

Localizing MemoQ

Madeleine Lenker
PhD Student at LRC/UL

1

Localization is...

"The aspect of development and testing relating to the translation of the software and its presentation to the end user. This includes translating the program, choosing appropriate icons and graphics, and other cultural considerations. It also may include translating the program's help files and the documentation. You could think of localization as pertaining to the presentation of your program; the things the user sees."

M Rätzmann, C. De Young: Galileo Computing, Software Testing and Internationalization, chapter 6
Internationalization and Localization Testing, Galileo Press GmbH, Bonn 2002, p. 240

Content

- Evaluation of approaches
- Awareness for using a CAT tool in translating software
- Testing strategies
- Feedback loops

Translating Software...

When translating software, usually, a "software localization" tool as SDL Passolo or Alchemy Catalyst is used.

The advantages of a SW localization tool are:

- Import of .dll, .exe files possible
- WYSIWYG editing/translation of dialog boxes, menus, etc.
- Immediate resizing during translation (if required)
- Validation checks e.g. for ampersands, double hotkeys

How does the localization of MemoQ work?

Approach:

Using MemoQ to localize MemoQ

Some aspects to be aware of....

- no preview
- no hotkey check

5

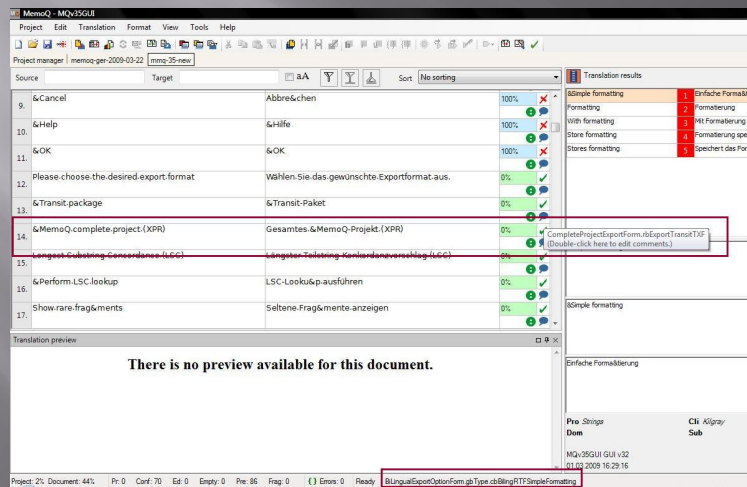
Translation itself...

The strings are delivered as MBD (MemoQ Bilingual Document) files. When extracted they have an "ID context".

Therefore, it is possible to know where the dialog is in the software, how many strings belong to the dialog, etc.

6

Translation itself...

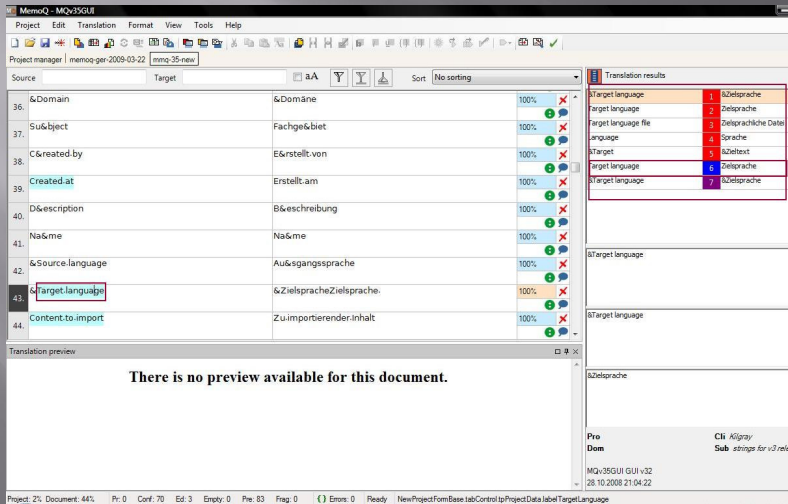


Creating a Term base...

Attaching a term base (or more than one if desired) to a project is essential since MemoQ displays the terms within the segment while translating.

In this way you can always see if a term is in the term base or not. This provides consistency during translation, also later on when you translate string updates.

Creating a Term base...



Source	Target	Progress
36. &Domain	&Domäne	100%
37. S&ubject	Fachge&biet	100%
38. C&reated by	E&stellt: von	100%
39. C&reated at	Erstellt: am	100%
40. D&escription	B&eschreibung	100%
41. N&a&me	N&a&me	100%
42. &Source: language	Auf&sgangssprache	100%
43. &Target: language	&ZielspracheZielsprache:	100%
44. C&ontent: to: import	Zu importierender Inhalt	100%

Translation results

1	&Zielsprache
2	&Zielsprache
3	&Zielsprache: Datei
4	&Zielsprache
5	&Zielsprache
6	&Zielsprache
7	&Zielsprache

There is no preview available for this document.

What happens after the translation?

Assuming we have done all our checks (e.g. spell checks etc.), the translated MBD file will be processed again

A build comes back and the Quality Assurance (QA) stage begins, in this case linguistic QA as well as layout. If a functional issue is detected, it is logged in Mantis, Kilgrays Issue Tracking System.

Linguistic QA means...

- Checking dialog boxes
- Check for concatenated strings
- Check for truncated strings
- Check for missing hot keys and duplicate hot keys by displaying them
- Layout checks

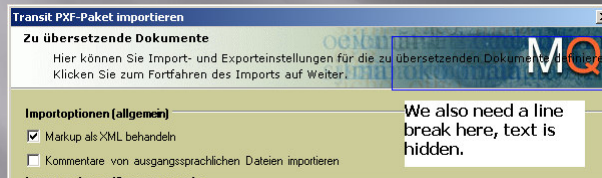
11

Reporting issues...

- Commented screenshots for faster identification of the issue
- Excel datasheet, including translators and developers comments (sometimes dialog box resizing is not possible, so the translation has to be altered/shortened)
 - Such datasheets help to keep track of issues being resolved or still pending
 - Issue Tracking Systems such as Mantis log bugs according to categories, severity, etc.

12

How does it look like...



In this case, a line break was not possible, so the translation had to be shortened.

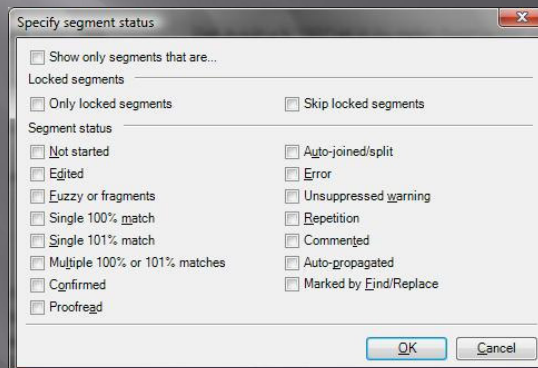
13

String Updates...

There is a feature called Views. Using Views, you are only presented with the updated strings instead of all. Views are basically an extracted part of a file.

Sometimes, it is good to have all strings available when you need to see the whole context or have to change a term on a global level. This facility is available due to MemoQ's convenient filter options.

Views can easily inserted back into the file containing all of the strings.



Lessons we learnt...

- Reserve more space for dialog text, keeping localization in mind when creating new dialog windows, etc.
- Educate developers not to hard-code text
- When designing new features, give some thought to how to name them – think of the poor translators who have to come up with a great translation
- Consistency of the source strings
- Issues on the source strings were improved

Achievements...

Fast and easier translation of the strings due to

- more space for dialogs resulting in less resizing issues;
- time reduction in build checking due to developed checking strategies;
- faster translation process due to well maintained TM and TB and good teamwork.

References

<http://msdn.microsoft.com/en-us/goglobal/bb688162.aspx>

Systematic validation of localisation across all languages,
Martin Ørsted, Microsoft Ireland, Martin.Orsted@microsoft.com

M Rätzmann, C. De Young: Galileo Computing, Software Testing and Internationalization, chapter 6 Internationalization and Localization Testing 239 – 304, Galileo Press GmbH, Bonn 2002

Thank you for your
attention!

Partially supported by

