Anhang

The localization tool *L10N*

1 Introduction

Currently available localization tools are mainly designed for the localization of *Windows* programmes. Therefore, the company *Georg Heeg eK* developed the localization tool *L10N*, so that applications created with *VisualWorks* can easily be localized, too.

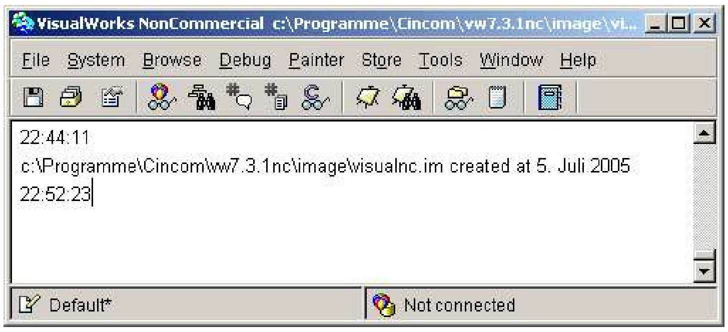
L10N can be run on all platforms *VisualWorks* is running. These are namely *Windows* (*95/98/ME/NT/2000*), *PowerMac*, *Intel Linux*, *AIX*, *SGI Irix*, *Compaq UNIX*, *HP-UX*, and *Solaris*.

Unlike other localization tools and translation memory systems, *L10N* does not create a separate file for the localization project. Instead, the data accumulated in the translation memory is stored in a *VisualWorks* image file. The data accumulated in the translation memory can be stored in an export file by exporting them.

The entire text of an application to be translated is the totality of all texts seen by the user. In *VisualWorks,* those texts are called user messages. The subdivision into translation units is made accordingly to the user messages. Each user message represents a translation unit.

2 The user interface

The localization tool *L10N* is implemented in the *VisualWorks 7.3.1* environment (see Figure A.1). It can be opened by selecting the first icon on the right in the toolbar, a blue book, or in the **Window** menu the menu item **1. [FUI] Localizer**.

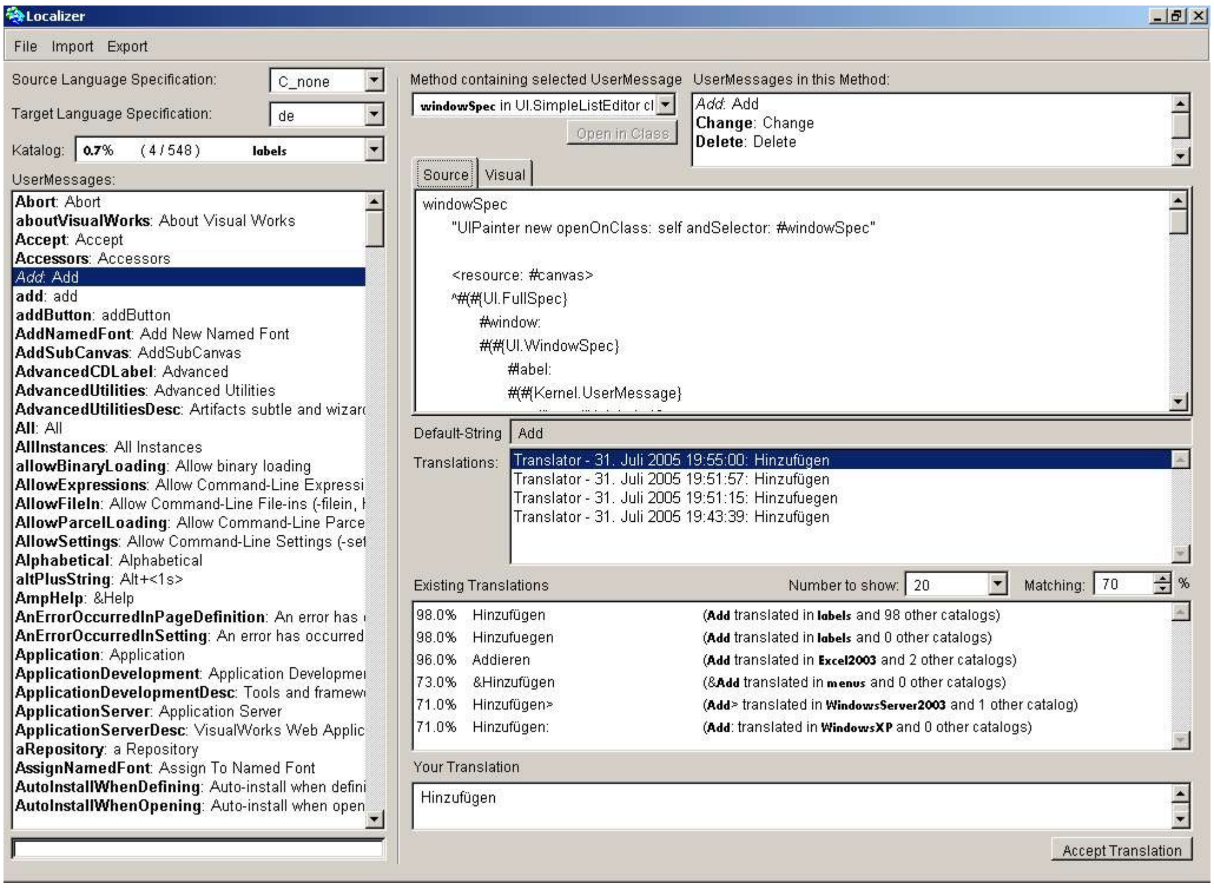

VisualWorks NonCommercial c:\Programme\Cincom\vw7.3.1nc\image\vi...
File System Browse Debug Painter Store Tools Window Help

```
22:44:11
c:\Programme\Cincom\vw7.3.1nc\image\visualnc.im created at 5. Juli 2005
22:52:23
```

Default* | Not connected

Fig. A.1: The *VisualWorks* launcher with the ***L10N*** icon

The localization tool will be opened in a separate window, which is shown in Figure A.2.



Source Language Specification: C_none
Target Language Specification: de
Katalog: 0.7% (4 / 548) labels

UserMessages:

Abort: Abort
aboutVisualWorks: About Visual Works
Accept: Accept
Accessors: Accessors
Add: Add
add: add
addButton: addButton
AddNamedFont: Add New Named Font
AddSubCanvas: AddSubCanvas
AdvancedCDLabel: Advanced
AdvancedUtilities: Advanced Utilities
AdvancedUtilitiesDesc: Artifacts subtle and wizarc
All: All
AllInstances: All Instances
allowBinaryLoading: Allow binary loading
AllowExpressions: Allow Command-Line Expressi
AllowFileIn: Allow Command-Line File-ins (-filein, I
AllowParcelLoading: Allow Command-Line Parce
AllowSettings: Allow Command-Line Settings (-set
Alphabetical: Alphabetical
altPlusString: Alt+<1s>
AmpHelp: &Help
AnErrorOccurredInPageDefinition: An error has (
AnErrorOccurredInSetting: An error has occurred
Application: Application
ApplicationDevelopment: Application Developmei
ApplicationDevelopmentDesc: Tools and framew
ApplicationServer: Application Server
ApplicationServerDesc: VisualWorks Web Applic
aRepository: a Repository
AssignNamedFont: Assign To Named Font
AutoInstallWhenDefining: Auto-install when defini
AutoInstallWhenOpening: Auto-install when open

Method containing selected UserMessage: **windowSpec** in UI.SimpleListEditor cl

UserMessages in this Method:
Add: Add
Change: Change
Delete: Delete

```
windowSpec
	"UIPainter new openOnClass: self andSelector: #windowSpec"

	<resource: #canvas>
	^#(#{UI.FullSpec}
		#window:
		#(#{UI.WindowSpec}
			#label:
			#(#{Kernel.UserMessage}
```

Default-String: Add

Translations:
Translator - 31. Juli 2005 19:55:00: Hinzufügen
Translator - 31. Juli 2005 19:51:57: Hinzufügen
Translator - 31. Juli 2005 19:51:15: Hinzufuegen
Translator - 31. Juli 2005 19:43:39: Hinzufügen

Existing Translations — Number to show: 20 — Matching: 70 %

98.0%	Hinzufügen	(**Add** translated in **labels** and 98 other catalogs)
98.0%	Hinzufuegen	(**Add** translated in **labels** and 0 other catalogs)
96.0%	Addieren	(**Add** translated in **Excel2003** and 2 other catalogs)
73.0%	&Hinzufügen	(&**Add** translated in **menus** and 0 other catalogs)
71.0%	Hinzufügen>	(**Add**> translated in **WindowsServer2003** and 1 other catalog)
71.0%	Hinzufügen:	(**Add**: translated in **WindowsXP** and 0 other catalogs)

Your Translation: Hinzufügen

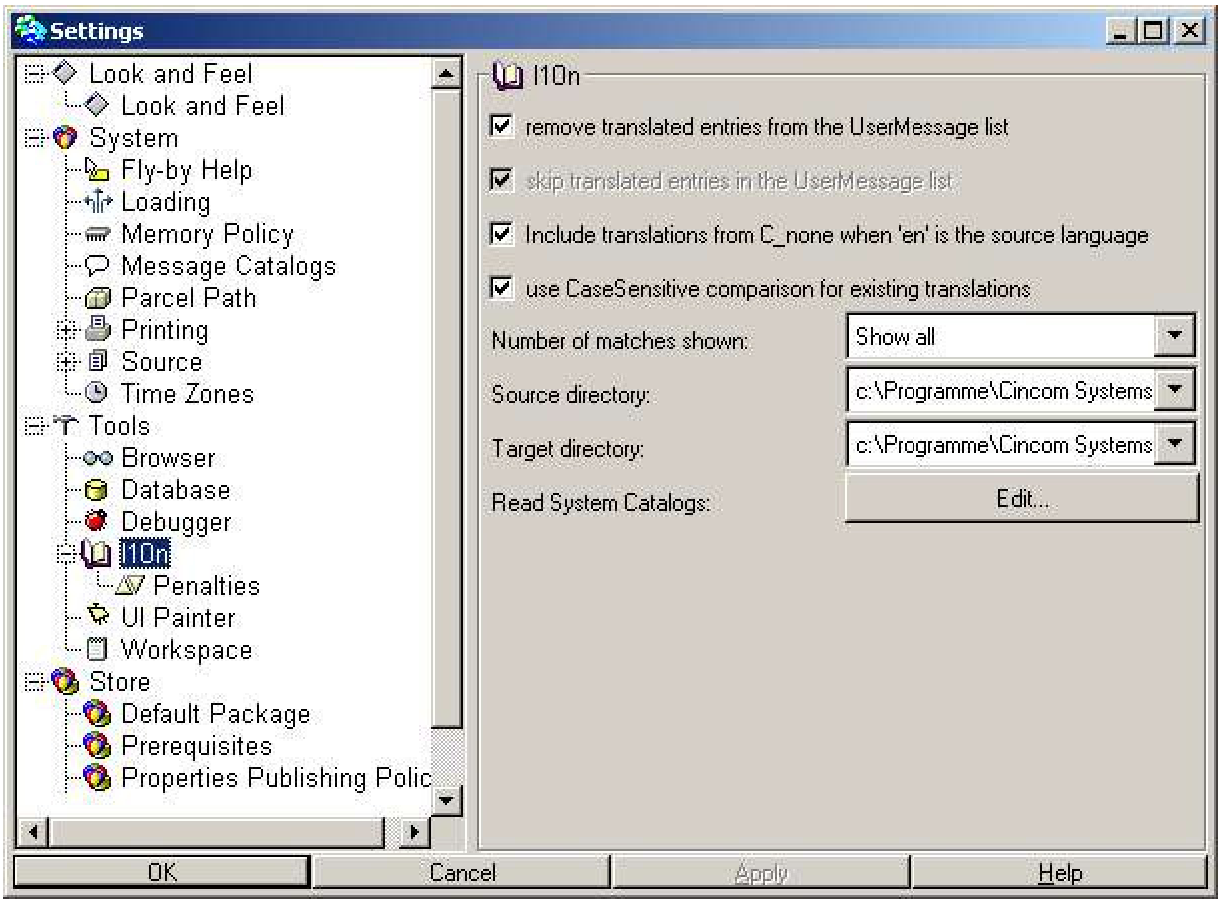
Fig. A.2: The localization tool *L10N*

The menu bar comprises the menu items **File**, **Import** and **Export**. The main functions of the localization process can be accessed using the **File** menu. These are the commands **Accept Translation** (Strg+S), **Read System** and **Write and Compile Catalogs** (Umschalt+Strg+S).

The **Read System** command (it can also be selected by pressing the **Edit...** button in the **Settings** dialog box) starts the search for constructs with user messages in all the methods of the application. The user messages are displayed in the **UserMessages** window.
The **Accept Translation** command confirms the translation entered and writes it in the resident translation memory.

The **Write and Compile Catalogs** command writes the translations into the LBL files and the IDX files, so that they can be reused in other *VisualWorks* installations.
The **File** menu also comprises the menu items **Options**, **Original Language**, **Empty TM** and **TM Stats**.

After choosing the **Options** menu item, the **Settings** dialog box will be opened. In this dialog box, the settings for the localization tool can be changed (see Figure A.3). It comprises the check boxes **Remove translated entries from the UserMessage list**, **Skip translated entries in the UserMessage list**, **Include translation from C_none when 'en' is the source** and **Use CaseSensitive comparison for existing translations**. Furthermore, it comprises the fields **Number of matches shown** (**Show all**, **1**, **3**, **10**, **25**, **50**, **100**), **Source directory** and **Target directory**.



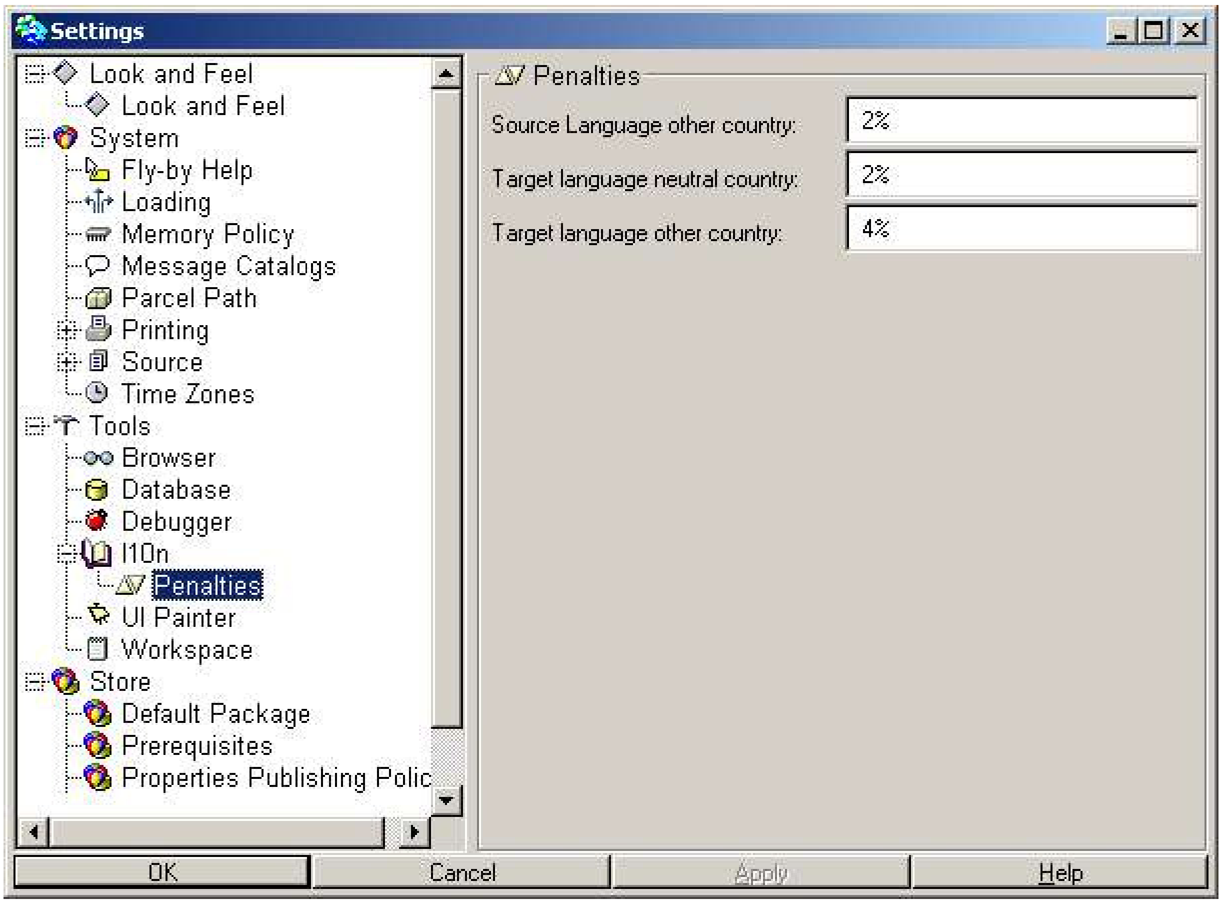
l10n

- [✓] remove translated entries from the UserMessage list
- [✓] skip translated entries in the UserMessage list
- [✓] Include translations from C_none when 'en' is the source language
- [✓] use CaseSensitive comparison for existing translations

Number of matches shown:	Show all
Source directory:	c:\Programme\Cincom Systems
Target directory:	c:\Programme\Cincom Systems
Read System Catalogs:	Edit...

Fig. A.3: The **Settings** dialog box

In the **Penalties** dialog box (see Figure A.4), which can be accessed via the window at the left hand side of the **Settings** dialog box, you can adjust the penalties for the matches found in the translation memory. The penalties can be set from 0 per cent to 100 per cent, if the source text has another territory identifier than the source text of the match found in the memory, if the translation has no territory identifier, in contrast to the translation of the match found in the memory, or if the translation has another territory identifier than the translation found in the memory.



Penalties

Source Language other country: 2%

Target language neutral country: 2%

Target language other country: 4%

Fig. A.4: The **Penalties** dialog box

Using the menu item **Original Language**, you can display the user interface of an application either in the original language or in the target language. The menu item **Empty TM** can be used to empty the resident translation memory. With the menu item **TM Stats** you can display the number of entries currently stored in the resident translation memory (see Figure A.5).

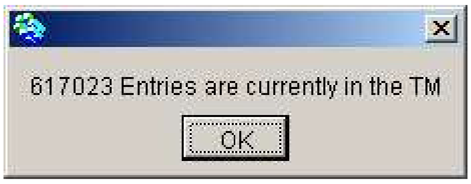

617023 Entries are currently in the TM

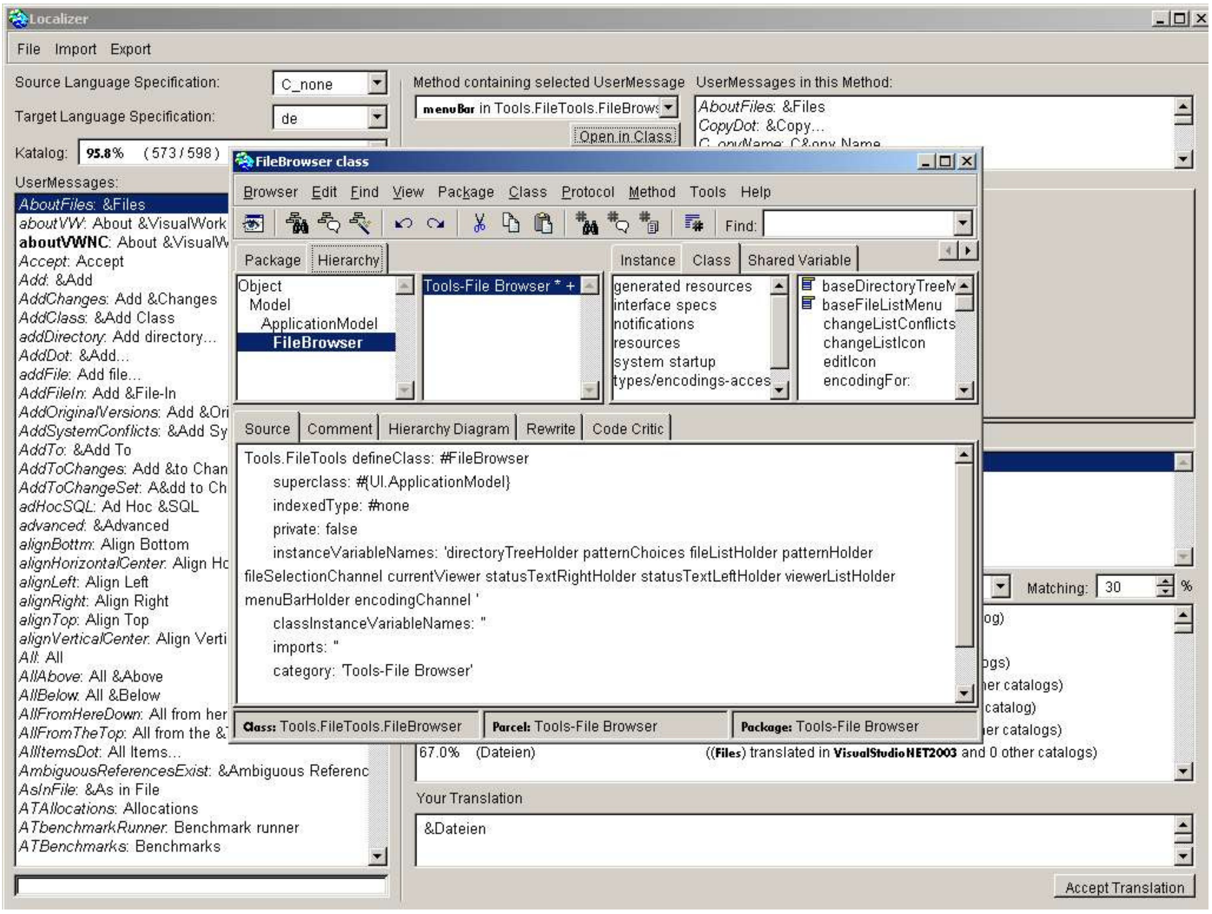
Fig. A.5: Translation memory statistics

The **Import** menu comprises the menu items **Import Catalog**, **Import CSV Catalog**, **Import CSV Catalog Folder**, **Import TM**, **Import TMX** and **Import TBX**. This menu can be used to import glossaries in the LBL file format or in the CSV file format, resident translation memories (formats: translation memory (.tm), Smalltalk file outs (.st), Smalltalk parcel sources (.pst), Smalltalk change files (.cha), Smalltalk parcels (.pcl), workspaces (.ws), text files) as well as TMX files (version 1.1) and TBX files.

The **Export** menu comprises the menu items **Export TM**, **Export TMX** and **Export TBX**. For the export, the same formats are supported as for the import.

In the upper part of the left hand third of the *L10N* window, you can select the source language and the target language, as well as the catalog to be translated. The display indicates how many translation units in the catalog are already translated. Further down is the **UserMessages** window with the translation units. Untranslated units are bold, translated units are italic. Further down is a window for the search for specific keys.

In the upper part of the right hand side of the *L10N* window, the methods containing the current translation unit are displayed. In the window to the right, the other translation units of the corresponding method are displayed. With the **Open in Class** button**,** you can open the class where the method is implemented (see Figure A.6).


Localizer
File Import Export
FileBrowser class
Browser Edit Find View Package Class Protocol Method Tools Help
Find:
Package Hierarchy
Instance Class Shared Variable
Source Comment Hierarchy Diagram Rewrite Code Critic
Open in Class
Accept Translation

Source Language Specification: C_none
Target Language Specification: de
Katalog: 95.8% (573 / 598)
Method containing selected UserMessage: menuBar in Tools.FileTools.FileBrows
UserMessages in this Method: AboutFiles: &Files; CopyDot: &Copy...

UserMessages:
- *AboutFiles*: &Files
- *aboutVW*: About &VisualWork
- **aboutVWNC**: About &VisualW
- *Accept*: Accept
- *Add*: &Add
- *AddChanges*: Add &Changes
- *AddClass*: &Add Class
- *addDirectory*: Add directory...
- *AddDot*: &Add...
- *addFile*: Add file...
- *AddFileIn*: Add &File-In
- *AddOriginalVersions*: Add &Ori
- *AddSystemConflicts*: &Add Sy
- *AddTo*: &Add To
- *AddToChanges*: Add &to Chan
- *AddToChangeSet*: A&dd to Ch
- *adHocSQL*: Ad Hoc &SQL
- *advanced*: &Advanced
- *alignBottm*: Align Bottom
- *alignHorizontalCenter*: Align Hc
- *alignLeft*: Align Left
- *alignRight*: Align Right
- *alignTop*: Align Top
- *alignVerticalCenter*: Align Verti
- *All*: All
- *AllAbove*: All &Above
- *AllBelow*: All &Below
- *AllFromHereDown*: All from her
- *AllFromTheTop*: All from the &
- *AllItemsDot*: All Items...
- *AmbiguousReferencesExist*: &Ambiguous Referenc
- *AsInFile*: &As in File
- *ATAllocations*: Allocations
- *ATbenchmarkRunner*: Benchmark runner
- *ATBenchmarks*: Benchmarks

Hierarchy: Object / Model / ApplicationModel / FileBrowser
Package: Tools-File Browser * +
Protocols: generated resources, interface specs, notifications, resources, system startup, types/encodings-acces
Methods: baseDirectoryTreeM, baseFileListMenu, changeListConflicts, changeListIcon, editIcon, encodingFor:

```
Tools.FileTools defineClass: #FileBrowser
	superclass: #{UI.ApplicationModel}
	indexedType: #none
	private: false
	instanceVariableNames: 'directoryTreeHolder patternChoices fileListHolder patternHolder
fileSelectionChannel currentViewer statusTextRightHolder statusTextLeftHolder viewerListHolder
menuBarHolder encodingChannel '
	classInstanceVariableNames: ''
	imports: ''
	category: 'Tools-File Browser'
```

Class: Tools.FileTools.FileBrowser **Parcel:** Tools-File Browser **Package:** Tools-File Browser

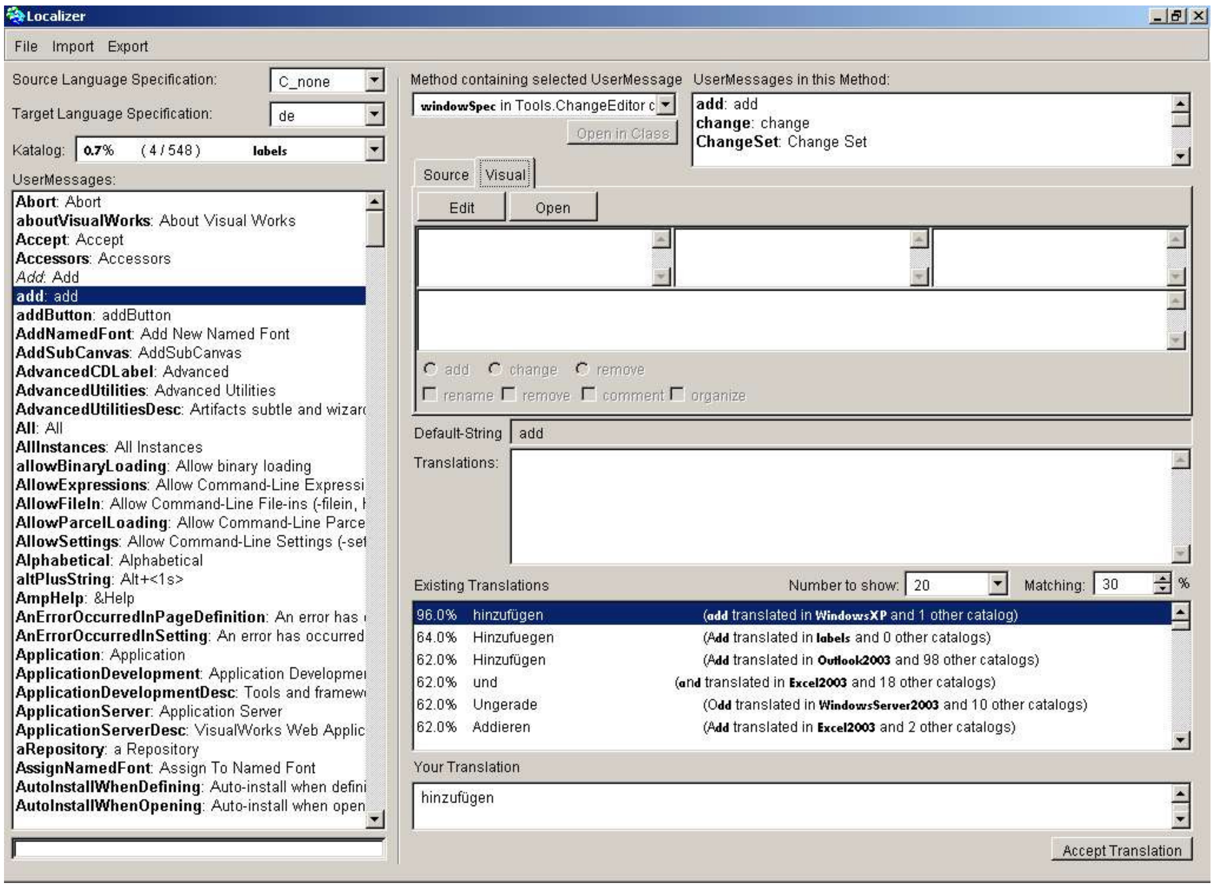
Matching: 30 %

67.0% (Dateien) ((**Files**) translated in **VisualStudioNET2003** and 0 other catalogs)

Your Translation: &Dateien

Fig. A.6: The class where the method is implemented

Further down is a window with the source coding of the method with the current translation unit. The methods created with the *VisualWorks Canvas* tool can also be displayed in a WYSIWYG mode (see Figure A.7).



Source Language Specification: C_none

Target Language Specification: de

Katalog: 0.7% (4 / 548) labels

UserMessages:

Abort: Abort
aboutVisualWorks: About Visual Works
Accept: Accept
Accessors: Accessors
Add: Add
add: add
addButton: addButton
AddNamedFont: Add New Named Font
AddSubCanvas: AddSubCanvas
AdvancedCDLabel: Advanced
AdvancedUtilities: Advanced Utilities
AdvancedUtilitiesDesc: Artifacts subtle and wizar
All: All
AllInstances: All Instances
allowBinaryLoading: Allow binary loading
AllowExpressions: Allow Command-Line Expressi
AllowFileIn: Allow Command-Line File-ins (-filein, f
AllowParcelLoading: Allow Command-Line Parce
AllowSettings: Allow Command-Line Settings (-set
Alphabetical: Alphabetical
altPlusString: Alt+<1s>
AmpHelp: &Help
AnErrorOccurredInPageDefinition: An error has
AnErrorOccurredInSetting: An error has occurred
Application: Application
ApplicationDevelopment: Application Developme
ApplicationDevelopmentDesc: Tools and framew
ApplicationServer: Application Server
ApplicationServerDesc: VisualWorks Web Applic
aRepository: a Repository
AssignNamedFont: Assign To Named Font
AutoInstallWhenDefining: Auto-install when defini
AutoInstallWhenOpening: Auto-install when open

Method containing selected UserMessage: **windowSpec** in Tools.ChangeEditor c

UserMessages in this Method:

add: add
change: change
ChangeSet: Change Set

Default-String: add

Translations:

Existing Translations — Number to show: 20 — Matching: 30 %

96.0%	hinzufügen	(**add** translated in **WindowsXP** and 1 other catalog)
64.0%	Hinzufuegen	(**Add** translated in **labels** and 0 other catalogs)
62.0%	Hinzufügen	(**Add** translated in **Outlook2003** and 98 other catalogs)
62.0%	und	(**and** translated in **Excel2003** and 18 other catalogs)
62.0%	Ungerade	(**Odd** translated in **WindowsServer2003** and 10 other catalogs)
62.0%	Addieren	(**Add** translated in **Excel2003** and 2 other catalogs)

Your Translation: hinzufügen

Fig. A.7: A method in the WYSIWYG mode

In the **Default String** window further down, the current source text is displayed. Further down is the **Translations** window with the current translation and any other former translation. Further information displayed in this window is the date and the time the translations were confirmed.

In the window further down, the translations found in the imported glossaries or translation memories are displayed. You can adjust the number of the matches shown (options: **Show all**, **1**, **3**, **10**, **25**, **50**, **100**) and the minimum quality of the matches (30 per cent up to 100 per cent).

Right at the bottom of the screen is a field where you can enter a new translation, as well as the **Accept Translation** button. With this button, you can confirm a translation and write it in the resident translation memory.

3 Evaluation

3.1 Introduction

L10N is a compact localization tool, which was especially designed for the localization of Smalltalk applications. It has the most important features of a localization tool, like the extraction of the text to be translated from the source coding, the WYSIWYG mode and the leverage function. *L10N* runs on several platforms like *Windows*, *Linux*, *Unix* and *PowerMac*. It supports the translation memory and terminology exchange formats TMX and TBX, which guarantees an uncomplicated data exchange with other translation memory systems and localization tools.

3.2 Advantages

The intuitive user interface makes the tool very user friendly. After a short settling-in period, the tool is easy to use, due to the fact that the tool's features and options are restricted to the essential part. There is a high conformity of the steps performed by the system with the user's expectation.

The status of the translation units and catalogs is quickly perceptible. Untranslated units in the **UserMessages** window are stressed by bold type, translated units by italic type. Furthermore, translated units can be extracted from the user messages list.

A specific feature of *L10N* is the possibility to switch between the original language and the target language within the WYSIWYG mode at all times during the translation process. This function is very useful to keep the consistency within the translation.

An example from the localization of *VisualWorks* is the **Smalltalk** menu in the **Workspace** window. (see Figures A.8 and A.9).

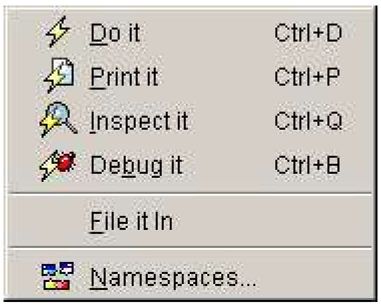


Fig. A.8: The **Smalltalk** menu in the **Workspace** window

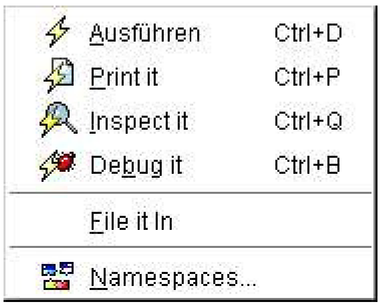
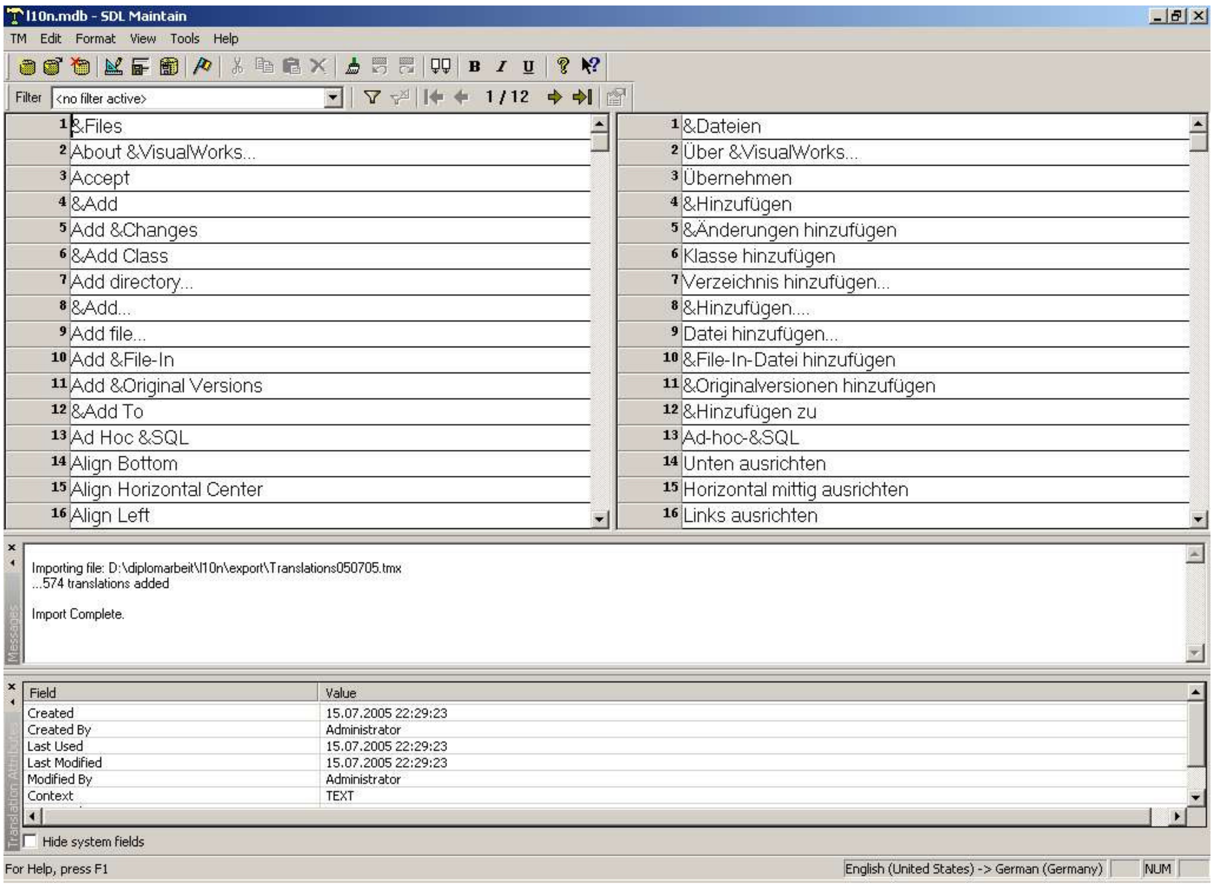


Fig. A.9: The **Smalltalk** menu in the WYSIWYG mode of *L10N*, option: display translation

In this case, the imperative form *Do it* has been translated with the infinitive form *Ausführen*. The sorting of the translation units in the **UserMessages** window is alphabetically and not in the order of their occurrence in the text. Therefore, the menus and dialog boxes can not be translated in a continuous flow. The possibility to switch between the source texts and the translations allows the translator to see, which translation strategy has been used so far, and to adopt it to the other translation units in the same menu or dialog box.

Another advantage of *L10N* is given by the manifold import and export possibilities. So, functions that are not implemented in *L10N*, can be executed with other programmes, for example the maintenance of the translation memory. Because a translation will be automatically written in the translation memory when confirming it, the translation memory usually contains also translation units that are not suitable for further use. The TMX or TBX export allows the data to be exchanged and manipulated with other translation memory or terminology systems (see Figure A.10).



1	&Files	1	&Dateien
2	About &VisualWorks...	2	Über &VisualWorks...
3	Accept	3	Übernehmen
4	&Add	4	&Hinzufügen
5	Add &Changes	5	&Änderungen hinzufügen
6	&Add Class	6	Klasse hinzufügen
7	Add directory...	7	Verzeichnis hinzufügen...
8	&Add...	8	&Hinzufügen....
9	Add file...	9	Datei hinzufügen...
10	Add &File-In	10	&File-In-Datei hinzufügen
11	Add &Original Versions	11	&Originalversionen hinzufügen
12	&Add To	12	&Hinzufügen zu
13	Ad Hoc &SQL	13	Ad-hoc-&SQL
14	Align Bottom	14	Unten ausrichten
15	Align Horizontal Center	15	Horizontal mittig ausrichten
16	Align Left	16	Links ausrichten

```
Importing file: D:\diplomarbeit\l10n\export\Translations050705.tmx
...574 translations added

Import Complete.
```

Field	Value
Created	15.07.2005 22:29:23
Created By	Administrator
Last Used	15.07.2005 22:29:23
Last Modified	15.07.2005 22:29:23
Modified By	Administrator
Context	TEXT

Fig. A.10: Imported TMX file from *L10N* to *SDLX*

3.3 Suggestions for improvement

From the view of the translator, the subdivision into translation units by user messages is not the best solution, because the different meanings of homonyms can not be represented by different translations. Because only one translation per search key can be entered, a translation that reflects all the different meanings of the different concepts has to be found.

An example is the user message **Back**. On the one hand, this message can be found in the toolbar of a so-called **Inspector** and means *zurück* or *rückgängig machen* (see Figure A.11).

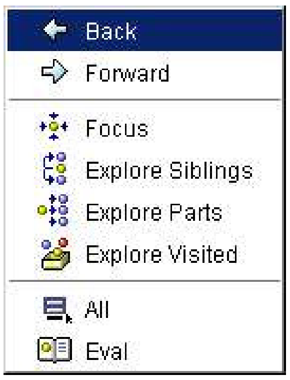


Fig. A.11: Elements of an **Inspector** toolbar (WYSIWYG mode in *L10N*)

On the other hand, the English user message **Back** also represents a menu item in the window menu of any *VisualWorks* window and means in this case as much as *nach hinten* (see Figure A.12).

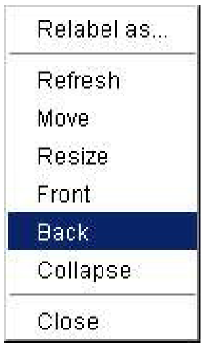


Fig. A.12: Elements of a windows menu

The alphabetical sorting of the translation units in the **UserMessages** window and therefore the processing sequence is not very practical. It would be better to have the possibility to sort and to translate the units accordingly to the order of their occurrence in the text. The translator could then translate all elements of a menu or a dialog box one after the other and would not have to get familiar with another context after each translation unit.

Because the search for translation units is only possible for their search key, but not for their description, not all translation units can be found. An example is the menu item **Class Methods**, which has the search key **ClassDefinitions**. With the entry *Methods*, this user message will not be found, although fuzzy match search is basically supported bei *L10N*. According to the assumption that the translator will not know all the search keys with their descriptions, the possibility to search for the descriptions would be an advantage.

The following improvement suggestions are not as significant as the ones already mentioned.

The **Default-String** window, which is used to display the entire current translation unit, should be adjustable, so that longer translation units can also be entirely displayed.

A check for double-time assigned hot keys is only possible in the WYSIWYG mode, that means, elements, that can not be displayed in the WYSIWYG mode, can not be checked. Therefore, a feature to check the hot keys for their uniqueness would be helpful.

The statistics created by the system only give information about the number of entries in the translation memory (including glossaries) and are therefore little informative concerning the leverage gained by a translation memory. A comparison of the translation units in the current project with the ones in the translation memory or in glossaries would be helpful to find out, whether it pays to include the memory or the glossaries in the project.

There should be a possibility to display translation memories and glossaries that are included in the project and also a possibility to exclude them from the project.

Another improvement would be the possibility to include comments to the translation units, for example for translation alternatives.

3.4 Summary

L10N is a user-friendly localization tool, which is compatible with other tools for computer-aided translation, due to the support of the standardised data exchange formats TMX and TBX. It has been especially developed for the localization of Smalltalk applications. For future versions of the tool, a solution for the treatment of homonyms should be found. It would also be an advantage if the sorting of the translation units could be chosen accordingly to their occurrence in the text. Another improvement could be more extensive statistical information about the project and checking functions, for example for double-time assigned hot keys.