OmegaT

The Simple Way of Translation

Andre Hagestedt

OmegaT is a computer-assisted translation tool (CAT) and can translate multiple files in multiple file formats simultaneously, and consult multiple translation memories, glossaries and dictionaries.

WorldWide Translations

Andre Hagestedt

Tokyo, Japan

email@andrehagestedt.com

3/22/2015

Content

General Information about OmegaT

1. Short Description

OmegaT (see details on Wikipedia) is a computer-assisted translation tool (CAT) written in the Java programming language. The free software, originally developed by Keith Godfrey in 2000, and currently developed by a team led by Didier Briel, is intended for professional translators. OmegaT is used 1/3 as much as Wordfast, Déjà Vu and MemoQ, and 1/8 as much as the market leader Trados.

OmegaT has many features including the following:

- Fuzzy matching
- Match propagation
- Simultaneous processing of multiple-file projects
- Simultaneous use of multiple translation memories
- User glossaries with recognition of inflected forms
- Document file formats include all major formats (see details on Wikipedia)
- Unicode (UTF-8) support: can be used with non-Latin alphabets
- Support for right-to-left languages
- Integral spelling checker & Tag validation
- Editor supports undo, redo, copy and paste, switching between uppercase and lowercase, etc. Extensive search & replace capabilities including regular expressions.
- Editable segmentation rules
- Compatible with other translation memory applications (TMX, TTX, TXML, XLIFF, SDLXLIFF)
- Interface to Google Translate (API v2) and other machine translation (MT) engines
- Team Project Feature (not covered in this document)

2. Details for the OmegaT setup used

a. Name: OmegaT
b. Version used: 3.1.8 Beta
c. Supported OS: Windows, Mac OS X, Linux
d. Plugin: Okapi Filters Plugin for OmegaT
e. Plugin: External Finder Plugin
f. MT engine used: Google Translate API v2 (Paid Service)

3. Features of OmegaT in detail

OmegaT shares many features with mainstream CAT tools. These include creating, importing and exporting translation memories, fuzzy matching from translation memories, glossary look-up, and reference and concordance searching.

OmegaT also has additional features that are not always available in other CAT tools. These include:

- OmegaT can translate multiple files in multiple file formats simultaneously, and consult multiple translation memories, glossaries and dictionaries.
- With regard to supported file types, OmegaT allows the user to customize file extensions and file encodings. For a number of document types, the user can choose selectively which elements must be translated (e.g. in OpenOffice.org Writer files, choose whether to include bookmarks; in Microsoft Office 2007/2010 files, choose whether to translate footnotes; or in HTML, choose whether to translate ALT text for images). The user can also choose how non-standard elements in third-party translation memories should be handled.
- OmegaT's segmentation rules are based on regular expressions. Segmentation can be configured based on language or based on file format, and successive segmentation rules inherit values from each other.
- In the Edit window, the user can jump directly to the next untranslated segment, or go forward or backwards in history. Users can use undo and redo, copy and paste, and switch between uppercase and lowercase in the same way as one would in an advanced text editor. The user can choose to see the source text of segments that have already been translated. The edit pane also has inline spell-checking using Hunspell dictionaries, and interactive spell-checking is done using the mouse.
- Users can insert fuzzy matches using a keyboard shortcut or using the mouse. OmegaT shows the degree of similarity in fuzzy matches using colours. OmegaT can also display the date, time and the name of the user who translated any given segment. Glossary matches can be inserted using the mouse. The user can choose to have the source text copied into the target text field, or to have the highest fuzzy match automatically inserted.
- In the Search (or Search & Replace) window, the user can choose to search the current files' source text, target text, other translation memories, and reference files. Searches can be case sensitive, and regular expressions can also be used. Double-clicking a search result takes the user directly to that segment in the edit window.
- After translation, OmegaT can perform tag validation to ensure that there are no accidental tag errors. OmegaT can calculate statistics for the project files and translation memories before the project starts, or during the translation to show the progress of the translation job.
- OmegaT can get machine translations from Apertium, Belazar and Google Translate, and display it in a separate window.
- When OmegaT starts, a short tutorial called "Instant Start" is displayed.
- The font and font size for the Editor, Match and Glossary Viewer windows can be selected via the "Font..." dialog under the Options menu item. See also "Language Character Sets and Fonts".

4. Translation memories in TMX format

OmegaT's internal translation memory format is not visible to the user, but every time it autosaves the translation project, all new or updated translation units are automatically exported and added to three external TMX memories: a native OmegaT TMX, a level 1 TMX and a level 2 TMX.

- The native TMX file is for use in OmegaT projects.
- The level 1 TMX file preserves textual information and can be used with TMX level 1 and 2 supporting CAT tools.
- The level 2 file preserves textual information as well as inline tag information and can be used with TMX level 2 supporting CAT tools.

Exported level 2 files include OmegaT's internal tags encapsulated in TMX tags which allows such TMX files to generate matches in TMX level 2 supporting CAT tools. Tests have been positive in Trados and SDLX.

OmegaT can import TMX files up to version 1.4b level 1 as well as level 2. Level 2 files imported in OmegaT will generate matches of the same level since OmegaT converts the TMX level 2 tags of the foreign TMX. Here again, tests have been positive with TMX files created by Transit.

5. Glossaries

For glossaries, OmegaT mainly uses tab-delimited plain text files in UTF-8 encoding with the .txt extension. The structure of a glossary file is extremely simple: the first column contains the source language word, the second column contains the corresponding target language words, the third column (optional) can contain anything including comments on context etc. Such glossaries can easily be created in a text editor.

Similarly structured files in standard CSV format are also supported, as well as TBX files.

One valuable addition for your glossary folder might be the Microsoft Terminology Collection. Just download the glossary of your language and place it into the glossary folder of your OmegaT project.

Preparing OmegaT

In the following we describe preparing OmegaT in the way we found it most useful.

1. **Installing OmegaT**

   Installing OmegaT is straight forward. The downloadable version can be found here and the installation is quick and easy.

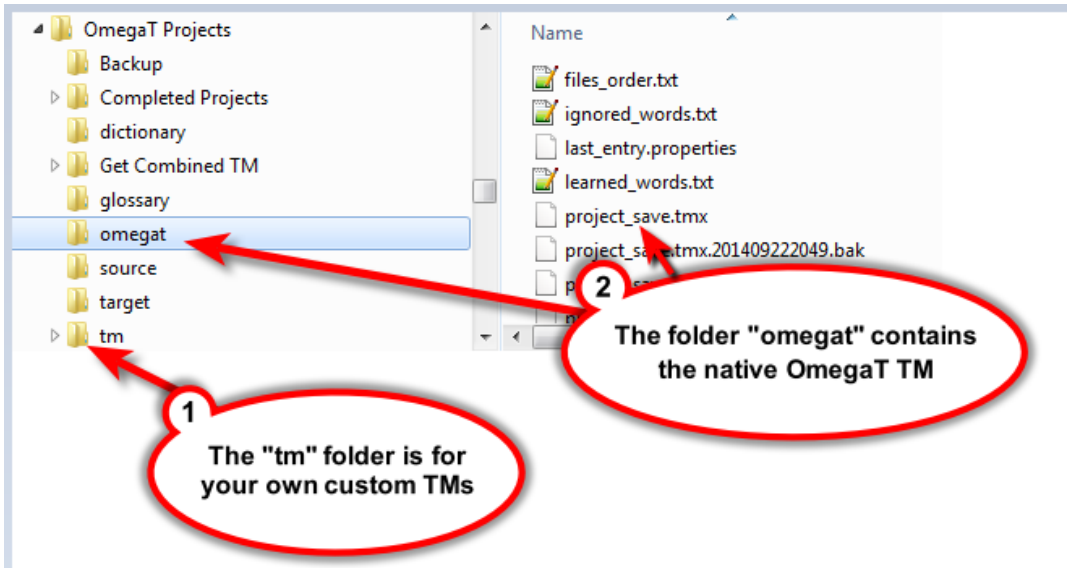
2. **Creating a Project**

   This is straight forward as well and only requires to choose a location and a project name. OmegaT will create all necessary folders (e.g. source, target, TM, etc.) and project files. Once the project is created the project source files can be added. Later, one can always add more source files to the project or remove source files.

3. **Creating a Translation Memory**

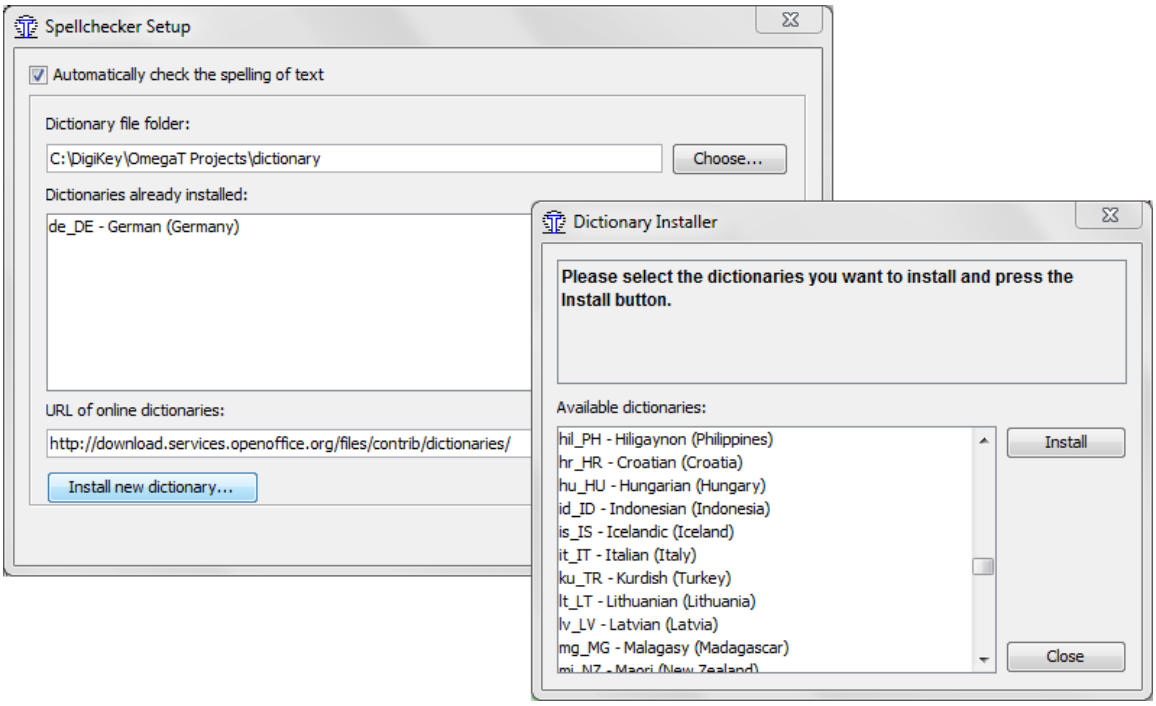
   In order to work with OmegaT no translation memory (TM) needs to be created, a native TM will be created on the fly while translating the project. However, to have leverage from translations you already did before, you can save your own TM (TMX file) in the **tm** folder of your OmegaT project folder. If you have multiple TMs for different projects you can merge them together into one Global TM using TMX Merger, which is a free Java command-line script created by Henry Pijffers for merging two or more TMX files.

   Merging TMs is not absolutely necessary since OmegaT supports using multiple TMs simultaneously but it seemed to be a good idea to us to avoid having too many TMs and to avoid loading time (it might require more system resources in terms of RAM though).



4. Spelling Checker

OmegaT has an integrated spelling checker. To install the appropriate dictionary for your language you have to open the “Spell Checking” window under “Options”, click on “Install new dictionary” and choose your dictionary (you will have to have an internet connection so that OmegaT can connect to the dictionary repository and this will take a few seconds):



Note: Unfortunately there are some languages for which there is no spell checking (e.g. Japanese, Chinese).

5. Creating a Glossary

As for the TM there is no need to create a glossary to start working in OmegaT. As soon as you add the first term the glossary will be automatically created. Adding new entries in OmegaT is extremely easy by just selecting a term, doing a right-click on it, choosing "Add glossary entry" from the pop-up menu and then providing the translation and comments for it. But if a glossary is already available it would of course be useful to use it in OmegaT as well. If you need to convert it just use the details provided here.

6. Installing the Okapi Filters Plugin

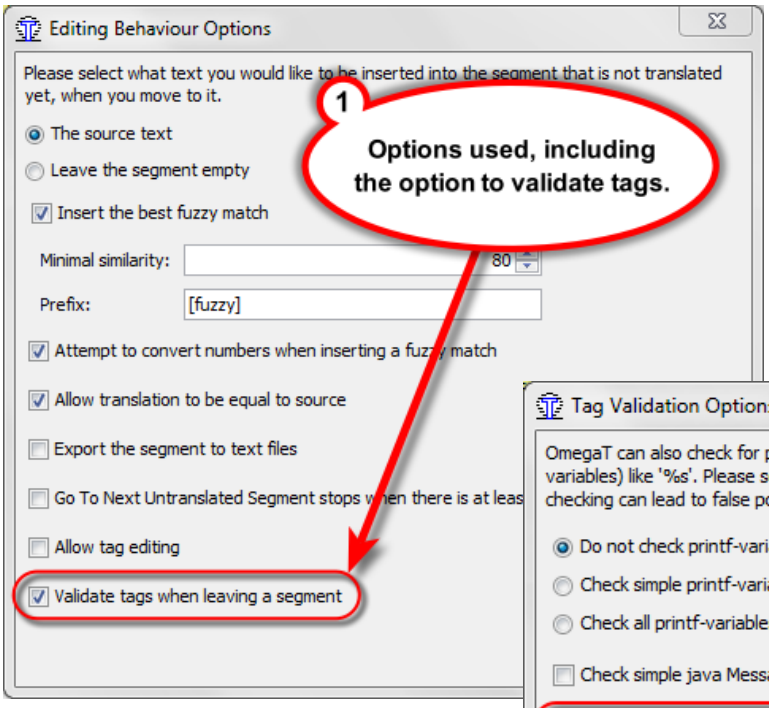
This step is necessary to be able to translate TTX files and other file formats (Documentation, Download) which OmegaT does not support. Here is a list of supported formats:

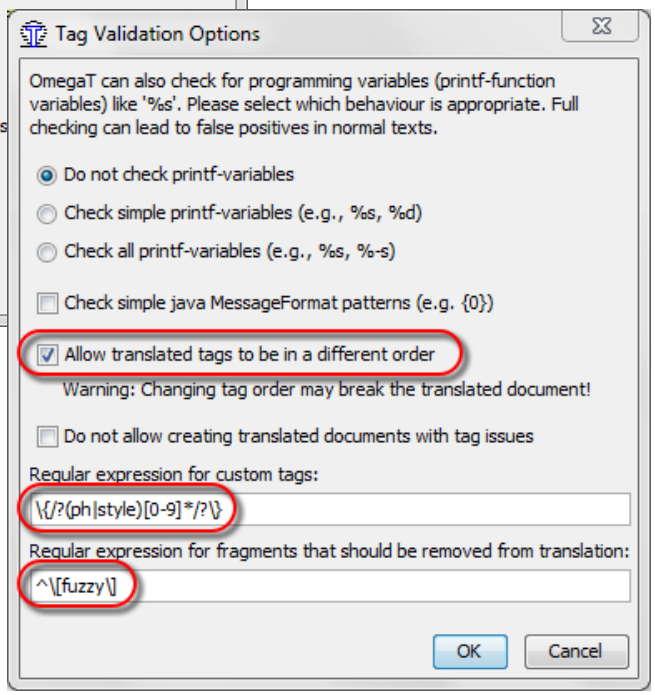
- InDesign IDML files (using the IDML Filter)
- JSON files (using the JSON Filter)

- Qt TS files (using the TS Filter)
- Trados TagEditor TTX files (using the TTX Filter)
- Transifex projects (using the Transifex Filter)
- Wordfast Pro TXML files (using the TXML Filter)
- XLIFF 1.2 documents (using the XLIFF Filter – Note: XLIFF is already supported by OmegaT but there seems to be a problem. The Okapi XLIFF filter worked without any problems!)

7. Turning on Tag Validation

To make sure tags are not corrupted the option for tag validation (in the **Editing Behavior** window under the Options menu) should be turned on. This would be enough for standard file types with standard tags but some documents define their own tags so these tags need to be declared in OmegaT (in the **Tag Validation** window) as custom tags. This is done using regular expressions. The tool I use to create and test my own regular expressions is The Regex Coach.





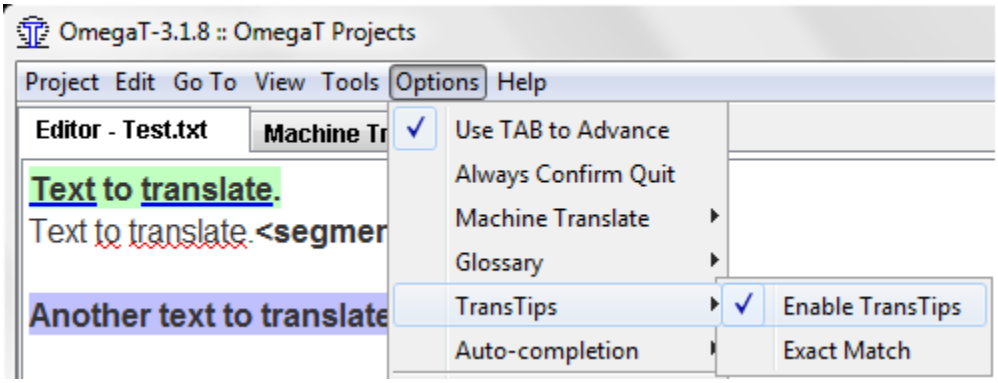
The screenshot on the right shows two regular expressions. The first one is to support custom tags and the second one is to remove the [fuzzy] tag which is used by OmegaT to mark fuzzy matches:

^\[fuzzy\]

The box to allow tags to be in a different order should also be checked since tags often need to be moved due to the fact that word orders are different between source and target language.

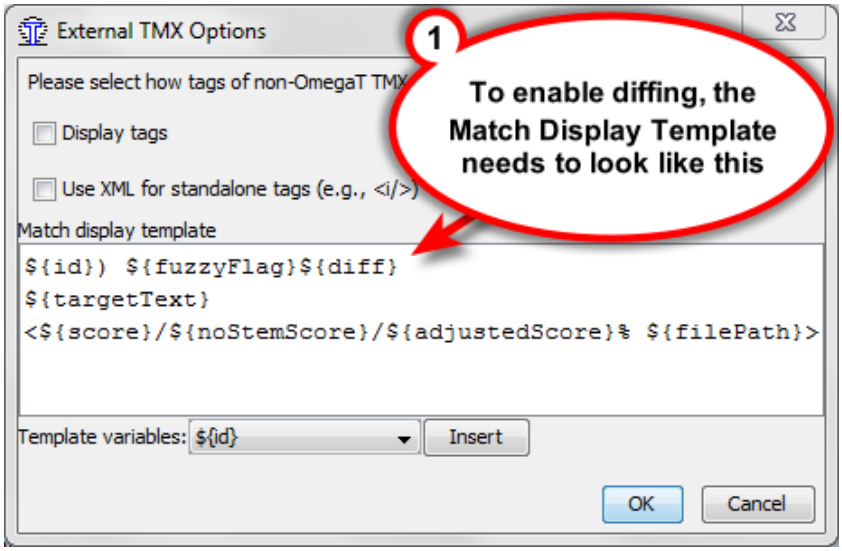
8. Enabling TransTips

To make full use of the glossary the option **TransTips** should be enabled. When glossary entries are identified in the source segments, they will now be underlined in blue. A right click on the underlined words opens a popup, containing the glossary entry or entries, which can then be inserted in the target segment.



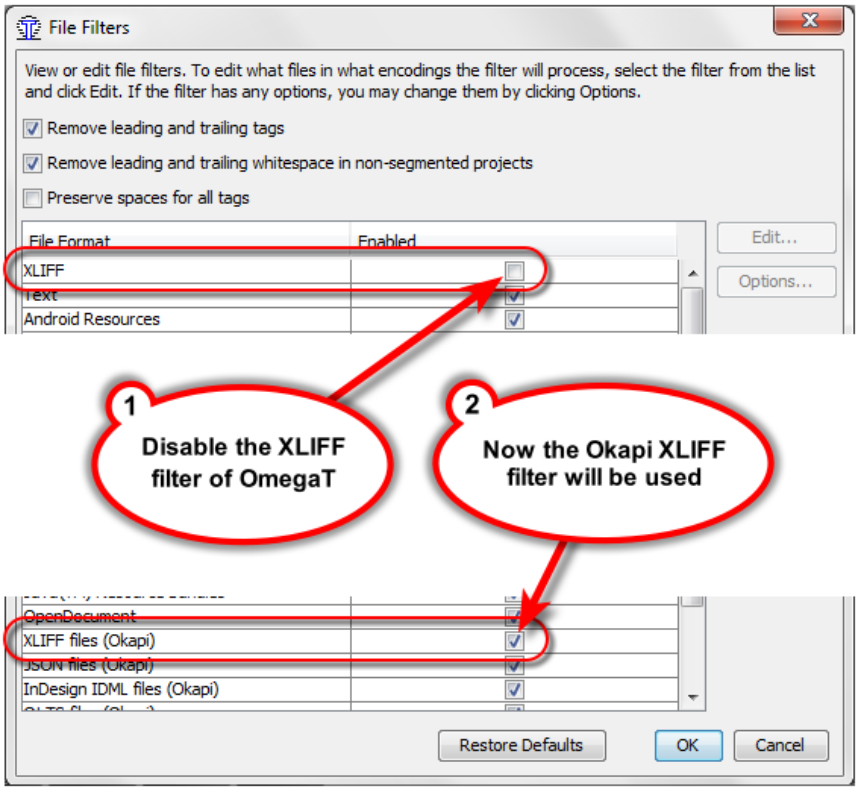
9. Enabling "Diffing"

To get more information about the differences between the source and the fuzzy matches found by OmegaT the diffing option should be enabled by changing the **Match display template** in the dialog **External TMXs..** under the **Options** menu to look like the following:



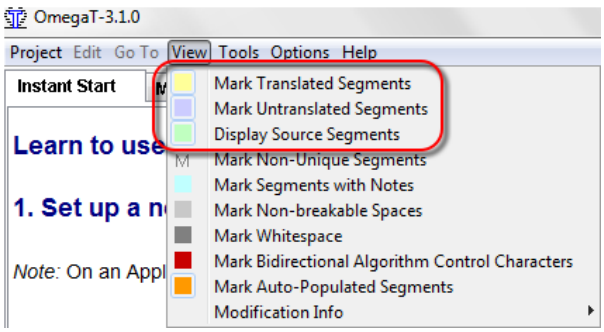
10. Modifying the File Filters

For some reason the XLIFF filter that is implemented in OmegaT did not work very well for us so we disabled it to make OmegaT use the Okapi XLIFF filter instead (in order for this to work the Okapi Filter Plugin has to be installed). This made all issues go away so if you have any problems we recommend to use this workaround.



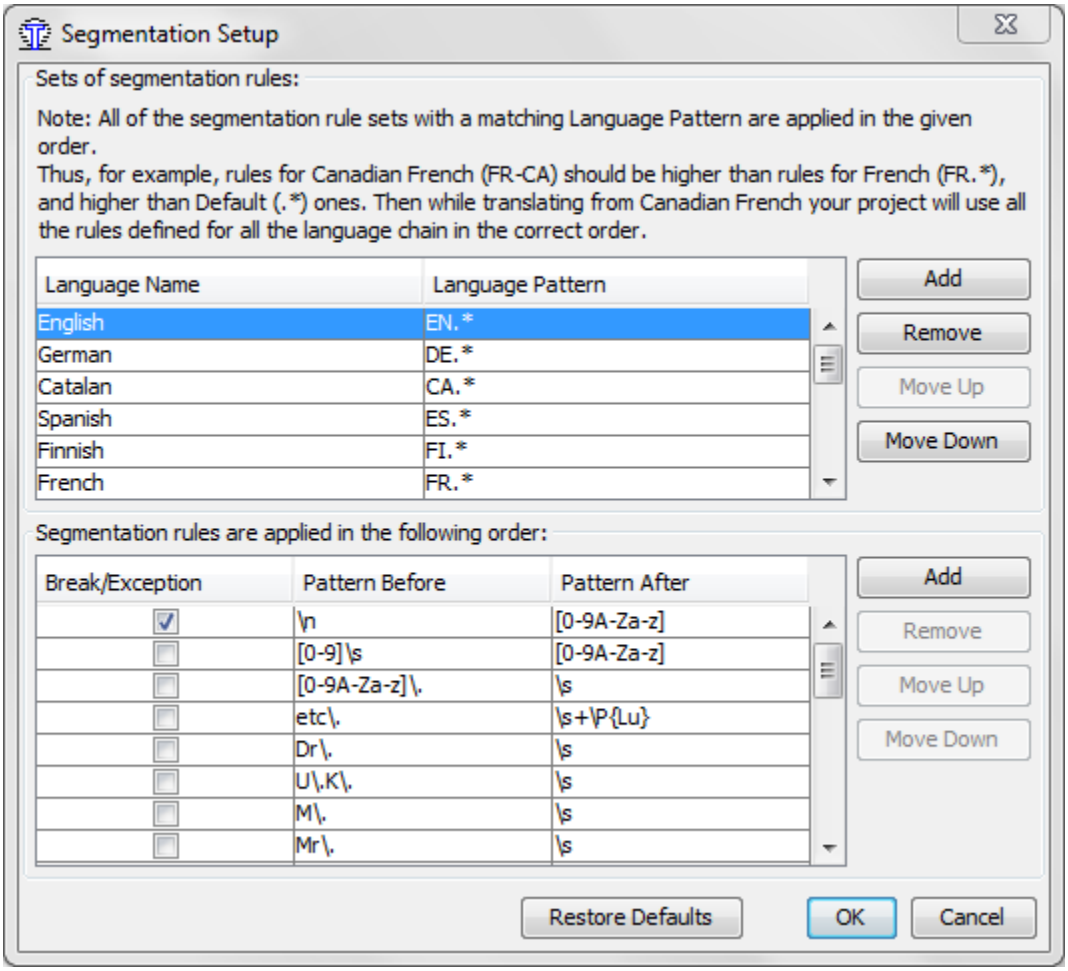
11. Changing the View

To work comfortable we changed the view by checking the 3 options "Mark Translated Segments", "Mark Untranslated Segments" and "Display Source Segments" under the menu "**View**". This is of course a matter of preference so just take this as a suggestion and play around with the options until you find the view you are most comfortable with.



12. Modifying the Segmentation Rules

In the dialog **Segmentation Setup** (Options >> Segmentation…) you should move your source language to the top of the list and add the first 3 segmentation rules as shown in the following screenshot.



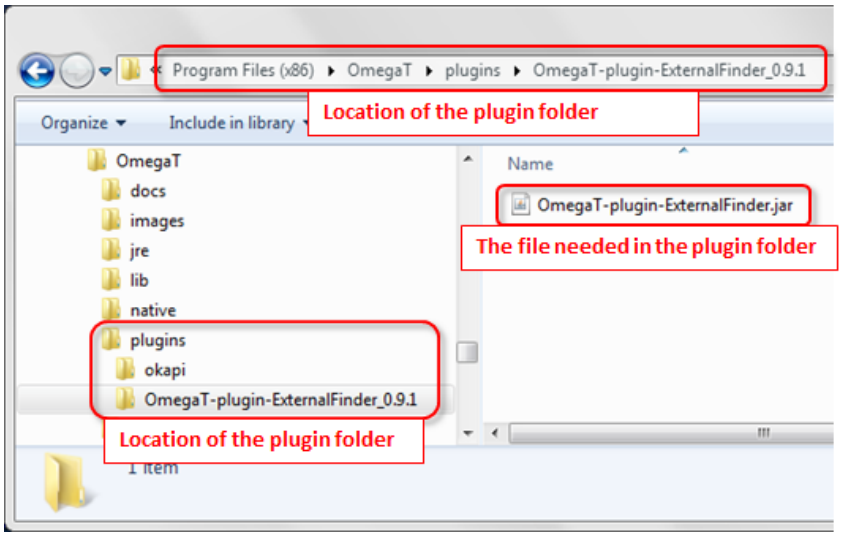
In case you want to copy and paste the three segmentation rules I created a little table below for you. Make sure that they will be the first three rules (using "Move Up").

Break/Exception	Pattern Before	Pattern After
☑	\n	[0-9A-Za-z]
☐	[0-9]\s	[0-9A-Za-z]
☐	[0-9A-Za-z]\.	\s

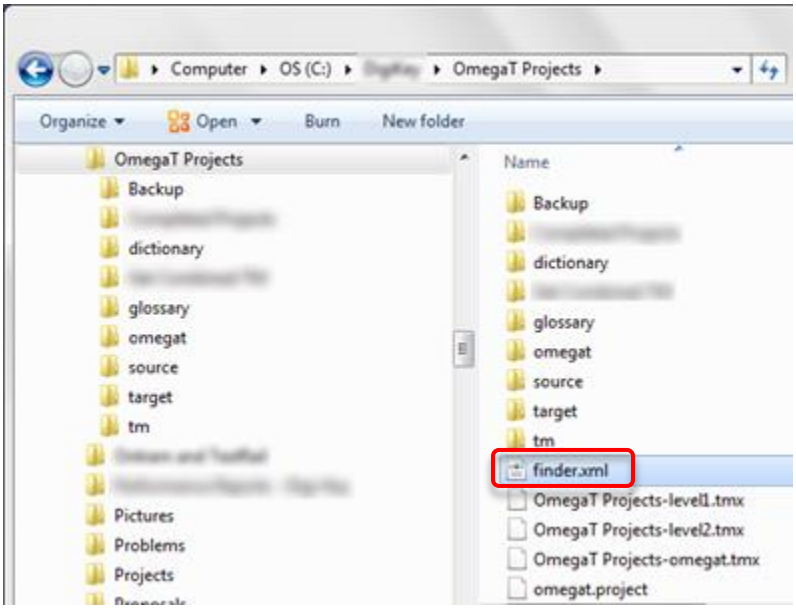
13. Installing the External Finder Plugin

The External Finder Plugin for OmegaT is extremely useful if you need to look up a term in a web-based dictionary, find a definition for it or if you need any other information about that term using a web-based resource.

The plugin comes in a ZIP file and there are several files but only one, the **JAR** file (see screenshot below), needs to be copied to the OmegaT **plugins** folder. However, you can just copy the entire folder to the plugins folder if you prefer that.



After "installing" the plugin you need to populate the **Tools** menu and the context menu (available through a Right-Click on a selected term) with your **External Links**. To do this you need to modify the **finder.xml** file and place it into the root folder of your project (see screenshot below).



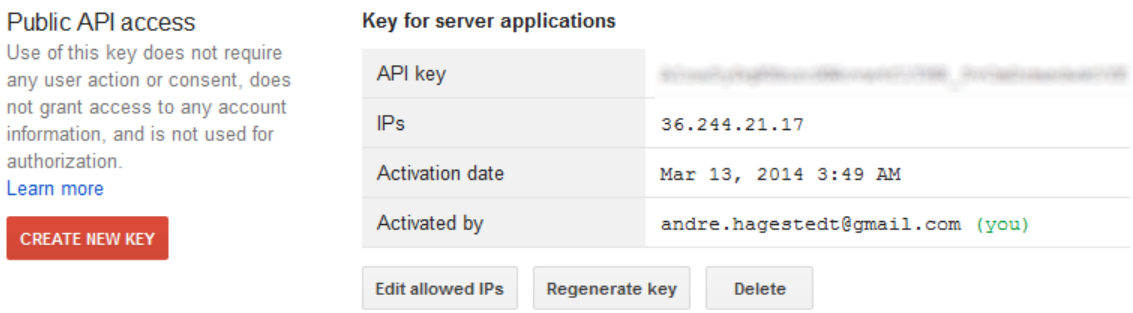
14. Using Google Translate API v2 for Machine Translation

If you want to do this you need to be aware that the **Google Translate API v2** is a **paid service**. Google Translate API pricing is based on usage with $20 per 1 Million characters of text. The charges are adjusted in proportion to the number of characters actually provided. For example, if you were to translate 500k characters, you would be billed $10.

A Google account is required and after providing billing information and creating a Google Translate project you need to have Google create an **API key** for you. You then need to add this API key to the **OmegaT.l4J.ini** file. A detailed procedure is provided on the OmegaT website. However, this procedure looks a bit intimidating and lacks a little bit of clarity here and there (e.g. on how you get the correct **IPv4 address**).

Tip: As mentioned above, getting the correct IPv4 address might be difficult. Using **ipconfig** on the command line does NOT provide you with the correct IP. The easiest way to get the correct IP is to "**Ping**" yourself with an online tool like this.

The following screenshot shows which key (Public API access) you need to create for OmegaT. The key consists of 39 characters and is blurred out in the screenshot below.



In the **OmegaT.l4J.ini** file you need then to replace the placeholder **XXXXXXX** in the following section of the **OmegaT.l4J.ini** file with the API key:

```
# Google Translate v2 API key
-Dgoogle.api.key=XXXXXXX
```

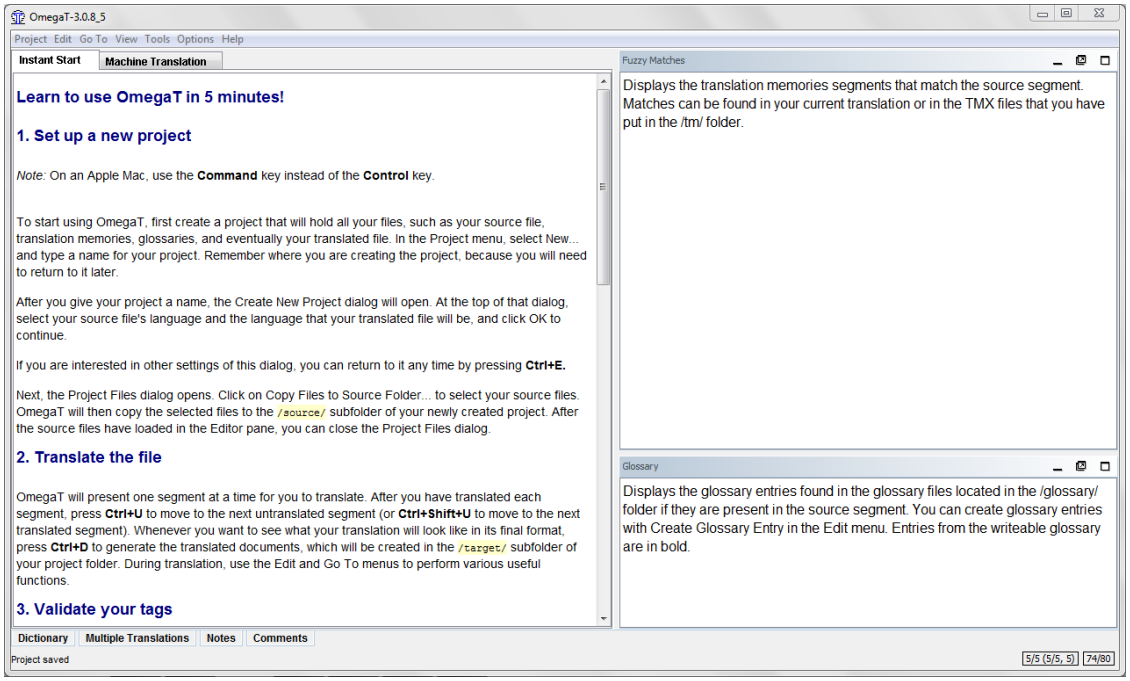
Conclusion

All together we highly recommend the use of **OmegaT,** as set up in this report, to everyone who does not want to spend a lot of money for a paid solution. We even recommend OmegaT for those who do own a professional CAT tool because we found it very pleasant to work with OmegaT due to its simplicity. It provides everything needed and does not feel overwhelming due to complexity, hundreds of functions which nobody ever needs, or a chaotic user interface.

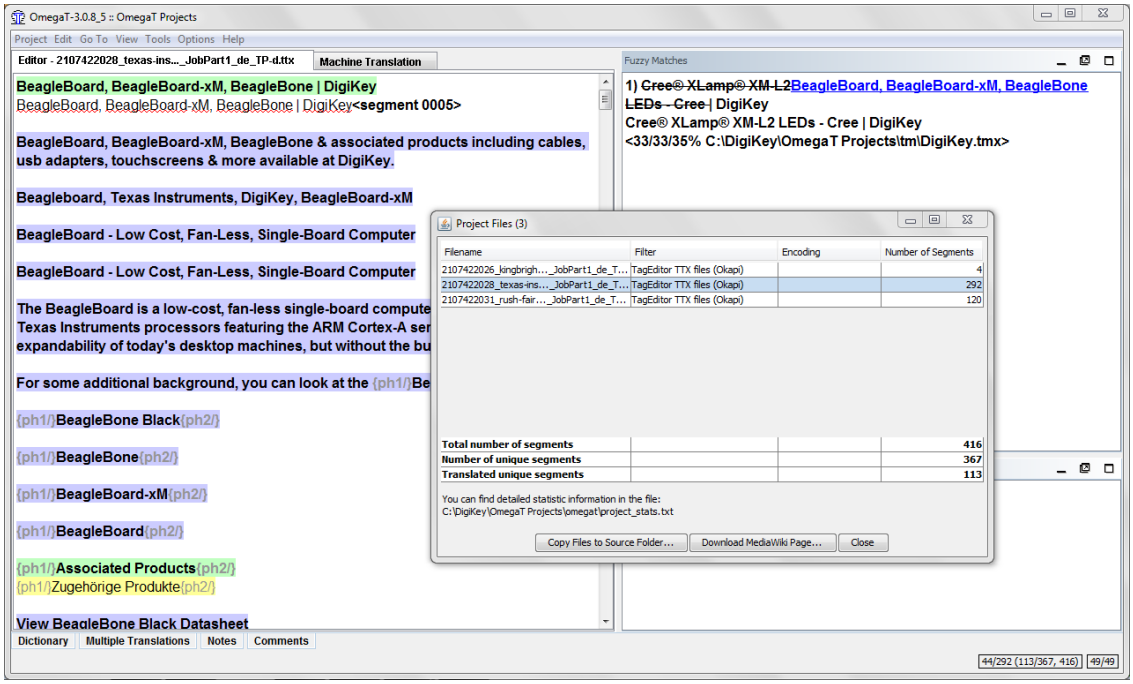
We do also recommend OmegaT because of its extensive editing and search & replace capabilities, the instant TM leverage, the immediate translation propagation, the possibility of modifying the segmentation rules, the glossary features, the interfaces to a variety of machine translation services, etc. It even offers a Team Project features but this was not tested by us so we can't say anything about how it needs to be set up and how well it works.

Screenshots

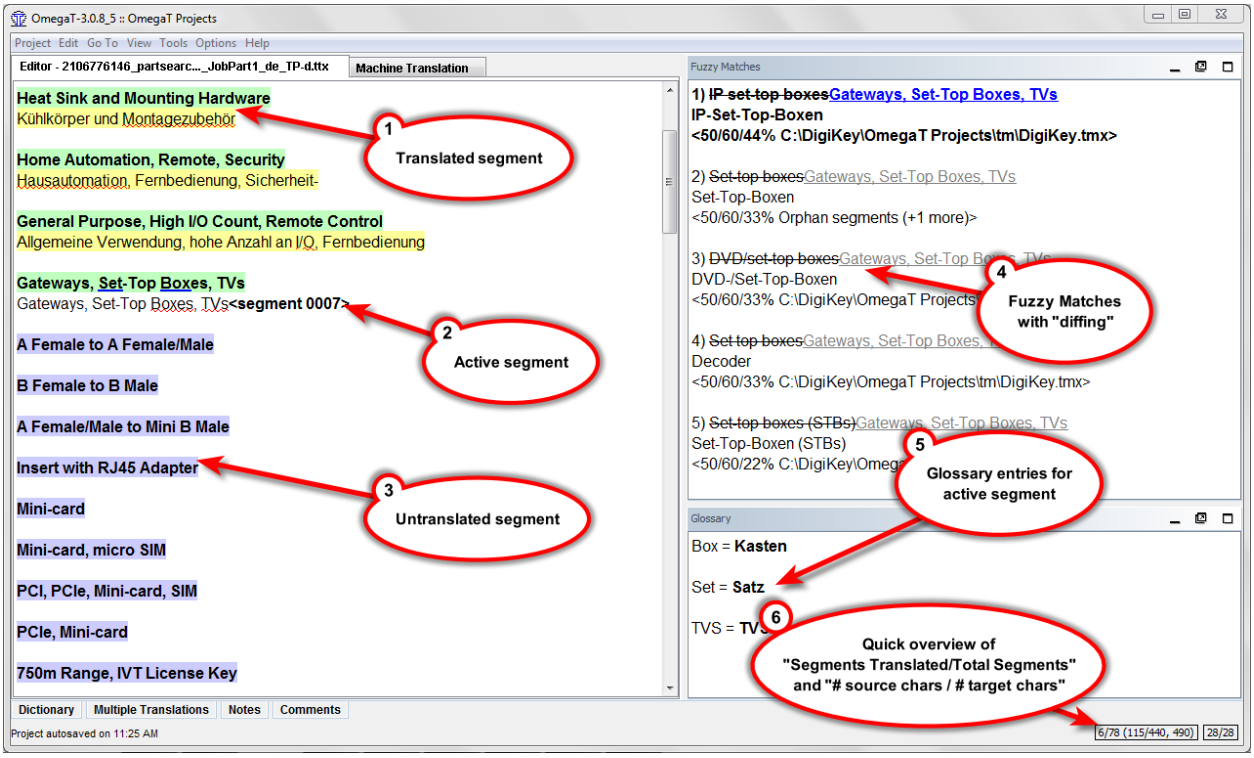
Start Window with “How To” instructions to quickly learn how to use OmegaT:



OmegaT Window and Project Dialog:


OmegaT-3.0.8_5 :: OmegaT Projects
Project Edit Go To View Tools Options Help
Editor - 2107422028_texas-ins..._JobPart1_de_TP-d.ttx
Machine Translation
BeagleBoard, BeagleBoard-xM, BeagleBone | DigiKey
BeagleBoard, BeagleBoard-xM, BeagleBone | DigiKey<segment 0005>
BeagleBoard, BeagleBoard-xM, BeagleBone & associated products including cables, usb adapters, touchscreens & more available at DigiKey.
Beagleboard, Texas Instruments, DigiKey, BeagleBoard-xM
BeagleBoard - Low Cost, Fan-Less, Single-Board Computer
BeagleBoard - Low Cost, Fan-Less, Single-Board Computer
The BeagleBoard is a low-cost, fan-less single-board compute
Texas Instruments processors featuring the ARM Cortex-A ser
expandability of today's desktop machines, but without the bu
For some additional background, you can look at the {ph1/}Be
{ph1/}BeagleBone Black{ph2/}
{ph1/}BeagleBone{ph2/}
{ph1/}BeagleBoard-xM{ph2/}
{ph1/}BeagleBoard{ph2/}
{ph1/}Associated Products{ph2/}
{ph1/}Zugehörige Produkte{ph2/}
View BeagleBone Black Datasheet
Fuzzy Matches
1) ~~Cree® XLamp® XM-L2~~BeagleBoard, BeagleBoard-xM, BeagleBone ~~LEDs - Cree~~ | DigiKey
Cree® XLamp® XM-L2 LEDs - Cree | DigiKey
<33/33/35% C:\DigiKey\OmegaT Projects\tm\DigiKey.tmx>
Project Files (3)
Filename
Filter
Encoding
Number of Segments
2107422026_kingbrigh..._JobPart1_de_T... TagEditor TTX files (Okapi) 4
2107422028_texas-ins..._JobPart1_de_T... TagEditor TTX files (Okapi) 292
2107422031_rush-fair..._JobPart1_de_T... TagEditor TTX files (Okapi) 120
Total number of segments 416
Number of unique segments 367
Translated unique segments 113
You can find detailed statistic information in the file:
C:\DigiKey\OmegaT Projects\omegat\project_stats.txt
Copy Files to Source Folder...
Download MediaWiki Page...
Close
Dictionary
Multiple Translations
Notes
Comments
44/292 (113/367, 416)
49/49

OmegaT Window with some explanations:


OmegaT-3.0.8_5 :: OmegaT Projects
Project Edit Go To View Tools Options Help
Editor - 2106776146_partsearc..._JobPart1_de_TP-d.ttx
Machine Translation
Heat Sink and Mounting Hardware
Kühlkörper und Montagezubehör
Home Automation, Remote, Security
Hausautomation, Fernbedienung, Sicherheit-
General Purpose, High I/O Count, Remote Control
Allgemeine Verwendung, hohe Anzahl an I/O, Fernbedienung
Gateways, Set-Top Boxes, TVs
Gateways, Set-Top Boxes, TVs<segment 0007>
A Female to A Female/Male
B Female to B Male
A Female/Male to Mini B Male
Insert with RJ45 Adapter
Mini-card
Mini-card, micro SIM
PCI, PCIe, Mini-card, SIM
PCIe, Mini-card
750m Range, IVT License Key
1
Translated segment
2
Active segment
3
Untranslated segment
Fuzzy Matches
1) IP-set-top boxesGateways, Set-Top Boxes, TVs
IP-Set-Top-Boxen
<50/60/44% C:\DigiKey\OmegaT Projects\tm\DigiKey.tmx>
2) Set-top boxesGateways, Set-Top Boxes, TVs
Set-Top-Boxen
<50/60/33% Orphan segments (+1 more)>
3) DVD/set-top boxesGateways, Set-Top Boxes, TVs
DVD-/Set-Top-Boxen
<50/60/33% C:\DigiKey\OmegaT Projects\
4) Set-top boxesGateways, Set-Top Boxes, TVs
Decoder
<50/60/33% C:\DigiKey\OmegaT Projects\tm\DigiKey.tmx>
5) Set-top boxes (STBs)Gateways, Set-Top Boxes, TVs
Set-Top-Boxen (STBs)
<50/60/22% C:\DigiKey\Omega
4
Fuzzy Matches with "diffing"
5
Glossary entries for active segment
Glossary
Box = Kasten
Set = Satz
TVS = TVS
6
Quick overview of "Segments Translated/Total Segments" and "# source chars / # target chars"
Dictionary
Multiple Translations
Notes
Comments
Project autosaved on 11:25 AM
6/78 (115/440, 490)
28/28