Translator’s Guide

Multilizer® 5.1 – Translator's Guide

December 2002

Table of Contents

1

Conventions used in this book

The following typographical conventions have been used in this book.

Names of windows, menu options and key buttons are printed in **bold Sans serif**.

Texts for figures and references to chapters and sections in this guide are shown in *italic Sans serif*.

MORE INFO

The more info symbol is used when there is additional information available either in this document, in Multilizer application online help or on the MULTILIZER Web pages at `http://www.multilizer.com.`

NOTE!

The note symbol is used to emphasize certain tasks or issues that are of importance in the current topic.

TIP!

The text marked with the Tip symbol gives useful hints, which may simplify tasks described in the current chapter.

CHAR.SET

This symbol is used in describing different character sets. Character sets are one central factor to be taken into consideration when localizing software.

2

Overview

This is the manual of MULTILIZER® Translator Edition™ ("Multilizer"), the free *translation tool* for translating Multilizer software globalization projects.

Multilizer naming conventions

NOTE!

Multilizer has different editions: **Multilizer Enterprise** and **Multilizer for VCL** are for developers or project managers. These editions have full funcionality to scan, translate and build software and from here on in are referred to as **Multilizer**. The free-of-charge edition for the translator is called **Multilizer Translator Edition** – this edition is capable only for translating projects. **Multilizer Translator Edition Pro** is an edition for QA personnel and it can be used for translation, QA work and building software.

About Multilizer

Multilizer is used in companies that develop software with accompanying content, and want to localize it. For example, the company might have created software in English and wants it to work in German. To get the translations done they have sent you a *Localization Kit* (→Localization Kit, p. 3) including the texts to be translated.

In software localization, Multilizer supports multiple different development tools, programming languages, and content formats. However, from the translator's point of view there is no difference, whether he is translating a Java project or C++ project. He always uses Multilizer to complete the translations.

Translator's role

The workflow below shows the tasks of the developers and translators in the context of the Multilizer localization process.

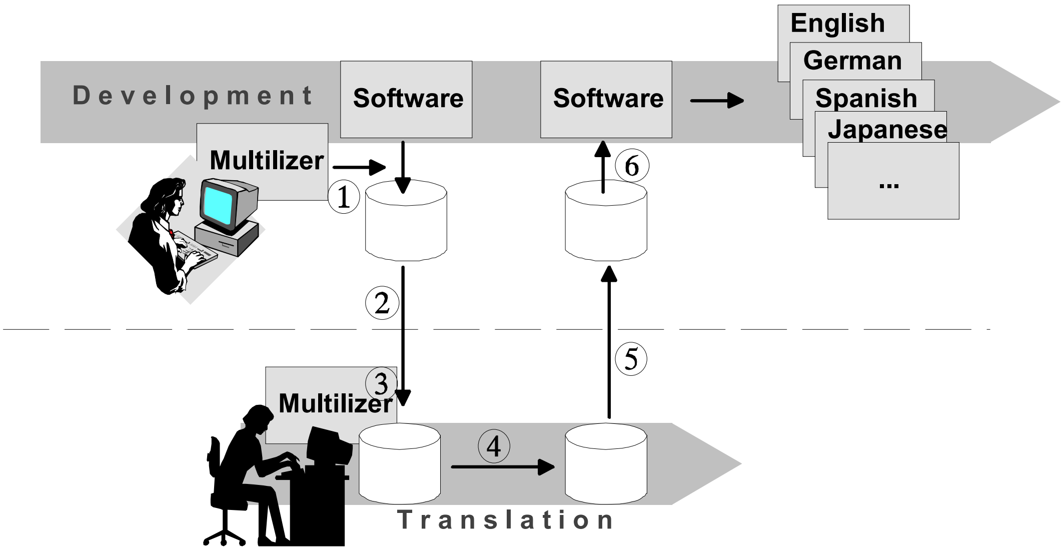


Figure 1. *Multilizer globalization process*

① The developer starts a software localization project by scanning the software and content with Multilizer, and creating a Multilizer project file. ② He creates a Localization Kit that includes the translation tool and the texts to be translated. The Kit is sent to the translator.

③ The translator opens the Localization Kit. It automatically installs Multilizer and the texts to be translated on the translator's computer. ④ The translator uses Multilizer to translate the project contents. ⑤ When finished, the translations are sent back to the developer.

⑥ The developer integrates the translations in the project file. Using Multilizer throughout has preserved the technical context of the translations, so it's very simple to create the localized versions of the software.

Localization Kit

Multilizer Localization Kit – *LocKit* – contains information that is needed in sharing the translation work among localization team members. It is created with Multilizer. It is a package that contains information needed to translate a Multilizer project.

Installation

The LocKit can be installed in either of the following ways:

- A self-extracting LocKit automatically installs a separate *translation tool* and other files on the translator's computer. Use this way only, if you don't have Multilizer® Translator Edition Pro™.
- Open the Multilizer package file (MLP) in Multilizer (File|Open…). The files will automatically be extracted from the LocKit.

Contents

The LocKit contains the following components:

Component	Required
Translation tool	
Multilizer project file.	✓
This manual.	
Other files, such as project related documentation, pictures etc.	

Multilizer translation tool

Multilizer translation tool is used for translating the contents of the *Multilizer project file*. The self-extracting LocKit automatically installs Multilizer as Multilizer® Translator Edition™.

If you have purchased a Multilizer® Translator Edition Pro™ license, LocKit must not contain the translation tool. Remember to inform this to the person that created the LocKit.

Multilizer project file

Multilizer project file is always included in the package. It includes the words and phrases to be translated and empty columns for other languages, which will include the translations done by you. Project file is edited with Multilizer.

After installing the LocKit, the Multilizer project automatically opens in Multilizer.

Multilizer project file can be opened later by simply choosing File|Open in Multilizer. Ensure that Open dialog shows files of type "Project files" – the file extension is .MPR.

Other files

The LocKit can also contain an arbitrary number of other files. They will be located in the same folder as the project file (.MPR).

Multilizer benefits for translators

Multilizer includes many features that support the translation of software. Although projects could be translated outside Multilizer, only Multilizer brings benefits like the following ones:

Simple and secure management of localization projects

- You always get the data in same format: you don't have to learn all the time new formats.
- The data is not corrupted: resource files and other specific formats can easily be corrupted when translating them. Multilizer prevents this from happening, because the files are never edited directly.
- Statistics gives you immediately the whole picture of the project status.

Faster and more accurate translations

- Multilizer always preserves context information. Your translations will always find the correct location in the localized software.
- Translation Memory ensures you translate only once. Later just automatically pre-translate the project by using the Translation Memory's contents.
- Import translation memories and glossaries from various formats; enable efficient re-use of your existing data.
- Translation Memory ensures linguistic consistency.
- If a word or phrase exists multiple times, you translate it just once.
- You can view the picture where the string to be translated is located, so you can check the translation's context.

Better maintainability

- The words and phrases are divided in logical sections (e.g., the texts of one Window view are shown as one section), which help to show the context. In addition, it helps you to complete the translation work systematically.
- For you one project equals one file! Software projects may contain hundreds of files and tens of executables, but you can do the work simply by maintaining one *Multilizer project file*.

3

Multilizer workspace

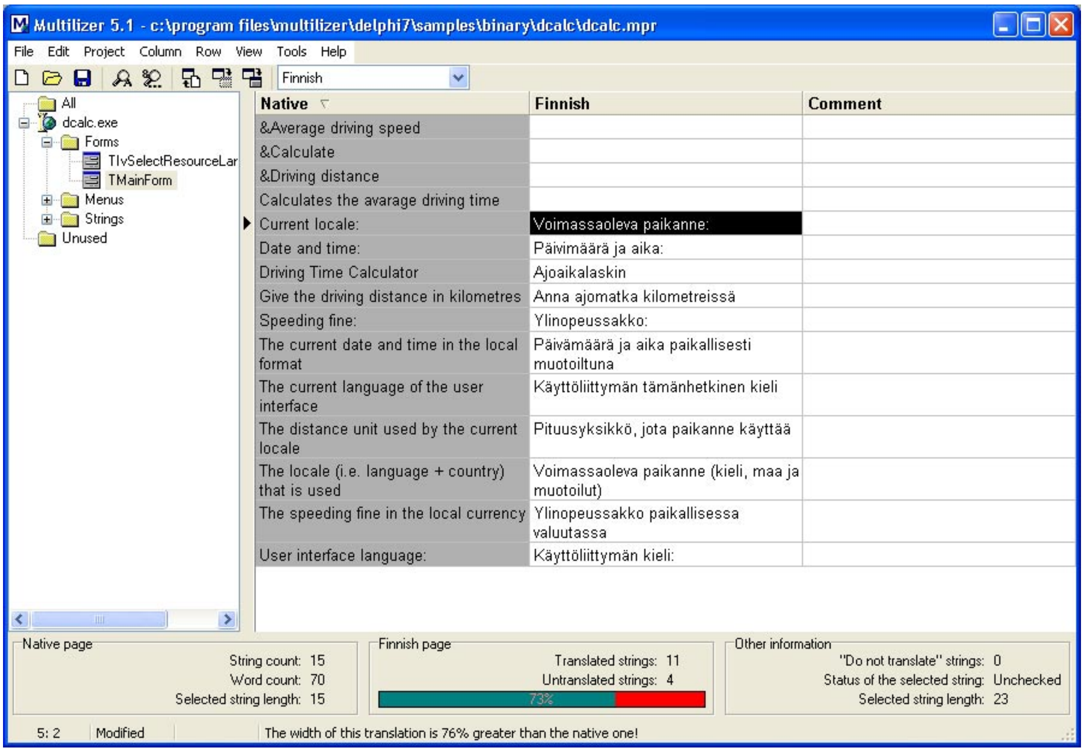
This chapter introduces the Multilizer workspace. Subsequent chapters refer to the different parts of the workplace, with the names introduced in this chapter.

When you start Multilizer and load the Multilizer project, you get into the workspace. It shows the native language along with the languages into which the software is translated.

The native language is the language in which the software was developed. In the following picture the software was developed in English and the target language will be German. If you have to translate the software into more than one language, you can choose the active language from the toolbar.

The central parts of the user interface are the:

- **Tool bar** (below the menus) – toggle on/off: Tools|Tool bar
- **Resource Tree** (at the left side).
- **Translation Grid.**
- **Statistics panel** (below the translation grid) – toggle on/off: Tools|Statistic panel
- **Status bar** (the bottom bar).



Native	Finnish	Comment
&Average driving speed		
&Calculate		
&Driving distance		
Calculates the avarage driving time		
Current locale:	Voimassaoleva paikanne:	
Date and time:	Päivimäärä ja aika:	
Driving Time Calculator	Ajoaikalaskin	
Give the driving distance in kilometres	Anna ajomatka kilometreissä	
Speeding fine:	Ylinopeussakko:	
The current date and time in the local format	Päivämäärä ja aika paikallisesti muotoiltuna	
The current language of the user interface	Käyttöliittymän tämänhetkinen kieli	
The distance unit used by the current locale	Pituusyksikkö, jota paikanne käyttää	
The locale (i.e. language + country) that is used	Voimassaoleva paikanne (kieli, maa ja muotoilut)	
The speeding fine in the local currency	Ylinopeussakko paikallisessa valuutassa	
User interface language:	Käyttöliittymän kieli:	

Native page
String count: 15
Word count: 70
Selected string length: 15

Finnish page
Translated strings: 11
Untranslated strings: 4
73%

Other information
"Do not translate" strings: 0
Status of the selected string: Unchecked
Selected string length: 23

5: 2 | Modified | The width of this translation is 76% greater than the native one!

Figure 2 *Multilizer workplace.*

Toolbar

The toolbar contains a number of buttons. These let you apply the most often needed functions quickly.

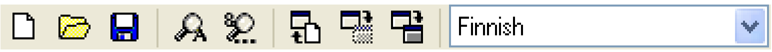


***Figure 3**. Multilizer toolbar*

The buttons are as follows:

Button	Shortcut	Action
📂	**Ctrl+O**	Open a Multilizer project. Corresponds to **File | Open...**
💾	**Ctrl+S**	Saves current Multilizer project. Corresponds to **File | Save**.
🔍	**Ctrl+F**	Opens Find dialog box, where you can specify a string to be searched in the project. Corresponds to **Edit | Find...** Choose **Edit | Find Again** to do the same search again. F3 does the same.
🔍	**Ctrl+I**	Moves to the next incomplete translation. Corresponds to **Edit | Next Incomplete Item**.
		Rescans the project. This feature is disabled in the Multilizer Translator Edition and in Multilizer Translator Edition Pro.
		Makes localized versions of the original software. This feature is disabled in Multilizer Translator Edition.
		Builds localized versions of the original software. This feature is disabled in Multilizer Translator Edition.

In addition to the buttons, you can use the drop-down list box to switch the target language.

Resource Tree

Resource tree the files that are included in the project. Each node below the root is called *target* and corresponds to one of the following:

- One software project
- One executable
- One source code file
- One database
- One content file

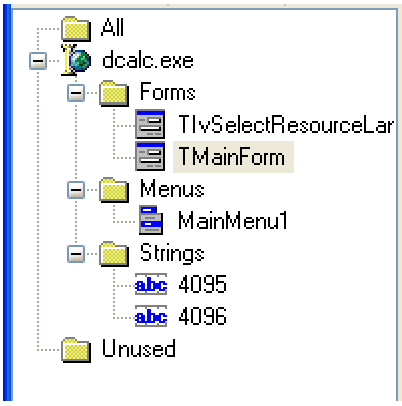


***Figure 4.** Resource tree*

Each target is divided in *pages*, which are shown in a tree structure. This simplifies navigating between different logical parts, such as menus for example.

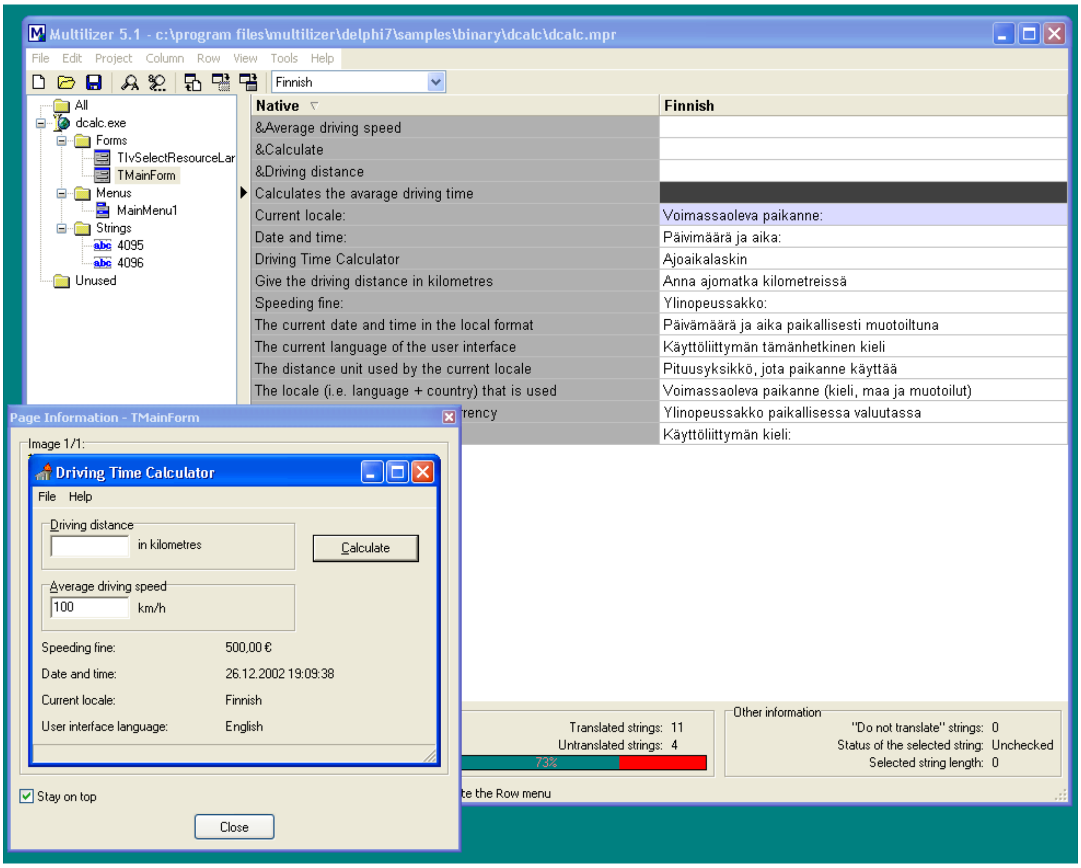
In Resource tree you can choose between:

- Showing all the strings in the project.
- Showing a certain resource, such as the menu for example.
- Showing strings that are no more used in the project, i.e., strings that have been removed from the software by the developer.

Visual translation context

If you choose a certain part of the software, you will get more information of it.

E.g., in the picture below the user has chosen to show the translations of a page called ‘TMainForm’ form and Multilizer opens the following window:



Native	Finnish
&Average driving speed	
&Calculate	
&Driving distance	
Calculates the avarage driving time	
Current locale:	Voimassaoleva paikanne:
Date and time:	Päivimäärä ja aika:
Driving Time Calculator	Ajoaikalaskin
Give the driving distance in kilometres	Anna ajomatka kilometreissä
Speeding fine:	Ylinopeussakko:
The current date and time in the local format	Päivämäärä ja aika paikallisesti muotoiltuna
The current language of the user interface	Käyttöliittymän tämänhetkinen kieli
The distance unit used by the current locale	Pituusyksikkö, jota paikanne käyttää
The locale (i.e. language + country) that is used	Voimassaoleva paikanne (kieli, maa ja muotoilut)
rency	Ylinopeussakko paikallisessa valuutassa
	Käyttöliittymän kieli:

Page Information - TMainForm

Image 1/1:

Driving Time Calculator

Driving distance: in kilometres

Average driving speed: 100 km/h

Speeding fine: 500,00 €

Date and time: 26.12.2002 19:09:38

Current locale: Finnish

User interface language: English

☑ Stay on top

Translated strings: 11

Untranslated strings: 4

73%

Other information

"Do not translate" strings: 0

Status of the selected string: Unchecked

Selected string length: 0

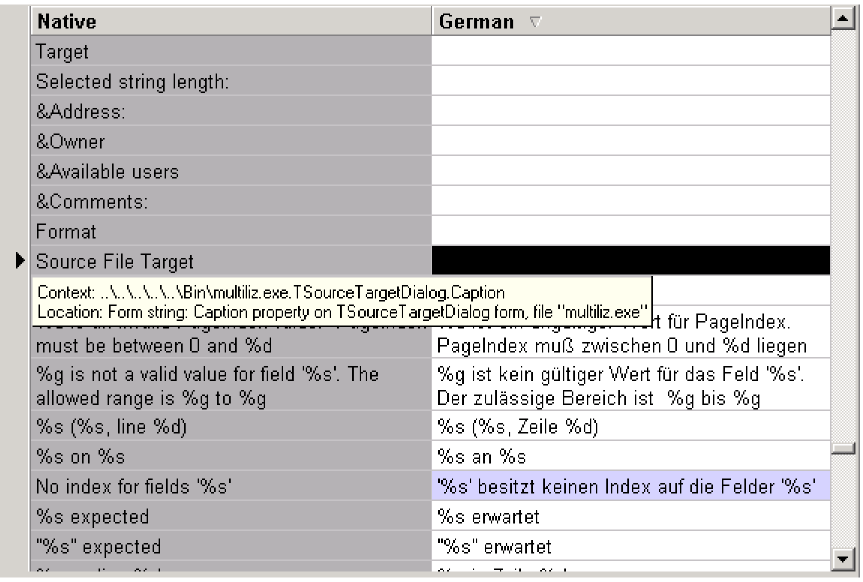
te the Row menu

Figure 5. *Page Information dialog*

This feature lets the user see the dialog and edit the related strings at the same time in the Translation Grid. Furthermore, the developer might have included notes in the Description field to give more information.

Translation Grid

The Grid is the area of Multilizer where editing is done. It consists of columns and rows. The intersection of a column and a row is a *cell*.



Native	German
Target	
Selected string length:	
&Address:	
&Owner	
&Available users	
&Comments:	
Format	
Source File Target	
must be between 0 and %d	t für PageIndex. PageIndex muß zwischen 0 und %d liegen
%g is not a valid value for field '%s'. The allowed range is %g to %g	%g ist kein gültiger Wert für das Feld '%s'. Der zulässige Bereich ist %g bis %g
%s (%s, line %d)	%s (%s, Zeile %d)
%s on %s	%s an %s
No index for fields '%s'	'%s' besitzt keinen Index auf die Felder '%s'
%s expected	%s erwartet
"%s" expected	"%s" erwartet

Context: ..\..\..\..\..\Bin\multiliz.exe.TSourceTargetDialog.Caption
Location: Form string: Caption property on TSourceTargetDialog form, file "multiliz.exe"

Figure 6. *Translation Grid*

You can only edit cells that are white. The grayed cells are locked and can be altered only by the developer.

Columns

Context column

The leftmost column (by default not shown, nor in the picture above) is called the *Context* column. It tells the native string’s code place in the original software, i.e., the technical context. (C.f. also Visual translation context, p. 7)

You can see it also by placing the cursor above the left margin of the grid. This shows the context in a hint window like in the picture above.

Native column

The strings extracted from the software are in the next column. The language can be any; it’s the language in which the original software was developed. This is called the *Native* column.

Language column

The following column contains the active target language’s translations. This is the area, where you as the translator do the translations. This is the *language* column.

NOTE!

The language column can be grayed, which means it can’t be edited. The reason for this is the operating system in which Multilizer is run doesn’t support the character set. This occurs for instance, if the language column is Arabic and the operating is Western Windows language edition.

Comment column

Both the developer and the translator can put comments to each translation. The comments are placed in the comment column.

TIP!

You can hide and view columns from the **View** menu.

Rows

Each row contains one string – i.e., *translation unit* – extracted from the program source and the existing translations of it.

Rows that cannot be translated are marked in gray. Normally the developer locks the rows in order to prevent the translation of strings that are used internally by the software, e.g., SQL statements.

Cells

The intersection of a row and a column is called a cell. Besides being the editing area for translations, the cells contain color codes to let the user see at once some key features of the string in the cell.

Cell colors are used to indicate some key characteristics of the string. Color-coding is applied to the cell background in the following way:

- Normally the cell color is white.
- If the string length exceeds a user-specified amount, the cell background is blue. The more the length differs from the native string's length the darker the shade of blue.
- Strings that are locked are shown in gray. These cannot be translated.

Status Bar

Status bar shows the selected cell, the dirty flag and the message area.

The dirty flag tells if the project has been modified or not.

In the message area you get helpful messages concerning the use of the right mouse button.

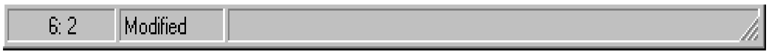


Figure 7. *Status bar*

Customizing Multilizer workspace

You can specify the visual feedback and way Multilizer works from Grid tab in general options (**Tools|Options|General**) (→ Maximum length of translations, p. 21). Default settings will give as much information as possible to the user.

Other workspace specific options are available on clicking different parts of Multilizer workspace.

Mouse action	Response
Right-click page in project view.	Shows a context menu for toggling visibility of visual context information.
Right-click a single cell in the native column.	Shows a context menu for copying cell contents to clipboard.
Right-click a single cell in any language column.	Shows a context menu for: • **State:** Set the state of translation (unchecked/translated). • **Edit:** Open an edit dialog for translation. • Editing commands (Cut, Copy, Paste, Paste native)
Right-click Native column header.	Shows a context menu for: • **Select:** Select all cells in the native column. • **Properties…** Opens a dialog for adjusting visual appearance for native column.
Right-click any language column header.	Shows a context menu for: • **Select:** Select all cells in the native column. • **Properties…** Opens a dialog for adjusting visual appearance for native column.
Click any column header	Sorts the column contents ascending/descending.

4

Working with Multilizer

Automated translation

It is recommended that you begin the translation work by applying the automated tasks. The automated tasks are the pre-translation and applying the same translation to all instances of the same Native string.

Pre-translation using Translation Memory

The pre-translation is based on the use of Translation Memory database (→ Using Translation Memory, p. 14).

Simply choose **Column|Translate|Using Translation memory…** and Multilizer will start translating the language column using contents of the Translation Memory database.

If there are several translations for one term, Multilizer will prompt the translator for the correct one.

Translation of duplicates

There are certain strings that are repeated many times. Typical strings in this category are, e.g., the 'OK' and 'Cancel' button texts. By choosing **Project | Translate | Duplicates** you can apply the same translation to all appearances of the string.

Although the same strings may have different translations, it makes sense to translate the duplicates at the beginning. The translations can easily be corrected afterwards.

Manual translation

After automatic translations, the rest of the translations need to be completed manually.

NOTE!

Multilizer stores by default all translations in Translation Memory when you save the project (→Importing glossaries to Translation Memory, p.16).

Editing translations is easy. The workplace is organized in such a way that you can always see the native language of the software along with the translations.

You have two possibilities for editing strings: you can either edit them in the cell or you can open an edit window. The choice is yours.

Using the cell area

Strings are easy to edit in the cell area. You can start editing the contents of a cell by double clicking the cell with the mouse, pressing the **F2** key, or simply starting to type.

You can stop editing the contents of a cell by clicking outside the cell with the mouse or by pressing the **F2**, **UP**, **DOWN** or **TAB** key. If the native cell contains line feeds you have to press the **Ctrl+UP**, **Ctrl+DOWN** or **Ctrl+TAB** keys.

Pressing **ENTER** stops editing and moves the cursor to the next cell if the native cell doesn't contain line feeds. If it does then pressing **ENTER** adds a line-feed to the cell. Press **Ctrl+ENTER** to stop editing and moving to the next cell.

TIP!

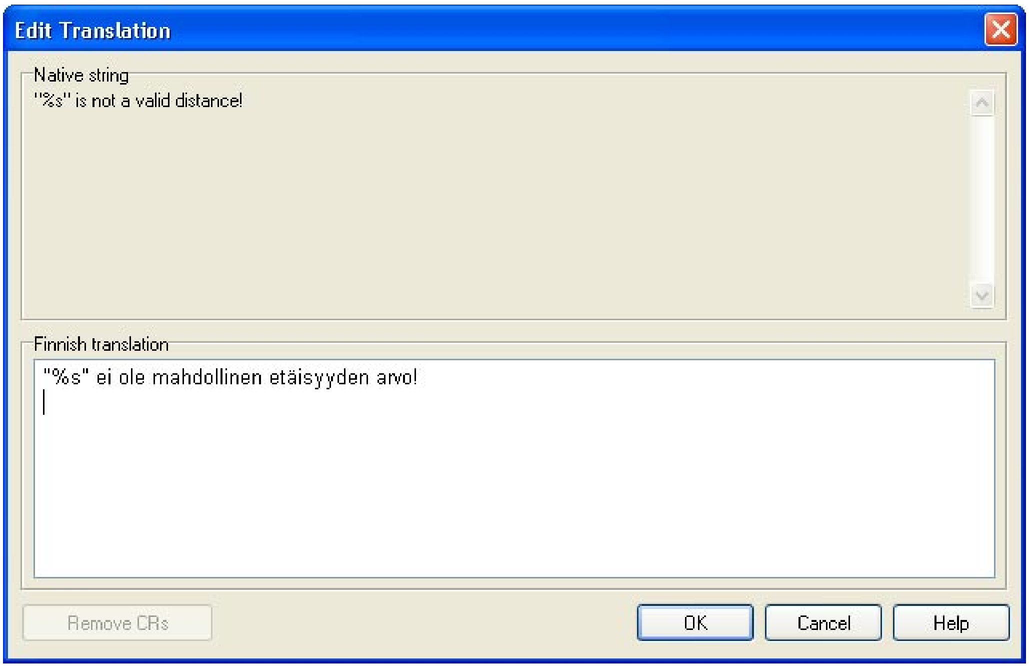
If you work with a language using the Latin alphabet with many diacritics, you may find string composing useful. It helps you enter diacritics in an intuitive way, if your keyboard doesn't include full support for the language.

Changing status of translation

Edit Window

In the grid you can also edit longer strings due to the multi-line wrapping of the cell contents. If you prefer to edit longer strings in a separate window, you can do it easily by opening an edit window.

You open the edit window by choosing **Edit | Cell...** from the menu, or with the shortcut **Ctrl+E**. You should not be editing a translation in the cell when you open the edit window.



Native string
"%s" is not a valid distance!

Finnish translation
"%s" ei ole mahdollinen etäisyyden arvo!

Figure 8. *Long strings are easy to edit in an edit window*

Finishing the work

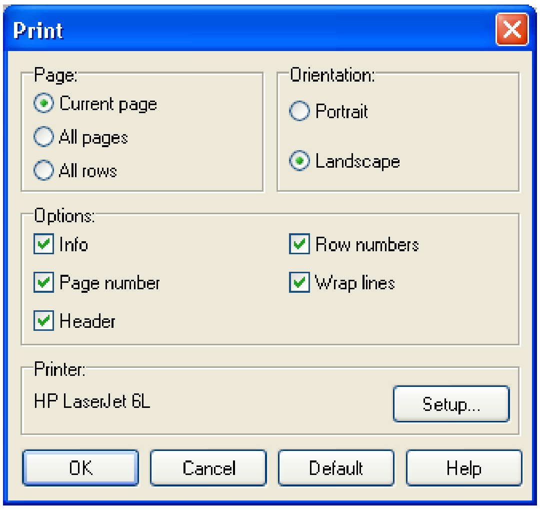
When you are finished with the translations, you have to send the translations back to the developer.

Choose **File | Exchange …** and follow the instructions.

5

Printing the translations

Multilizer lets you print out the translations. Choosing Print from the File menu brings the following dialog:



Print

Page:
- ● Current page
- ○ All pages
- ○ All rows

Orientation:
- ○ Portrait
- ● Landscape

Options:
- ☑ Info
- ☑ Row numbers
- ☑ Page number
- ☑ Wrap lines
- ☑ Header

Printer:
HP LaserJet 6L

Figure 9. *Print Dialog*

You can define whether to print all pages or the current page only.

NOTE!

In this context, printing a page means that you can choose to have any logical part of the program printed (e.g., a form file’s strings) or the entire list of strings.

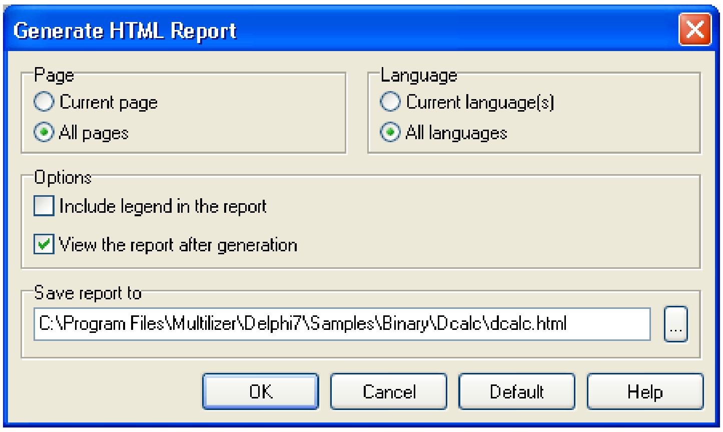
In the Options you can define what additional information should be printed along with the translations.

6

Creating reports

Multilizer lets you create reports with project statistics, such as string (translation unit), word and duplicate counts.

Choose **File | Report…** to generate a report. Following dialog appears:



Generate HTML Report

Page
○ Current page
● All pages

Language
○ Current language(s)
● All languages

Options
□ Include legend in the report
☑ View the report after generation

Save report to
C:\Program Files\Multilizer\Delphi7\Samples\Binary\Dcalc\dcalc.html

Figure 10. *HTML Report Dialog*

Multilizer will save the report in a HTML-file and opens it automatically in the default web browser.

TIP!

To understand the statistics, it is recommended that the legend is included in the report.

7

Using Translation Memory

Multilizer Translation Memory enables efficient re-use of translations in localization projects. It also improves the quality by using the same terminology consistently.

Multilizer Translation Memory is a database with translations. It is used each time the user chooses **Project | Translate | Using Translation Memory**. The more there are translations in the database the more user will automate translation work.

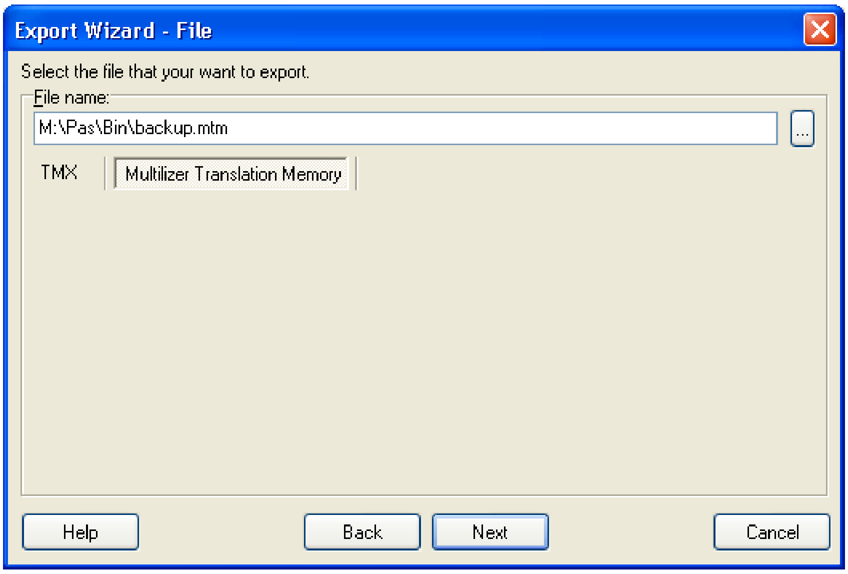
This chapter focuses Translation Memory maintenance:

- ‘Common tasks’ describes typical maintenance tasks, such as importing/exporting terminology to/from other systems, taking backups, etc.
- Exchanging translations with other products.
- The versatile support for file formats makes it possible to exchange translations between Multilizer and other products that use translation data.
- Translations can be imported to Multilizer Translation Memory (Importing glossaries to Translation Memory, p. 18) and to an existing Multilizer project.
- Translations can be exported from Multilizer Translation Memory (Export Translation Memory, p. 17) and from any existing Multilizer project.
- Microsoft® distributes with MSDN® (Microsoft Developer Network) glossaries that include translations for UI terminology in 24 languages. Any of these can be used in Multilizer, in order to use the same translations for UI items as Microsoft.
- TRADOS Workbench translation memory. Multilizer is compatible with Trados translation memory.
- SDLX. SDLX is a Translation memory product of SDL International. Multilizer is compatible with SDLX translation memory.
- Translation Memory integration’ describes how the common tasks are related to the use of Translation Memory and the entire localization process.

Common tasks

Backup Translation Memory

It is recommended that you make a backup of the Translation Memory every now and then. To do this, choose **Tools | Translation Memory…**, click **Maintenance** tab and press **Export** button. The Export Wizard appears. Press the Create a File button to export to a file. With the following Export settings Translation Memory contents are written in a Multilizer Translation Memory (MTM) file.


Export Wizard - File

Select the file that your want to export.

File name: M:\Pas\Bin\backup.mtm

Figure 11. Translation Memory backup settings

NOTE!

If the Translation Memory is on a server database, let the system administrator make the backups.

Restore Translation Memory

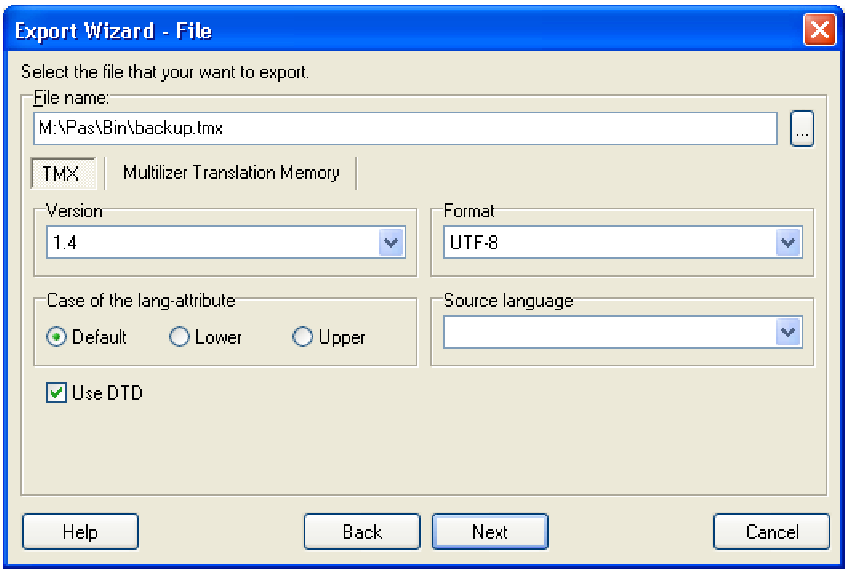
To restore a Translation Memory from a MTM-file (→ Backup Translation Memory, p. 14), choose **Tools | Translation Memory…**, click **Maintenance** tab and press **Restore** button.

WARNING!

Restoring a MTM-file erases current Translation Memory.

Export Translation Memory

To use Multilizer Translation Memory in other products, the translations can be exported to a TMX (Translation Memory Exchange) file. To do this, choose **Tools | Translation Memory…**, click **Maintenance** tab and press **Export** button. Click the TMX tab to get the following dialog:


Export Wizard - File

Select the file that your want to export.

File name: M:\Pas\Bin\backup.tmx

Version: 1.4

Format: UTF-8

Case of the lang-attribute: ● Default ○ Lower ○ Upper

Source language:

☑ Use DTD

Figure 12. Export to TMX

Create new Translation Memory

When running Multilizer for the first time, it automatically creates a new Translation Memory database. You can check the location by choosing **Tools | Translation Memory….**

If you want to change the location of the Translation Memory or change the database used, you have to create a new Translation Memory. To do this, choose **Tools | New Translation Memory….** Specify in the dialog new database settings.

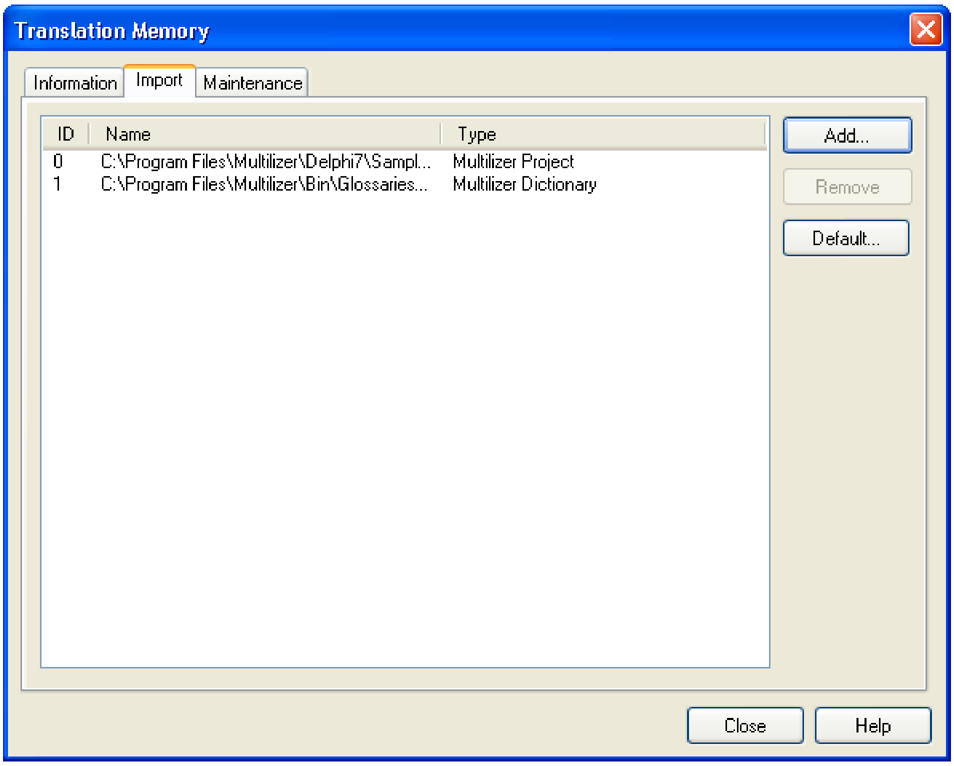
Before creating new Translation Memory, it is highly recommended to backup the existing one (→Backup Translation Memory, p. 14).

It is recommended that an experienced database user or a system administrator create the new Translation Memory.

Importing glossaries to Translation Memory

To use existing terminology for automated translation, it has to be imported as a *glossary* to Translation Memory.

To view currently imported glossaries, choose **Tools | Translation Memory…**, click **Import** tab.



ID	Name	Type
0	C:\Program Files\Multilizer\Delphi7\Sampl...	Multilizer Project
1	C:\Program Files\Multilizer\Bin\Glossaries...	Multilizer Dictionary

Figure 13. *Translation Memory contents*

After Multilizer installation there should be no glossaries in the Translation Memory.

To import a glossary, press **Add…** button. You can import both files and database tables:

- Files: TMX, Microsoft Glossary, Borland Translation Repository, text files, Multilizer project files. Several Multilizer-specific file formats are supported to leverage translations from earlier Multilizer versions.
- Database tables supported natively and others through ADO/ODBC and BDE.

Multilizer automatically stores translations in Translation Memory by importing the Multilizer Project File automatically in it. To configure this feature, choose **Tools | Options | General…** and **Translation** tab, and select the desired option for 'Save translations to the translation memory'.

Multilizer ships with a number of platform and development tool specific glossaries. They include *de facto* standard translations for common GUI elements. You can import these glossaries (located in *<Multilizer>/bin/glossaries* folder) and use them to get your software’s standard strings translated correctly. This increases the quality of the localization and the whole software.

Exchanging translations with other products

The versatile support for file formats makes it possible to exchange translations between Multilizer and other products that use translation data.

- Translations can be imported to Multilizer Translation Memory (→ Importing glossaries to Translation Memory, p. 16) and to an existing Multilizer project.
- Translations can be exported from Multilizer Translation Memory (→Export Translation Memory, p. 15) and from any existing Multilizer project.

Microsoft® distributes with MSDN® (Microsoft Developer Network) glossaries that include translations for UI terminology in 24 languages. Any of these can be used in Multilizer, in order to use the same translations for UI items as Microsoft.

TRADOS Workbench translation memory.

SDLX. SDLX is a *Translation memory* product of SDL International

Translation Memory integration

There are many possibilities using Multilizer Translation Memory. The correct way of using it depends on the localization projects, existing work-flows, and company policies.

Following chapters will show two typical scenarios in using Multilizer Translation Memory. The first one shows the situation in cases where translation is not outsourced. The latter shows a typical case with freelance translators.

Centralized Translation Memory database

Centralized Translation Memory is used typically in the company that develops and localizes software. By centralizing the terminology management, the entire company can use the terminology consistently.

Following figure illustrates a typical configuration:

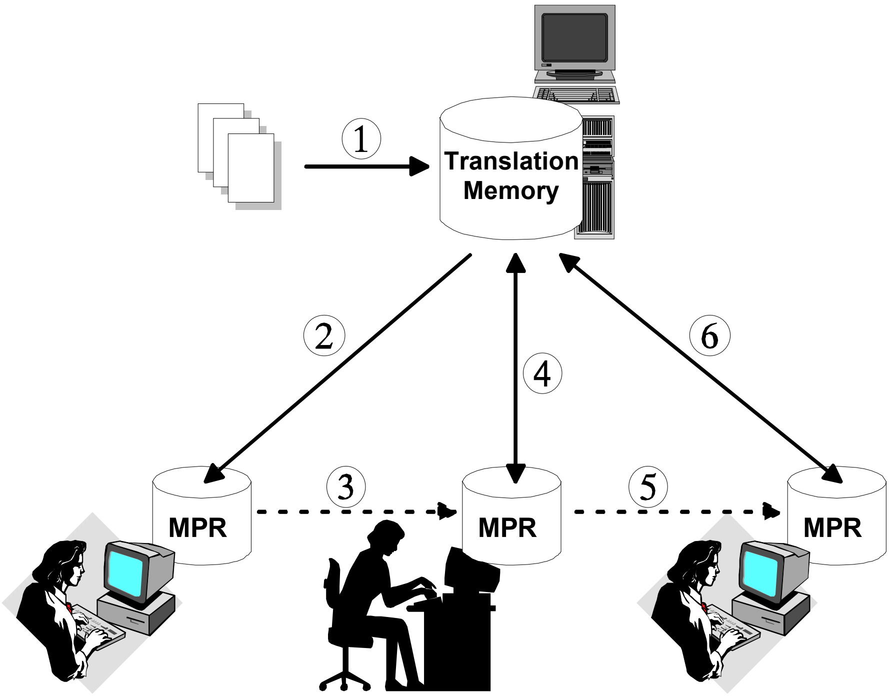


Figure 14. *Translation Memory integration with in-house translators*

① Existing translations and glossaries are imported in Multilizer Translation Memory (C.f. Importing glossaries to Translation Memory, p. 16)

② Developer creates a Localization Kit in the active Multilizer Localization project. The Kit includes the texts to be translated. Multilizer application is not included in the Kit.

③ Translator opens the Localization Kit (C.f. Installation, p. 3) in Multilizer.

④ Translator uses Translation Memory to speed up the process and achieve better quality in translations. Multilizer is configured not to add new (manual) translations in the Translation Memory in this phase.

⑤ When finished, translations are sent back to the developer. Developer integrates the translations in the Multilizer project.

Using Multilizer throughout has preserved the technical context of the translations, so it's very simple to create the localized versions of the software.

⑥ Quality Assurance engineers test the quality of the localized versions. In this scenario translations are added in the Translation Memory after validation, which ensures better quality.

Outsourced Translation

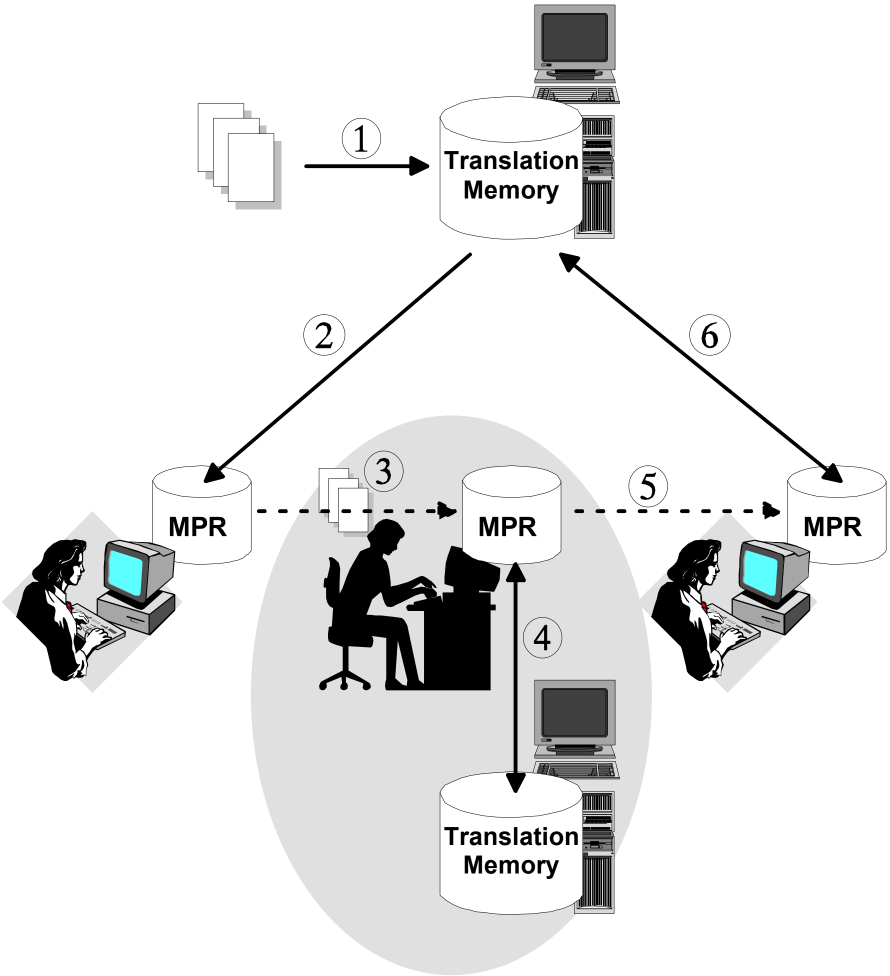


Figure 15. *Translation Memory integration using outsourced translation*

Activities in the software company

① Existing translations and glossaries are imported in Multilizer Translation Memory.

② The developer creates a Localization Kit that includes Multilizer® Translator Edition™ and the texts to be translated, and ③ optionally company/industry specific glossaries. The Kit is sent to the translator.

Activities in the Translation Agency

③ Translator opens the Localization Kit. It automatically installs Multilizer and the texts to be translated on the translator's computer. Optional glossaries are imported in the Translation Memory that may have translations from earlier projects already.

④ Translator uses Multilizer to translation, and uses Translation Memory to speed up the process and achieve better quality in translations.

⑤ When finished, the translations are sent back to developer.

Back in the software company

⑥ Developer integrates the translations in his copy of Multilizer localization database.

Using Multilizer throughout has preserved the technical context of the translations, thus it's very simple to create the localized versions of the software.

8

Further translation concerns

There are major differences between the translation of words and phrases in software and conventional translation work. The clearest differences are the following:

- There are a lot of one-word translations to do. The strings to be translated are mostly very short.
- The GUI (Graphical User Interface) elements have standard and mostly explicit translations.
- Strings may include characters and codes that have a special purpose in the context of the software functionality.
- The GUI may require a certain maximum length of translations.

In addition, there are many other features that can be derived from those mentioned above. Multilizer includes many features that help you do the translation work.

Translation memory

Since most of the software's strings are short, translation memory provides the most accurate approach to automating the translation work. (In document translation, machine translation may be the most suitable.).

Characters with a special purpose

Depending on the programming language and the programming technique that the developer has used, the strings in the dictionary may include special characters.

Sample string	Explanation
Cannot create file %s	%s is used to denote a string inside another string. E.g., if ‘temp.txt’ is assigned to %s, the software would show the following: ‘Cannot create file temp.txt’ So, %s must exist also in the translation.
%s (%s, line %d)	This is like the example above. There can be multiple special characters in one string to be translated. If you have to change the order of the special characters in your translation, inform the developer of this.
%0:s (%1:s, line %2:d)	If the code is property internationalized there should not be a string like in the above row but like in the left. As you can see, every variable has been indexed so you can freely change the order of variables.
&File	The & sign is used in menu items and button captions to show the hotkey, i.e., the character that is underlined and is used to trigger the menu. In the example at the left, the text would be shown as **File** in the menu. It’s up to you to decide which hotkey you want to use in the translation.
CODEBASE_ENUM	This kind of dictionary items can normally be left untranslated. The developer might be using it in a way related to software functionality. Normally the developer should mark strings that need no translations. These strings appear on a gray background in the dictionary.

TIP!

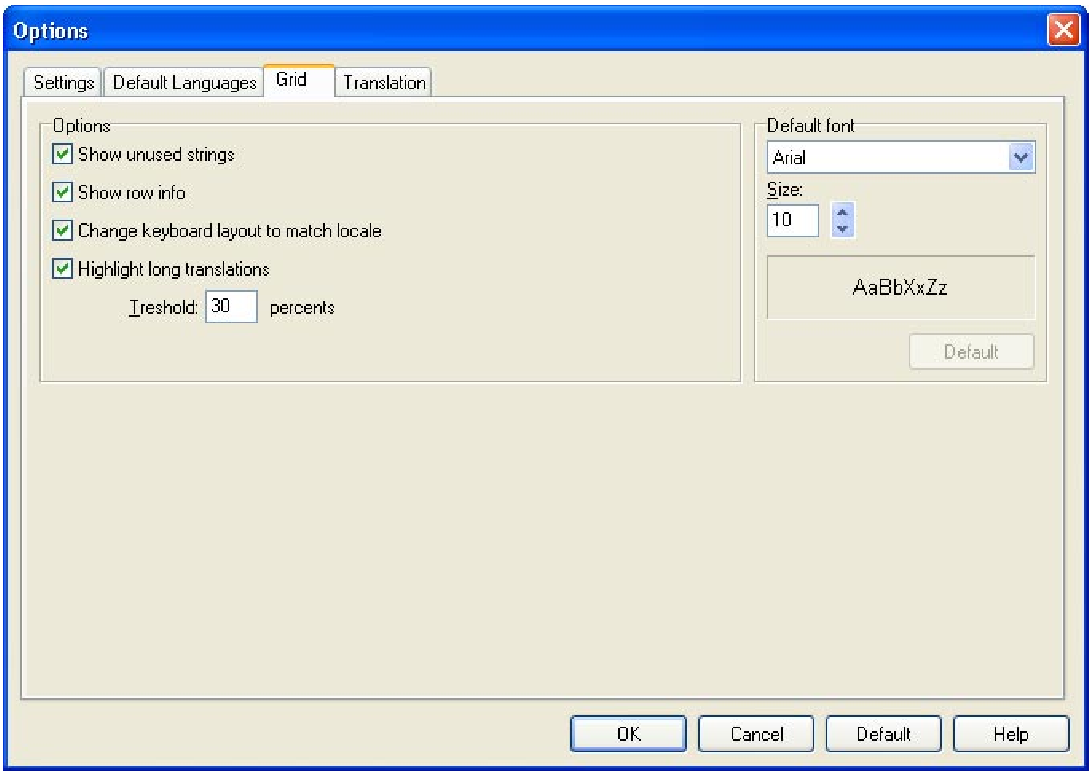
Remember to check possible comments that are attached to the translation unit. If there’s a comment available, there is a red dot at the right margin in the workspace.

Maximum length of translations

One common issue is the length of the strings to be translated. The string layout in the original software may accept only slight changes in string length. Therefore the translator has to pay special attention to this.

Multilizer helps in showing possible troublesome translations: it makes the cell background darker the more the translation’s length exceeds the native string’s length.

You can define the allowed threshold for showing color codes for the string length change: select **Tools | Options | General…** and choose the **Grid** tab.



Options

☑ Show unused strings

☑ Show row info

☑ Change keyboard layout to match locale

☑ Highlight long translations

Treshold: 30 percents

Default font

Arial

Size: 10

AaBbXxZz

Default

Figure 16. *Setting translation grid options*

If applicable, check if there are screen shots attached to the page, where the long string length difference appears. In addition, check if the developer has added a comment to the string.

TIP!

If you think there might be layout problems in the software with the translation, please let the developer know about it. Add your own comment about this in the dictionary.

Appendix I: Sharing translations with TRADOS® Translator's Workbench

Translations in Multilizer® are easily exported to TRADOS® Translator's Workbench, and vice versa.

Multilizer Wizards make it extremely easy to share translations between these systems.

All translations are transferred from one system to another using TMX (Translation Memory Exchange → www.lisa.org/tmx) files. TRADOS supports and requires TMX version 1.1 only.

Following instructions apply to:

- TRADOS® Translator's Workbench™: 2.3, 3.0 and 5.1.1 (Build 217 or newer).
- Multilizer 5.0 (Build 71, or newer) or Multilizer 5.1.

Export translations from Multilizer to Translator's Workbench

All existing translations in Multilizer can be used with Trados. To do this,

1. Use Multilizer Export Wizard to export translations to a file, and
2. Use Translator's Workbench to import the file to the active Translation Memory.

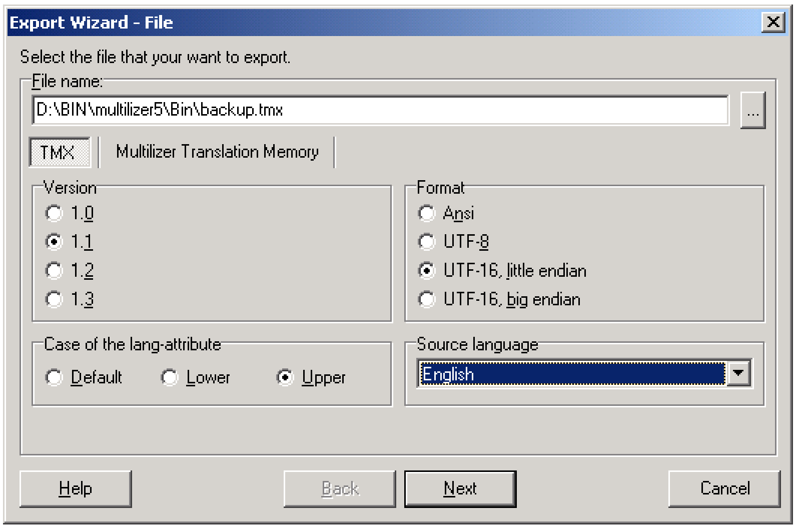
Multilizer

Multilizer Wizards assist both in exporting project translations as well as in exporting the entire Multilizer Translation Memory.

- Choose **File|Export** to export project translations.
- Choose **Tools|Translation Memory...**, click **Maintenance** tab, and press **Export...** button to export Multilizer Translation Memory.

Multilizer Export Wizard assists in creating the TMX file.

Use the following settings to create a TMX file supported by TRADOS 5.1.


Export Wizard - File

Select the file that your want to export.

File name: D:\BIN\multilizer5\Bin\backup.tmx

Version: ○ 1.0 ● 1.1 ○ 1.2 ○ 1.3

Format: ○ Ansi ○ UTF-8 ● UTF-16, little endian ○ UTF-16, big endian

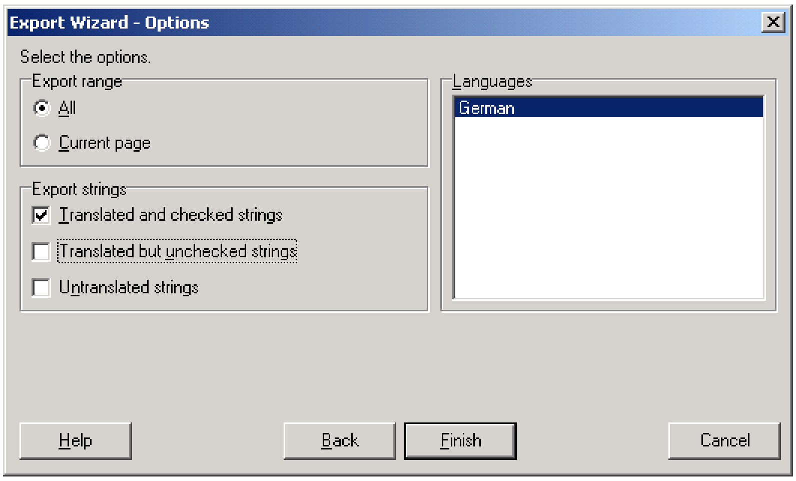
Case of the lang-attribute: ○ Default ○ Lower ● Upper

Source language: English

Figure 17.

In Trados 5.1 'Source language' must match the 'native' language in Multilizer, otherwise Trados will not be able to import the resulting file. Trados 2.3 and 3.0 require 'Neutral' as Source language.

If you are exporting a Multilizer Project, next screen lets you define how the translations are exported. If you are exporting the Translation Memory, following screen is skipped.



Export Wizard - Options

Select the options.

Export range

● All

○ Current page

Export strings

☑ Translated and checked strings

☐ Translated but unchecked strings

☐ Untranslated strings

Languages

German

Figure 18.

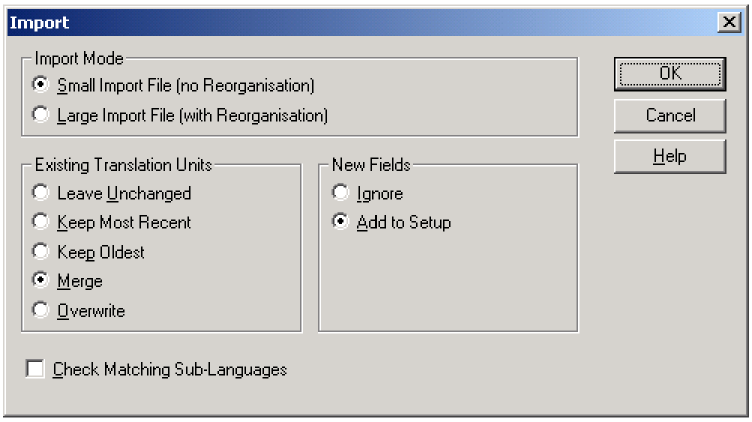
In most cases you should only export 'Translated and checked strings'. You can only import one language pair at a time to Trados. Ensure that you export only one language from Multilizer in addition to the Native language.

After finishing the Export Wizard, you need to import the text file to Trados.

Translator's Workbench

Open an existing Translation Memory in exclusive mode. Choose **File|Import...**

Ensure the Import Mode is as follows.



Import

Import Mode

● Small Import File (no Reorganisation)

○ Large Import File (with Reorganisation)

Existing Translation Units

○ Leave Unchanged

○ Keep Most Recent

○ Keep Oldest

● Merge

○ Overwrite

New Fields

○ Ignore

● Add to Setup

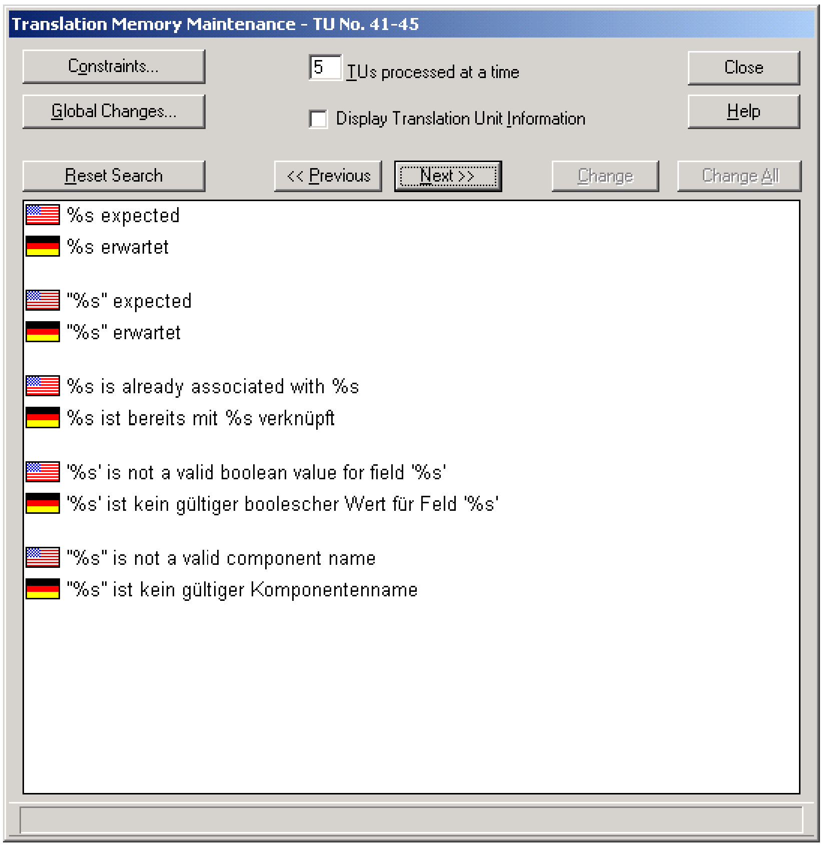
☐ Check Matching Sub-Languages

Figure 19.

Click OK and specify the TMX file to be imported.

On importing translations Trados doesn't make by default any difference between sub-languages (e.g., English UK and English US). If you want to differ sub-languages, check 'Check Matching Sub-Languages'.

You can check the availability of new translations by choosing **File|Maintenance**. In maintenance view you can browse through the data.



5 TUs processed at a time

□ Display Translation Unit Information

🇺🇸 %s expected
🇩🇪 %s erwartet

🇺🇸 "%s" expected
🇩🇪 "%s" erwartet

🇺🇸 %s is already associated with %s
🇩🇪 %s ist bereits mit %s verknüpft

🇺🇸 '%s' is not a valid boolean value for field '%s'
🇩🇪 '%s' ist kein gültiger boolescher Wert für Feld '%s'

🇺🇸 "%s" is not a valid component name
🇩🇪 "%s" ist kein gültiger Komponentenname

Figure 20.

Export translations from Translator’s Workbench to Multilizer

Any existing translations in Trados can be used in Multilizer. To do this,

1. Use Translator’s Workbench to export translations to a file, and
2. Use Multilizer Import Wizard to import the file to Multilizer Translation Memory.

Translator’s Workbench

Open the Translation Memory you want to export. Choose **File|Export…**

Add the language fields that you want to include in the TMX.

Translations must be exported in language pairs. Source language must be the same as the native language used in Multilizer projects. If there are several native languages, use the most common one.

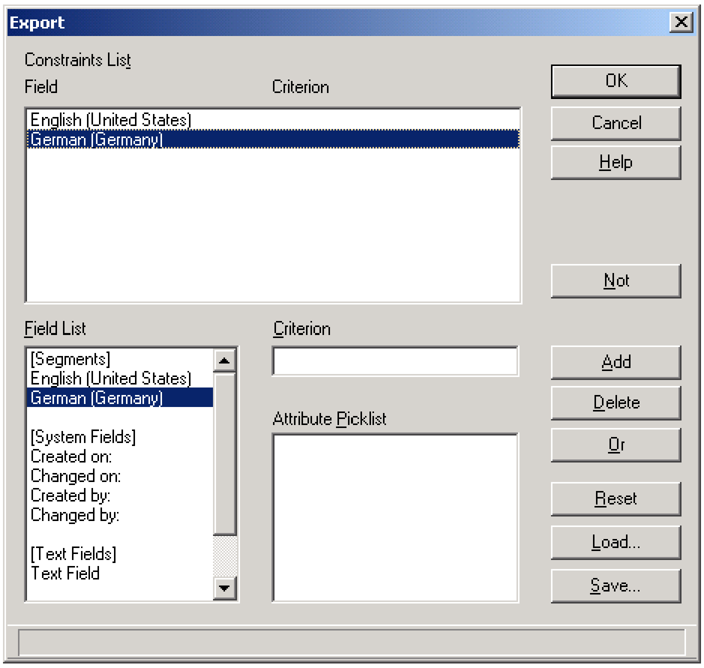
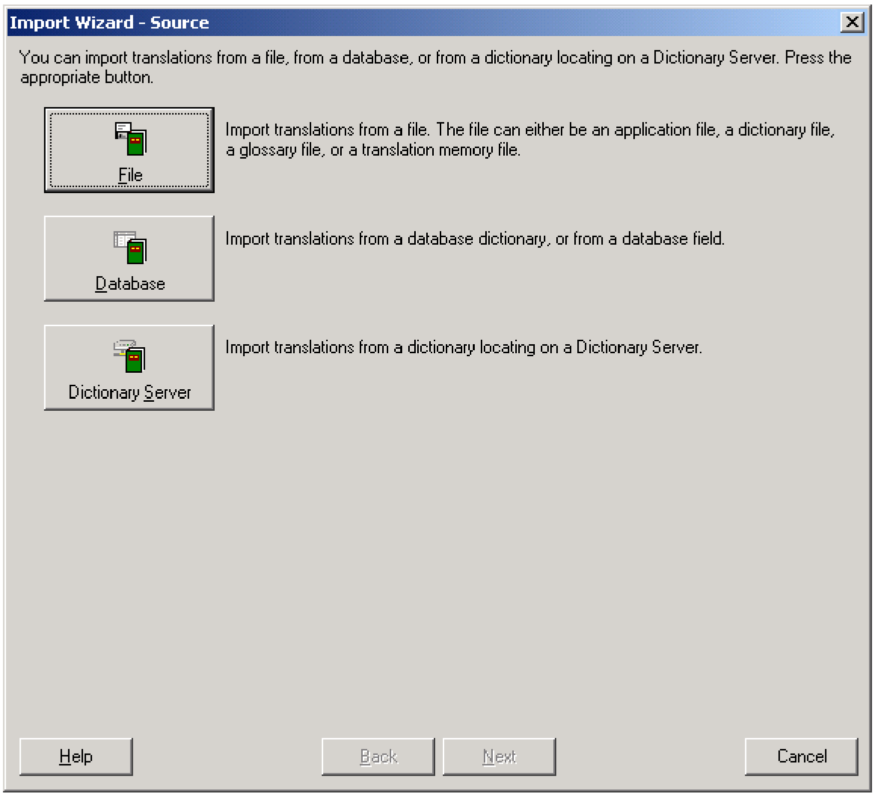


Figure 21.

Click OK and save translations as a TMX file.

Multilizer

Choose **Tools|Translation Memory...**, click the **Import** tab, and press the **Add...** button to run Multilizer Import Wizard. Choose file import.



You can import translations from a file, from a database, or from a dictionary locating on a Dictionary Server. Press the appropriate button.

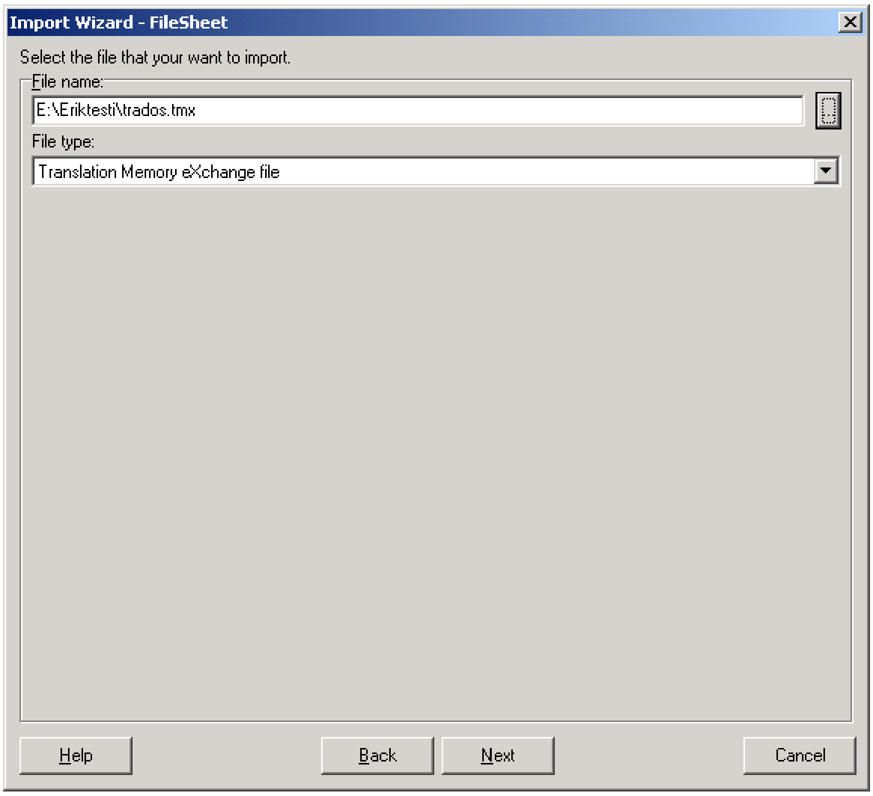
Import translations from a file. The file can either be an application file, a dictionary file, a glossary file, or a translation memory file.

Import translations from a database dictionary, or from a database field.

Import translations from a dictionary locating on a Dictionary Server.

Figure 22.

Specify the location of the TMX file created in Trados.



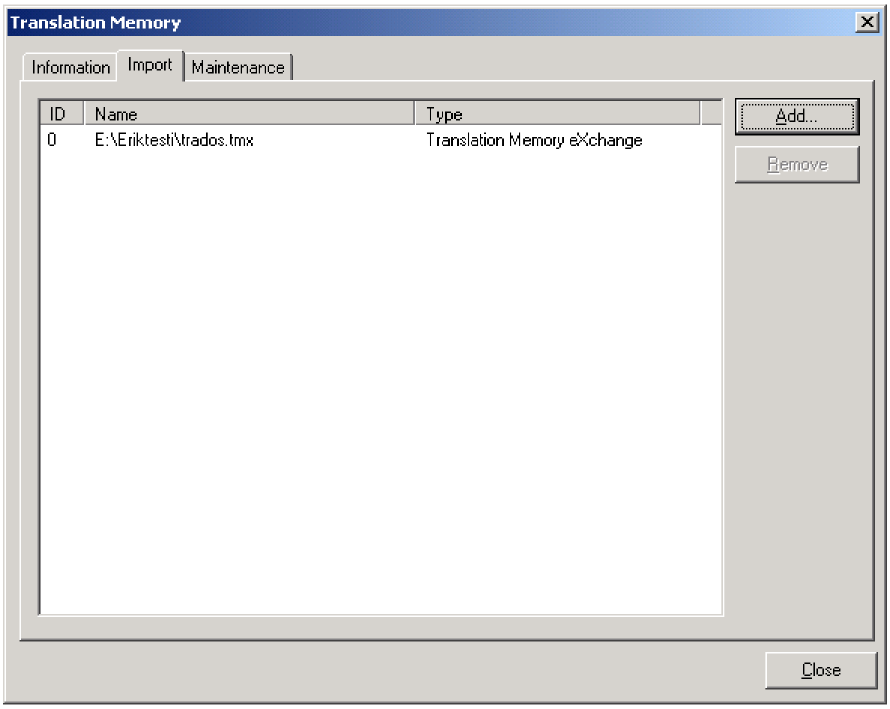
Select the file that your want to import.

File name: E:\Eriktesti\trados.tmx

File type: Translation Memory eXchange file

Figure 23.

Multilizer will import the translations to Multilizer Translation Memory and add the TMX file to the list of imported files.



ID	Name	Type
0	E:\Eriktesti\trados.tmx	Translation Memory eXchange

Figure 24.

Later, remove the imported translations from Multilizer Translation Memory simply by removing the TMX file from the list above.

Appendix II: Sharing terminology with TRADOS® MultiTerm

Terminology maintained in MultiTerm is easily exported to Multilizer. Multilizer Wizards make this extremely easily. All translations are transferred from one system to another using ANSI text files.

Following instructions apply to:

- TRADOS® MultiTerm 5.1.1 (Build 217, or newer)
- Multilizer® 5.1 (Build 70, or newer).

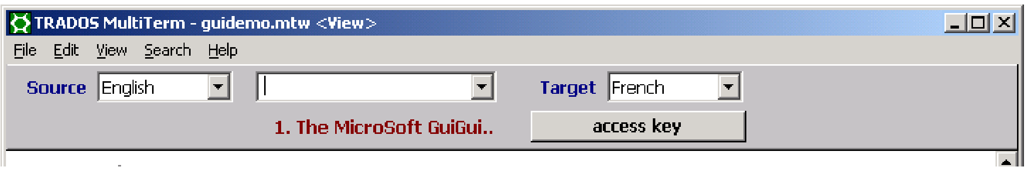
Export translations from MultiTerm to Multilizer

To export terminology from TRADOS MultiTerm to Multilizer,

1. Use TRADOS MultiTerm to export translations to a file, and
2. Use Multilizer Import Wizard to import the file to Multilizer Translation Memory.

MultiTerm

Easiest way to export terminology is to export a *language pair*. The active language pair consists in MultiTerm of Source and Target language as shown in the picture below.



TRADOS MultiTerm - guidemo.mtw <View>

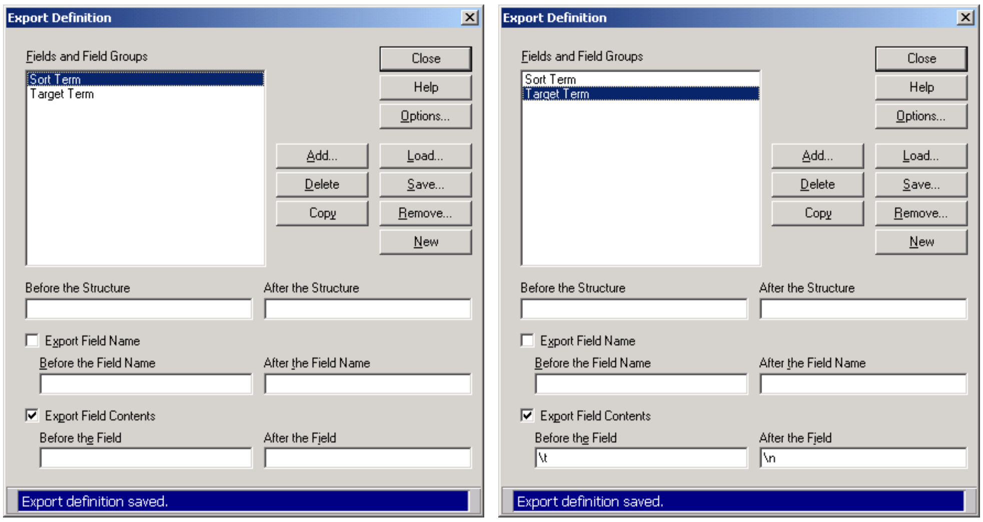
Source: English

Target: French

1. The MicroSoft GuiGui..

Figure 25.

Before exporting terminology, you have to define export properties. Open MultiTerm and the Termbase you want to export. Choose **File|Define Export…** Use the **Add…** button to define following fields and field properties to be exported.



Export Definition

Fields and Field Groups: Sort Term, Target Term

Before the Structure:
After the Structure:

☐ Export Field Name
Before the Field Name:
After the Field Name:

☑ Export Field Contents
Before the Field:
After the Field:

Export definition saved.

Export Definition

Fields and Field Groups: Sort Term, Target Term

Before the Structure:
After the Structure:

☐ Export Field Name
Before the Field Name:
After the Field Name:

☑ Export Field Contents
Before the Field: \t
After the Field: \n

Export definition saved.

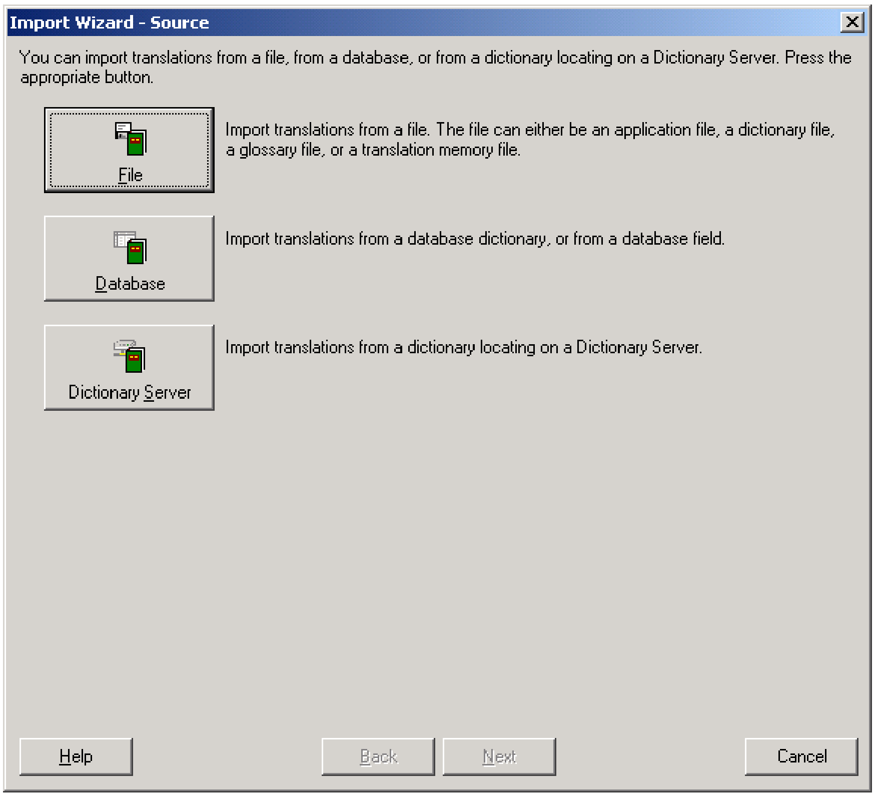
Figure 26.

You can save the Export Definitions for later use with **Save** button.

Close the dialog and choose **File|Export all entries…** from the main menu. Save the terminology in ANSI format.

Multilizer

Choose **Tools|Translation Memory...**, click **Import** tab, and press **Add...** button to run Multilizer Import Wizard. Choose file import.



Import Wizard - Source

You can import translations from a file, from a database, or from a dictionary locating on a Dictionary Server. Press the appropriate button.

File — Import translations from a file. The file can either be an application file, a dictionary file, a glossary file, or a translation memory file.

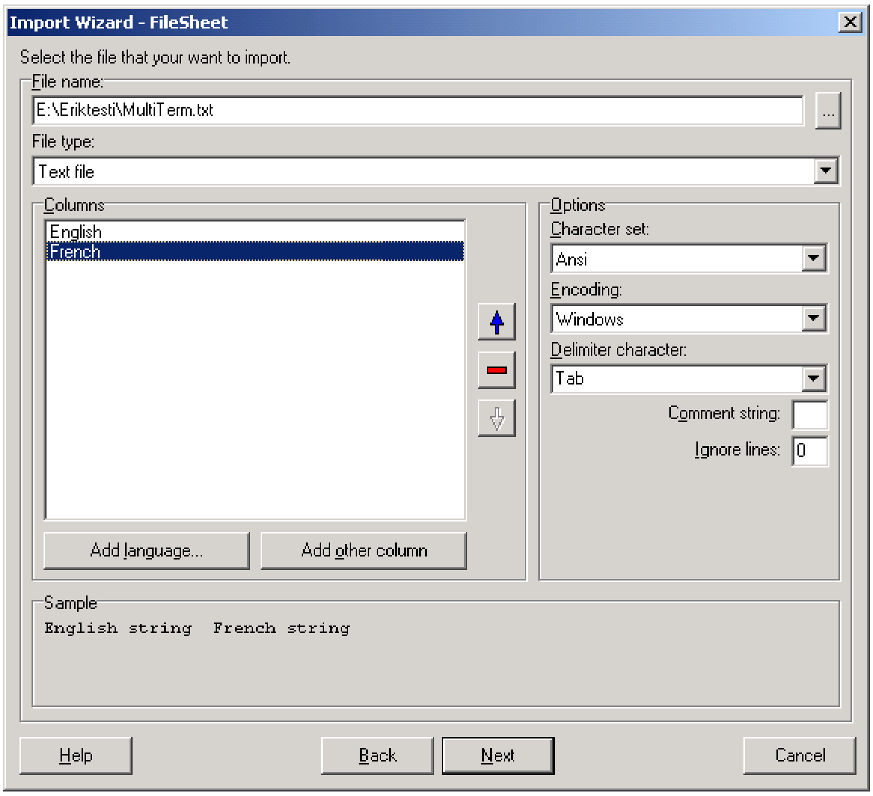
Database — Import translations from a database dictionary, or from a database field.

Dictionary Server — Import translations from a dictionary locating on a Dictionary Server.

Figure 27.

Specify the location of the text file created in MultiTerm.

Before the file can be imported, you have to specify the format of it.



Import Wizard - FileSheet

Select the file that your want to import.

File name: E:\Eriktesti\MultiTerm.txt

File type: Text file

Columns:
English
French

Options
Character set: Ansi
Encoding: Windows
Delimiter character: Tab
Comment string:
Ignore lines: 0

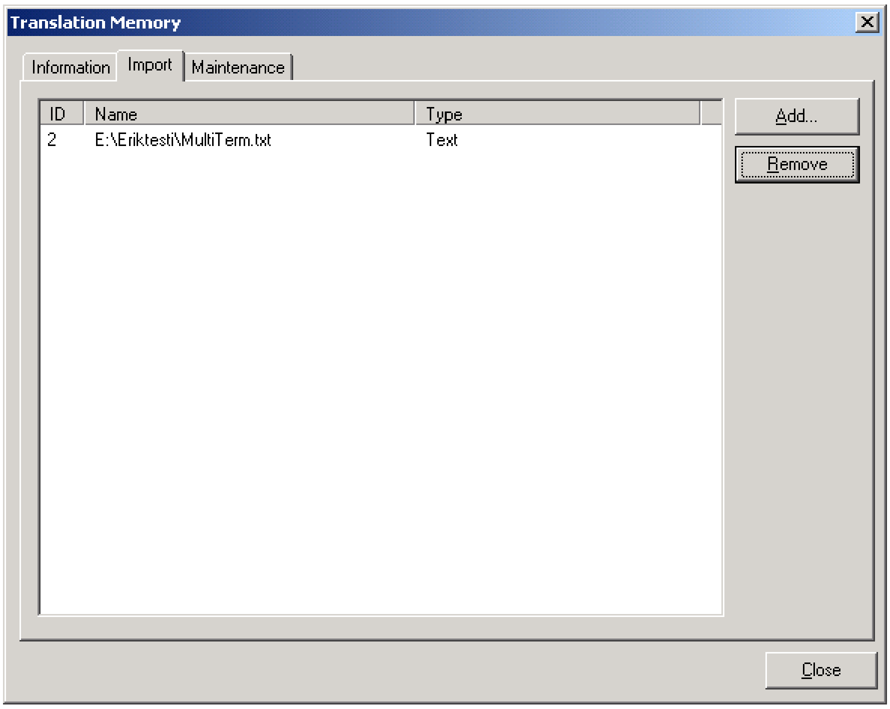
Sample

```
English string   French string
```

Figure 28.

Ensure the settings are as above and the languages to import are the same as you specified in MultiTerm.

Multilizer will import the terminology to Multilizer Translation Memory and add the text file to the list of imported files.



Translation Memory

ID	Name	Type
2	E:\Eriktesti\MultiTerm.txt	Text

Figure 29.

Index