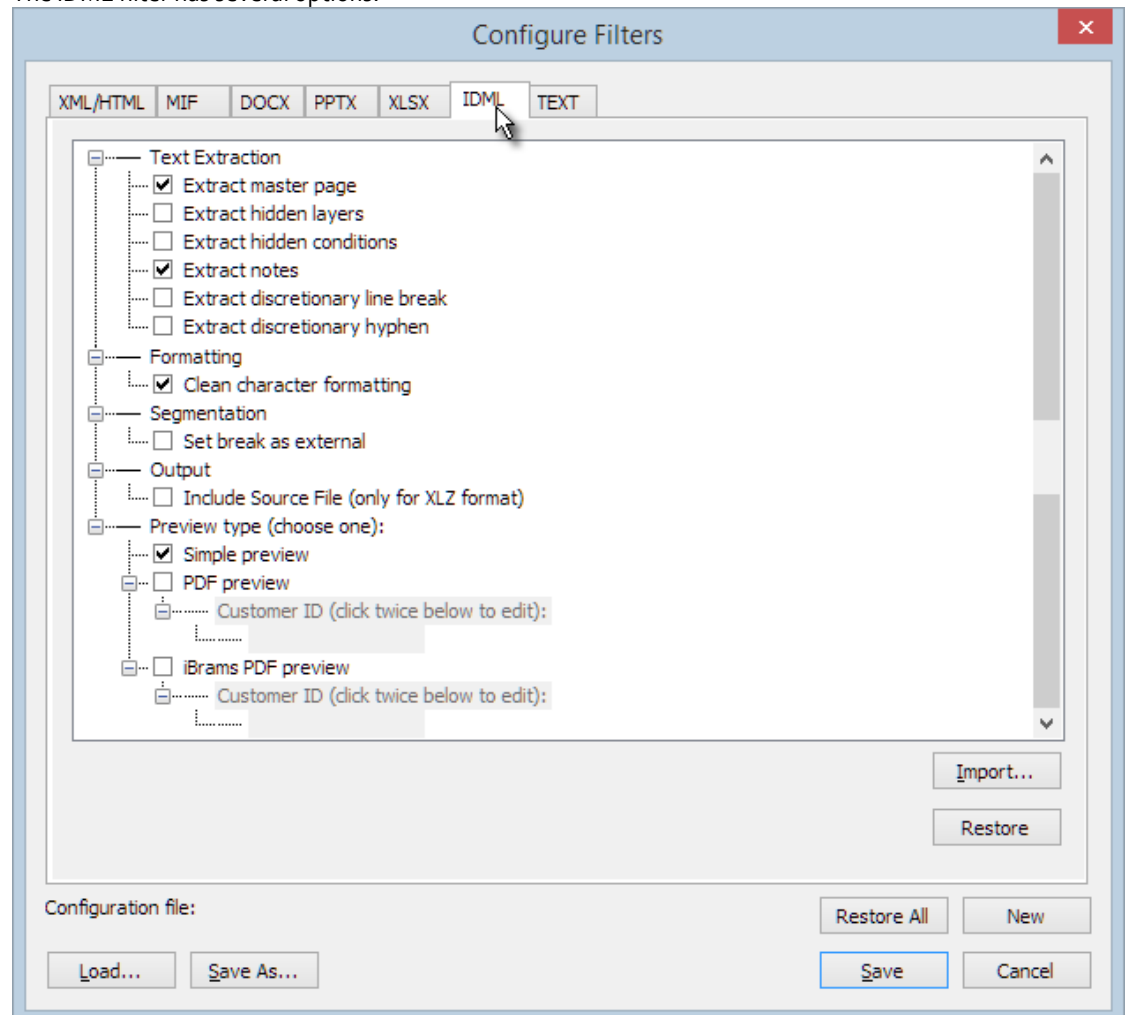
The **Target Language...** selection, however, does affect the back conversion. Depending on the selected language, some language settings are applied in the generated IDML file.

The **Inter-Segment Spaces** option selected in the Document Back Conversion Dialog and Document Cleanup Dialog affects the back conversion.



Additional IDML Filter Information

To configure the IDML filter, select the *IDML* tab in **Documents > Convert > Configuration....**
The IDML filter has several options.



Text Extraction

These parts of the IDML file are extracted:

- Normal spreads
- Text Boxes
- Tables
- Headers and footers
- Footnotes, Endnotes
- Variables (including Format definitions)
- Cross references (including Format definitions)
- Notes (optional)
- Master Pages (optional)
- Conditional texts (optional)
- Hidden layers (optional)



- ❑ Discretionary line break (optional)
- ❑ Discretionary hyphen (optional)

Note: TOCs and Indexes are NOT included. These can be re-generated in the translated IDML documents.

The **Text Extraction** option lets you select the following options when extracting for translation:

Option	Default Value	Description
Extract master page	True	When enabled, text from Master pages is extracted for translation.
Extract hidden layers	False	When enabled, text from hidden layers is extracted for translation
Extract hidden conditions	False	When enabled, text from hidden conditional text is extracted for translation
Extract notes	True	When enabled, text from notes is extracted for translation.
Extract discretionary line break	False	When enabled, discretionary line breaks are extracted
Extract discretionary hyphen	False	When enabled, discretionary hyphens are extracted
Clean character formatting	True	When enabled, the first character style use in the paragraph is extracted as the paragraph level. (It decreases the number of internal tags in the segments.)

Segmentation

Forced Line breaks are converted into `<br/>` tags. By default, these tags are extracted in Markup style, but you can also decide to extract them in `LogoportDoNotTranslate` style if forced line breaks need to be considered as end of segment marker.

Paragraph breaks are converted into `<pbreak/>` tags which are considered as end of segment markers. This will result in the segments being broken at all Paragraph breaks.

You can control the segmentation with the following option:

Option	Default Value	Description
Set Break as External	False	When enabled, forced line break represented as <code>
</code> tag is set in <code>LogoportDoNotTranslate</code> style, and considered as an end of segment marker.



Output options

Option	Description
Include the Source File (only for XLZ format)	When converting to XLZ this option disabled, the conversion creates an XLZ file without embedding the IDML source file within it. When converting to XLZ with this option enabled, the IDML source file is embedded within the XLZ output file..

Preview Type

This field is used only for specific advanced InDesign preview:

Option	Description
Simple preview	Previews the XLZ file in a web browser.
PDF preview	Please contact the support team for more details.
iBrams PDF preview	Please contact the support team for more details.

Track changes

For files with track changes, we recommend to approve/reject the track changes before the conversion.

If the IDML files contain inserted or deleted changes, IDML filter accepts inserted text and abandons deleted text. These changes however may change paragraph alignment, which may cause paragraph style to shift.

A list of all the accepted/rejected changes will be available on the `.err` file created during conversion.

Entities Management

Since IDML files are XML, extended characters are not converted into entities.

The special XML characters `<`, `>`, `&` and `"` are converted to entities during conversion.

Inline Graphics

The position of Inline graphics is preserved in the localized file.

Embedded Graphics

Since InDesign strips all preview information from IDML files (in order to keep the file size as small as possible), there is a loss of image previews in InDesign documents with either embedded images or when we do not have the original linked images.

Keep this in mind when using the IDML filter, as it will require all images to be copied and pasted manually from the source document during post-production.

Since this information is already lost in the source IDML file, we have no solution for this currently.

Font Handling in IDML PDF Preview

Online Review Client and XLIFF Editor use the fonts in the original InDesign file source text for the IDML PDF preview. Those fonts may not support the characters of the target language, failing to display the characters in the preview, the same way as when the IDML file is opened directly in InDesign. This occurs most likely for documents translated to languages with



characters outside the common Western character set (e.g. Chinese, Arabic, East European languages).

When a font used in an IDML file is unavailable on the InDesign server generating the preview, the font is substituted by an available font, in the same way as InDesign usually works.

You can avoid font problems by preparing the IDML files, by applying a font which supports the necessary target language characters before conversion to XLZ format. If you need additional fonts to be uploaded onto the preview server, please contact GeoWorkz.com.

Note: The HTML preview in Online Review Client and XLIFF Editor does not need to apply the original fonts, as it renders the preview with fonts available on the user's system. This allows the HTML preview to display the characters correctly even when the PDF preview fails to do so for the same document.

Known Limitations

All manual kerning and tracking information is set to 0 (zero) during conversion to RTF and XLZ.

Compatibility with legacy TW INX material

- ☒ Extract discretionary line break
- ☒ Extract discretionary hyphen
- ☐ Formatting
 - ☐ Clean character formatting
- ☐ Segmentation
 - ☒ Set break as external

If you need to prepare IDML files and leverage from TM containing legacy material prepared with the Translation Workspace INX filter, we recommend that you set the following options in the IDML Filter.

INX Filter

The Translation Workspace InDesign filter supports the InDesign Interchange format (INX), available from InDesign CS2 onwards.

You can use the **Export...** menu in InDesign and then select the InDesign Interchange format.

Defining text as Untranslatable

When any part of the source document should not be translated, you can modify the original InDesign file, to let the Translation Workspace InDesign Filter mark the specific text as untranslatable.

To mark text as untranslatable, create a tag in InDesign called `lionbridgeuntranslatable`, and apply it to the untranslatable parts of text.

During the filtering, any text marked with this tag is set as untranslatable.

INX TagML

The tags for the TagML that you get when you apply the convert to Translation Workspace RTF or XLIFF format filter to an INX file is shown in the following table.

Tag Name	Parent	Type	Explanation
document		External	The document
externallinks	Document	External	
index	Document	External	Index
masterspread	Document	External	A Master spread
normalspread	Document	External	A Normal Spread

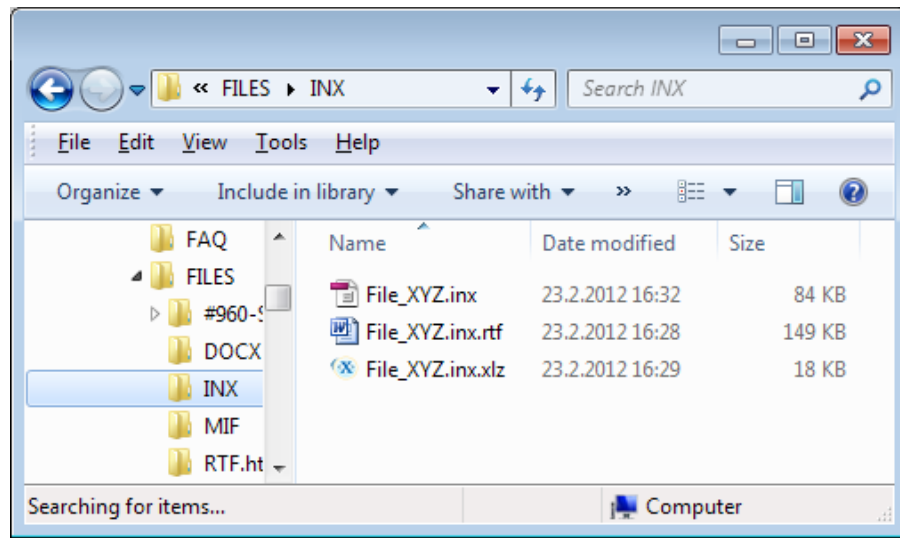


Tag Name	Parent	Type	Explanation
spread	Masterspread, normalspread	External	A Spread
cflo	Document, spread	External	Content Flow/Story
table	cflo	External	Specifies a table structure.
row	table	External	Table row
cell	row	External	Table cell
linebreak	Cell,Cflo	External	A line break
paragraph	Cell,Cflo	External	Specifies a paragraph of content
pbreak	Cell,Cflo	External	A Page break
address	externallinks	External	Link Address
topic	index	External	Index Topic
ColumnBreak	paragraph	Internal	Column break
ForcedLineBreak	paragraph	Internal	Forced line break
PageBreak	paragraph	Internal	Page break
PageNumber	paragraph	Internal	Page number
RightAlignedTab	paragraph	Internal	Tab (right aligned)
RightIndentTab	paragraph	Internal	Tab (right indent)
SDSq	paragraph	Internal	
SPTv	paragraph	Internal	
SectionName	paragraph	Internal	Section name
UnknownAid	paragraph	Internal	
autopagenummer	Paragraph	Internal	
cf	Paragraph	Internal	cf (Characters format) is the most common and the most important tag, which holds, in attributes, all formatting information, such as font name, size, color, and all kinds of styles, such as bold, italic, underline, strikethrough...
hairspace	Paragraph	Internal	
hyperlink	Paragraph	Internal	hyperlink (Hyperlink) specifies the presence of a hyperlink at the current location
image	Paragraph	Internal	An Image
indenthere	Paragraph	Internal	
indexmarker	Paragraph	Internal	Index Marker
nbh	Paragraph	Internal	A non breaking hyphen
punctuationspace	Paragraph	Internal	A Punctuation space
tablemarker	Paragraph	Internal	Table Marker
thinspace	paragraph	Internal	A thin space

Converting INX to RTF or XLIFF

You can choose the conversion to RTF or XLIFF from the Tag Style drop down list. A default INX configuration will be used for the conversion, even if a configuration file has been selected.

The converted file will be stored in the same folder as the original INX file.



Converting back to INX

You can use the **Document Back Conversion** function to generate a translated version of an INX file. The original INX file should be in the same folder as the file you are converting back to INX. Make sure *not* to modify the original file extension of your translated file, because the filter checks it to recognize the file as converted from an INX file.

Translation Workspace creates a new INX file and renames the source INX file with a `.bak` extension.

INX files are always UTF-8 xml files and are therefore not affected by the selected Codepage or the Character to entities setting.

Additional INX filter information

Translation Workspace extracts these elements from the INX file:

Master Spreads	Headers and footers	Tables
Normal Spreads	Text Boxes	

Note: TOCs and Indexes are NOT included.
These can be re-generated after the body text has been localized.

Paragraph and Line Breaks treatment

Line breaks are converted into the `<linebreak>` tags and are therefore considered as end of segment markers. This will result in the segments being broken at all Line breaks.

Paragraph breaks are converted into the `<pbreak>` tags and are therefore considered as end of segment markers. This will result in the segments being broken at all Paragraph breaks.

Entities management

Since INX files are XML, extended characters are not converted into entities.

The special XML characters `<`, `>`, `&` and `"` are converted to entities during conversion.

Inline graphics

The position of Inline graphics is preserved in the localized file.



Embedded graphics

Since InDesign strips all preview information from INX files (in order to keep the file size as small as possible), there is a loss of image previews in InDesign documents with either embedded images or when we do not have the original linked images.

Keep this in mind when using the INX filter, as it will require all images to be copied and pasted manually from the source document during postproduction.

Since this information is already lost in the source INX file, we have no solution for this currently.

Hidden and locked layers

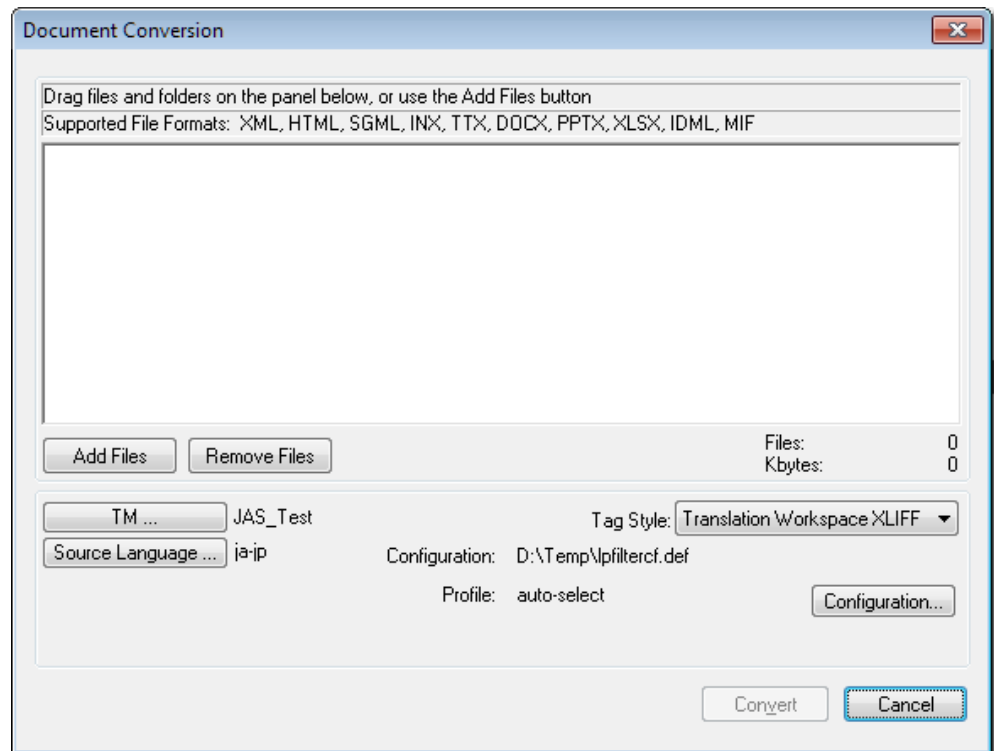
Localizable Content in hidden and locked layers is not extracted from the INX file.

External locations and hyperlinks are extracted as localizable content.

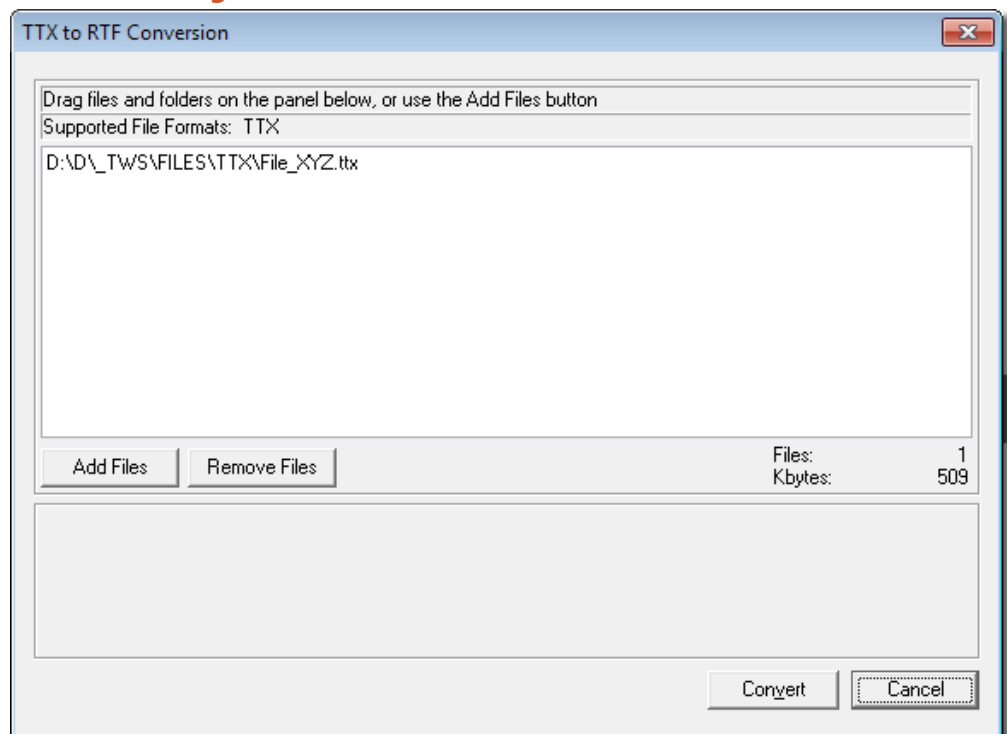


Industry standards (TTX and XLIFF)

To get TTX into XLIFF or RTF, use the **Translation Workspace Document** conversion dialog, which has a TTX filter.



TTX to RTF Conversion Dialog





The TTX to RTF Conversion dialog allows you to convert Trados TagEditor TTX files to Translation Workspace "tagged" RTF documents (that is, documents prepared for translation with Translation Workspace styles). You do not necessarily have to clean your bilingual files before conversion, but you can.

The Trados TagEditor Xtranslate (context TM) segments will be automatically converted to Translation Workspace locked 101% match segments.

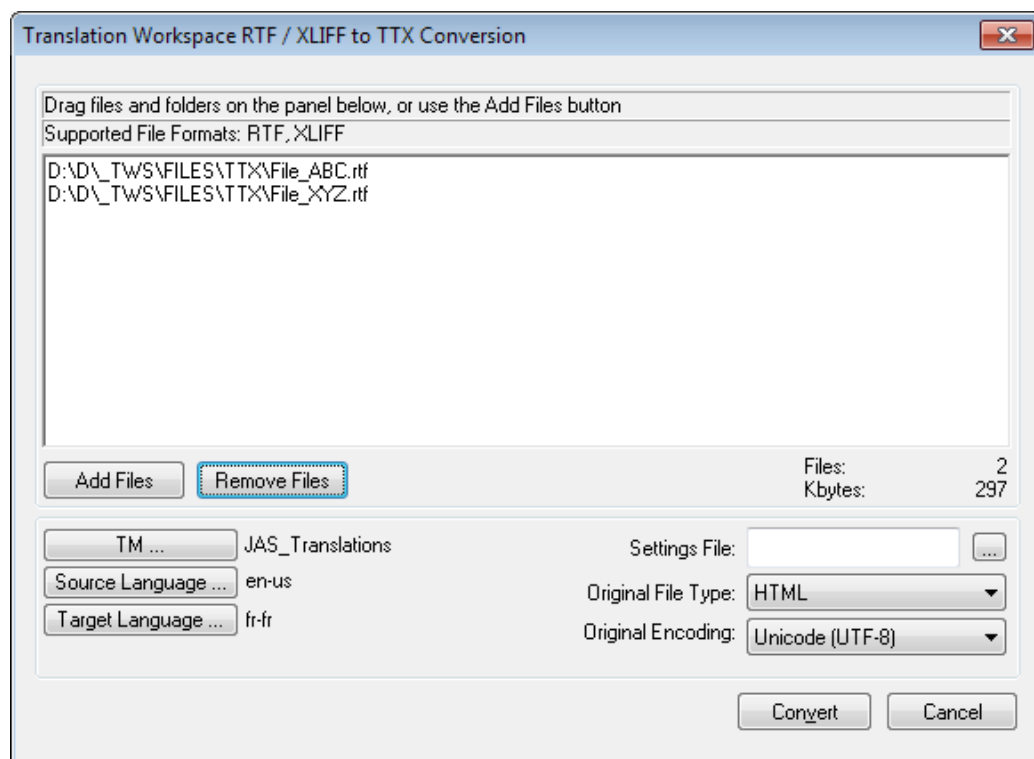
Drag the files or folders containing the files you want to convert on the dialog. The Translation Workspace client checks if the documents you chose are of the correct type (TTX) for conversion.

This simple dialog includes the following options:

Option	Description
Add Files	You may click on this button to select a number of files through the system's file open dialog. The same restrictions apply on the file type as explained above.
Remove Files	Clears the file selection. Does however not remove or delete files from your disk.
Convert	Clicking on this button starts the conversion, displaying a progress bar.
Cancel	Closes this dialog box without further action.

Note: Files in the Office 2007 DOCX format can be used in all batch functions. However, you need Office 2007 or the Office 2007 Compatibility Pack in order to use this function.

Translation Workspace RTF / XLIFF to TTX Conversion Dialog





The Translation Workspace **RTF / XLIFF to TTX Conversion** dialog allows you to convert documents that were converted with the TTX to RTF Conversion function back to Trados TagEditor TTX format. You do not necessarily have to clean your bilingual files before conversion, but you can.

Locked 101% matches in Translation Workspace will be automatically converted to Trados TagEditor Xtranslate (context TM) segments.

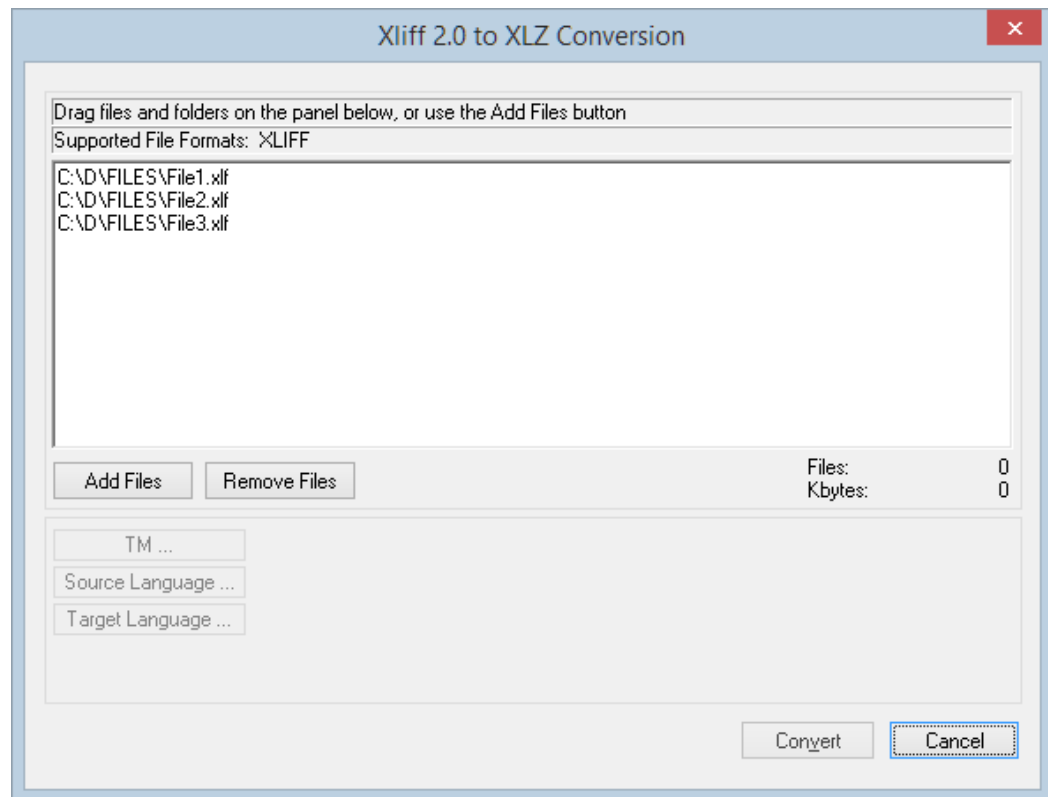
Drag the individual files or folders containing the files you want to convert on the dialog. The Translation Workspace client checks if the documents you chose are of the correct type (RTF/Doc/XLIFF) for conversion and only displays RTF/Doc/XLIFF documents.

This dialog contains the following options:

Option	Description																		
Add Files	You may click on this button to select a number of files through the system's file open dialog box. The same restrictions apply on the file type as explained above.																		
Remove Files	Clears the file selection. However, does not remove or delete files from your disk.																		
TM ...	Displays the TM selection dialog to choose the TM to which the selected files belong. This is necessary to retrieve the list of possible source and target languages available for the conversion.																		
Source Language	You should select a source and target language for the documents conversion.																		
Target Language	You should select a source and target language for the documents conversion.																		
DTD setting file	The DTD setting file is the Trados TagEditor .ini file that was used initially to create the original TTX files (before conversion to RTF). Note: This file is not needed if you are converting one of the following original file types to TTX: RTF, InDesign, or QuarkXPress.																		
Original File Type	This is the file format of the original file. The possible values are: <table><tr><td>HTML</td><td>FrameMaker</td><td>RTF</td></tr><tr><td>XML</td><td>PageMaker</td><td>Excel</td></tr><tr><td>SGML</td><td>InDesign</td><td>PowerPoint</td></tr><tr><td></td><td>QuarkXPress</td><td></td></tr><tr><td></td><td>Interleaf</td><td></td></tr><tr><td></td><td>Ventura</td><td></td></tr></table>	HTML	FrameMaker	RTF	XML	PageMaker	Excel	SGML	InDesign	PowerPoint		QuarkXPress			Interleaf			Ventura	
HTML	FrameMaker	RTF																	
XML	PageMaker	Excel																	
SGML	InDesign	PowerPoint																	
	QuarkXPress																		
	Interleaf																		
	Ventura																		
Original Encoding	This is the encoding of the files in their original format. This encoding will be used by the TagEditor when converting the translated TTX file to the target file in the original file format.																		
Convert	Clicking on this button starts the conversion, displaying a progress bar.																		
Cancel	Closes this dialog box without further action.																		



XLIFF 2.0 to XLZ Conversion Dialog



The **XLIFF 2.0 to XLZ Conversion** dialog allows you to convert XLIFF 2.0 documents to Translation Workspace XLZ format.

The XLIFF 2.0 metadata information available in the original file is extracted into the TW XLIFF 1.2 format which supports the metadata. The back conversion creates a valid XLIFF 2.0 file containing the preserved metadata module.

Drag the individual files or folders containing the files you want to convert on the dialog. The Translation Workspace client checks if the documents you chose are of the correct type (XLIFF 2.0) for conversion and conversion displays an error for XLIFF 1.2 documents.

This dialog contains the following options:

Option	Description
Add Files	You may click on this button to select a number of files through the system's file open dialog box. The same restrictions apply on the file type as explained above.
Remove Files	Clears the file selection. However, does not remove or delete files from your disk.
Convert	Clicking on this button starts the conversion, displaying a progress bar.
Cancel	Closes this dialog box without further action.



XLIFF 2.0 Translation Candidates module

XLIFF 2.0 Converter now supports the *XLIFF 2.0 – Translation Candidates Module* (XLIFF 2.0 specification, [5.1 Translation Candidates Module](#)). Once the data is available on the TW XLZ file, you can add it to the TM, and also display it in the XLIFF Editor Information panel.

There is an example of a translation candidate module (mtc) below:

```
<unit id="1">
  <mtc:matches>
    <mtc:match ref="#m1">
      <source>He is my friend.</source>
      <target>Il est mon ami.</target>
    </mtc:match>
    <mtc:match ref="#m1">
      <source>He is my best friend.</source>
      <target>Il est mon meilleur ami.</target>
    </mtc:match>
  </mtc:matches>
  <segment>
    <source><mrk id="m1" type="mtc:match">He is my
friend.</mrk></source>
  </segment>
  <segment>
    <source>Yet, I barely see him.</source>
  </segment>
</unit>
```

Conversion from XLIFF 2.0 to TW XLZ generates XLIFF 1.2 files which are valid and can be opened in the XLIFF Editor without any issues.

- Match candidate information available on the <unit> in XLIFF 2.0 is extracted to the corresponding <group> node in XLIFF 1.2
- Tags in match candidate module are converted from XLIFF 2.0 to XLIFF 1.2 if needed
- Reference between the match candidate and the source segment (<mrk> tag) in XLIFF 2.0 is preserved in a way which enables to link the translation candidate data extracted at the <group> level and the corresponding <trans-unit>.

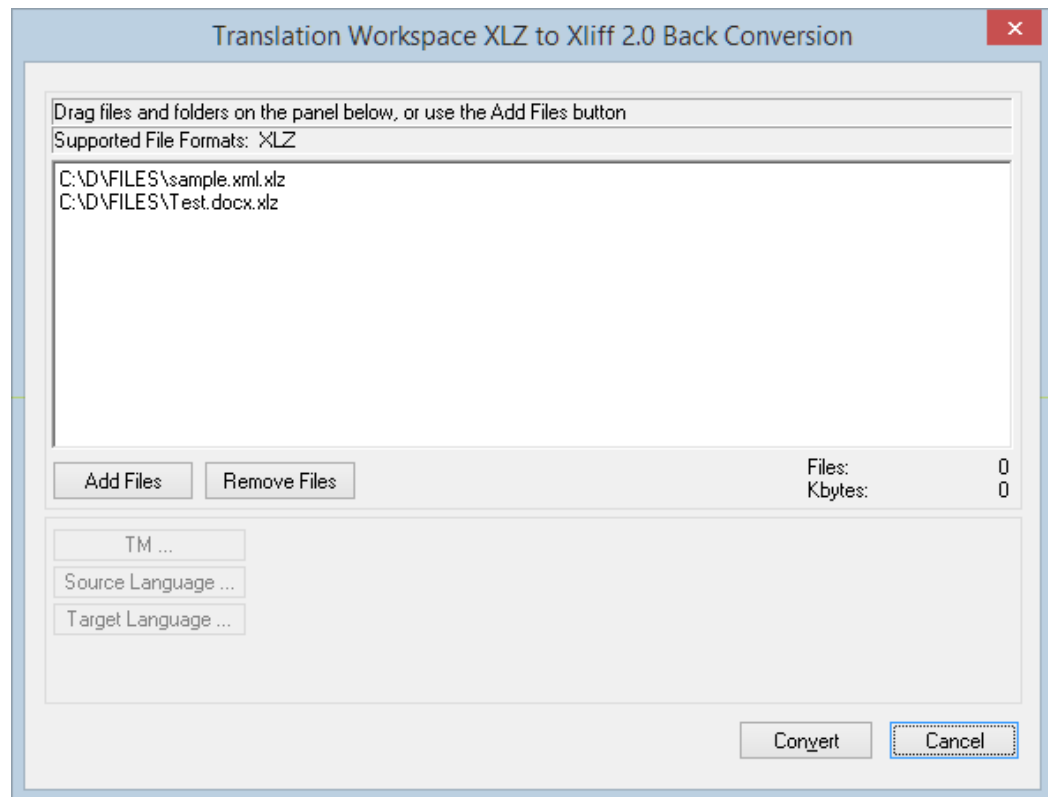
Back conversion from TW XLIFF creates valid XLIFF 2.0 files with translation candidate modules preserved as identical to those in the source XLIFF 2.0 file.

Storing Translation Candidate matches to a TM

While cleaning up XLZ documents in TW Tools and Word Client, you can store the Translation Candidate matches from XLIFF 2.0 into the TM instead of translated segments (as happens to the Alt-Trans units) by selecting the **Store Xliff Translation Candidate/Alt-Trans instead of segments** option in the **Document Cleanup** dialog (see Cleanup).



Translation Workspace XLZ to XLIFF 2.0 Back Conversion Dialog



The **Translation Workspace XLZ to XLIFF 2.0 Back Conversion** dialog allows you to convert Translation Workspace XLZ files converted from XLIFF 2.0 documents back to XLIFF 2.0 format.

The XLIFF 2.0 metadata information available in the original file was extracted into the TW XLIFF 1.2 format which supports the metadata. The back conversion creates a valid XLIFF 2.0 file containing the preserved metadata module.

Drag the individual files or folders containing the files you want to convert on the dialog. The Translation Workspace client checks if the documents you chose are of the correct type (Translation Workspace XLZ converted from XLIFF 2.0) for conversion and only displays from XLIFF 2.0 converted Translation Workspace XLZ documents. A `.bak` extension is added to a source XLIFF 2.0 file existing in the same folder with the TW XLZ file.

This dialog contains the following options:

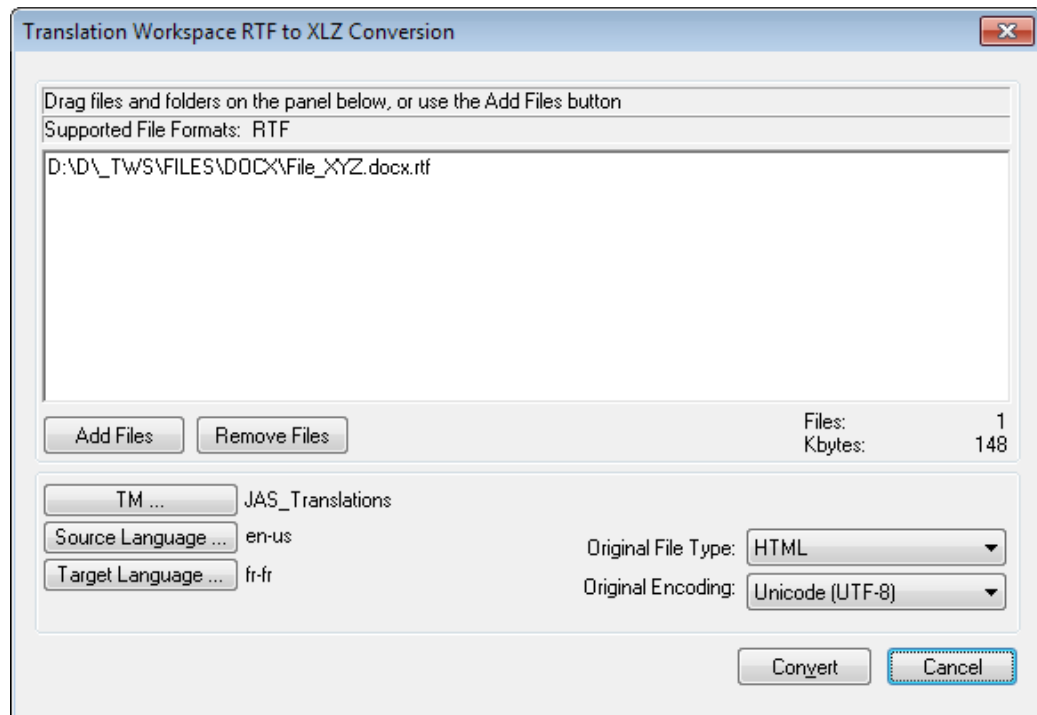
Option	Description
Add Files	You may click on this button to select a number of files through the system's file open dialog box. The same restrictions apply on the file type as explained above.
Remove Files	Clears the file selection. However, does not remove or delete files from your disk.
Convert	Clicking this button starts the conversion, displaying a progress bar.
Cancel	Closes this dialog box without further action.



Translation Workspace RTF to XLZ Conversion Dialog

The RTF to XLZ converter takes files you have converted to Translation Workspace tagged RTF and converts them to Translation Workspace XLZ files, which you can translate in the XLIFF Editor. The tagged RTF file **must not have been segmented or translated**. The filter recognizes Normal, Internal and External style text only. It does not strip tagged RTF before the operation, but will strip any extra formatting or styling.

Note: Layout constructs like tables, footnotes, endnotes, fields, links, graphics, and so on might cause unknown output from the conversion. SDL Trados tagged RTF files are supported. If the file has additional internal or external nonstandard styles, those will be marked by the `LogoportMarkup` or `LogoportDoNotTranslate` styles during back conversion.



To access the **Translation Workspace RTF to XLZ Conversion** dialog, either open the tools via **Start > All Programs > Translation Workspace > Tools > Translation Workspace Tools** and click **Convert**, then **RTF to XLZ** or open the RTF file and in the Translation Workspace ribbon, select the **Current Document** menu and **Convert RTF to XLZ**. You can also access the dialog from the Documents menu.

The converter has two options.

Option	Description
Original File Type	This can be set to a fixed list of data types and will be encoded into the XLIFF header. Setting this correctly enables back conversion of the XLZ file to the original file type in some cases.
Original Encoding	This can be set to the original encoding of the source material before conversion to RTF. It is also only set in the XLIFF header to act as a default if back conversion to original format is attempted directly from XLIFF. For the high level filters like OpenOffice XML, INX, MIF, and so on, it should be left at UTF-8, as that is the encoding of the internal representation.

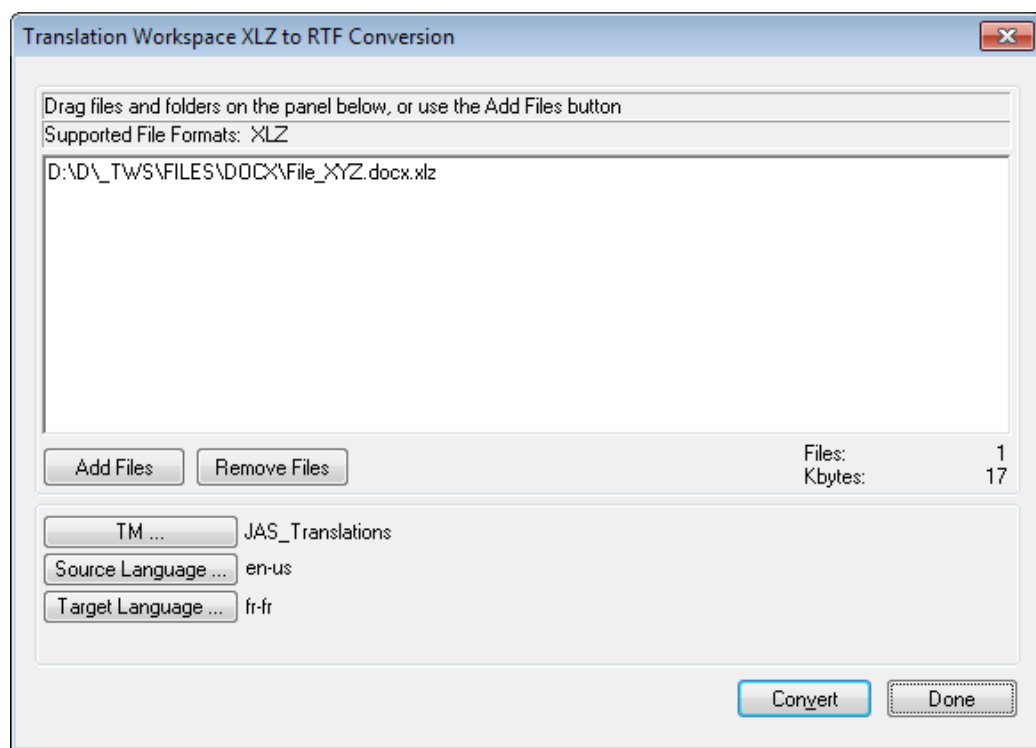


Besides conversion to the original format, the correct setting of these options will improve the ability to use the XLIFF Editor Preview feature for the file.

Translation Workspace XLZ to RTF Conversion Dialog

The XLZ to RTF converter takes Translation Workspace XLZ files and converts them to Translation Workspace RTF files, which you can work on in the Microsoft Word Plug-in or back convert from Translation Workspace RTF to the original format.

To access the **Translation Workspace XLZ to RTF Conversion** dialog, open the tools via **Start > All Programs > Translation Workspace > Tools > Translation Workspace Tools** and click **Convert**, then **XLZ to RTF**.



This converter has no options besides allowing you to change the TM, source, and target languages. The file will be converted to Translation Workspace RTF.

These language parameters are used in case where they are not present in the XLZ header. The Source and Target languages have an impact on the default fonts used for source and target segments in the generated RTF file.

For translated segments, the generated RTF will be bilingual and the match percentage will be correctly set in the RTF. For untranslated segments, the generated RTF will be unsegmented.

Note: The filter recognize segmented locked segments. If, in the XLIFF file the TU is translated and set as translate=no (locked), it will be set to locked style in the RTF file.