

MULTILIZERTM

G L O B A L L O C A L I Z A T I O N



Language Manager 4.0

Users's manual

MULTILIZER™

Language Manager - User's Manual

May 1999

Copyright © 1999 Innoview Data Technologies, Ltd. All rights reserved.

MULTILIZER is a trademark of Innoview Data Technologies, Ltd.

Delphi is a registered trademark of Inprise Corporation.

C++Builder is a trademark of Inprise Corporation.

Table of Contents

1	Preface	1
2	General	2
3	Running Language Manager	3
4	Creating a new project	5
	Specifying a program source location and type.....	5
	Entering Dictionary project information.....	6
	Specifying a Dictionary type	7
	Selecting languages used in the dictionary	8
	Finishing the Wizard.....	9
5	User interface	10
	The toolbar	10
	Left Flag bar	11
	Comment bar.....	13
	The Status Bar	13
	The Grid	14
	Columns.....	14
	Rows	14
	Cells	14
	Cell Editing	15
	Using the cell area	15
	Edit Windows	15
	Character composing.....	16
	Customizing the interface	16
6	Adding Strings to the Dictionary	18
	Updating the Dictionary	18
	Scanning.....	18
	Defining scanning options	19
	Excluding and including files to be scanned.....	20
	Source File scanning	21
	Form File scanning	22
	Application File scanning	23
	Resource Files scanning	23
	Text Files scanning.....	24
	Including special-purpose strings into dictionary	25
	Printing the dictionary	26
	Using Glossaries	27

	Custom glossary	27
	Read only glossaries	27
	Retrieving translations	28
	Opening the Custom Glossary.....	28
7	Inputting characters.....	29
	Character composing	29
	Keyboard layout changing.....	30
	Localized Windows versions	31
8	Interchanging dictionaries	32
	Exchanging a Dictionary.....	32
	Importing a Dictionary	36
	Starting the Import Wizard.....	36
	Specifying import properties	36
	Exporting a Dictionary	37
	Starting the Export Wizard.....	38
	Defining the Dictionary type.....	39
	Appendix A: Language Manager Command line	41
	Appendix B: Language Manager Short cuts	42
	Appendix C: Compose character sequences	43
	Appendix D: Inputting non-ASCII characters	44

1

Preface

The purpose of this manual is to familiarize you with Language Manager, which ships with MULTILIZER. To obtain the most precise definitions on use of components and technical details, please refer to the on-line help.

The following issues can be found in **intro.pdf**

- Typographical conventions used in this document.
- General information on localization
- How does MULTILIZER™ work, what is Language Manager for?
- What is the 'Native' language used in Language Manager

2

General

Language Manager is an easy-to-use utility for creating and maintaining dictionaries used by MULTILIZER components.

The dictionaries contain information on

- languages to be supported in the application,
- locale information and
- the translation data.

The translations are the equivalents of the string type properties in the program source. The translation data forms the central part of the outlook of Language Manager: when a new dictionary has been created, you automatically get into the translation editing view of Language Manager. This is natural, because many other types of information are gathered automatically by Language Manager.

The area where the translations are edited is later in this document referred as the 'string grid'.

Language Manager uses standard file formats

Standard file formats and databases are used in MULTILIZER. Therefore editing translations, language and locale information can be done without it. However, it is strongly recommended that you use Language Manager, because it automatically formats the data in the way MULTILIZER components use it.

If MLD dictionary format is used, Language Manager must be used. This is the only non-standard file format. This format contains all the information needed by MULTILIZER components in a single file. It is also optimized for speed in its only task: to provide localized software with translation and locale data.



All screen-captured images have been taken when the active language of Language Manager is English. Set to English by clicking the Earth-image on the left side of the Language Manager's tool bar.

3

Running Language Manager

Language Manager is used whenever new dictionaries are created or the old ones are maintained. To provide flexibility in use, Language Manager can be run in two ways, either

- from command line or
- in interactive mode.

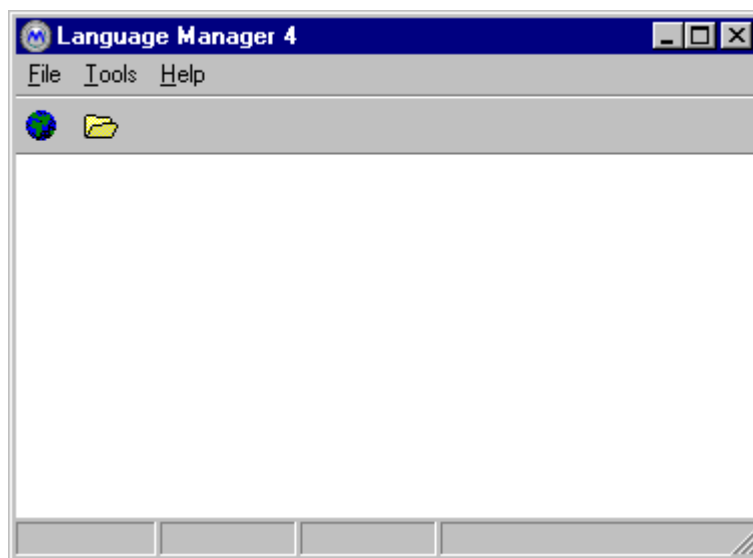
Interactive mode is the suggested way of using Language Manager. This brings all the convenience and functionality of Language Manager to the user. Furthermore, command line functionality is very limited.

Language Manager is run by simply double clicking it in the MULTILIZER program group.



If you run Language Manager from the command line, you must have run it at least once, creating the LMP file. For command line options, see Appendix A:.

When you start Language Manager in interactive mode, you get the following view:



Language Manager without any project loaded.

When you run Language Manager in interactive mode, you can choose between the following options.

- You can change the active language of Language Manager (click the Earth image)
- You can open an existing dictionary project or a dictionary file (File | Open... or click the Folder image)
- You can create a new dictionary (File | New...)
- You can open the custom glossary (File | Glossary).
- You can connect to a MDS Dictionary (Tools | MDS Manager...)

The following chapter explains how to create a new Dictionary.

Opening an existing Dictionary Project is done simply by locating the LMP file and opening it.

You can also open the Custom Glossary. This dictionary contains translations which can be used for automatic translations of words in new projects. Updating words in this dictionary saves your work in future projects.

If you connect and use a MDS Dictionary, you need MULTILIZER Dictionary Server. This is an extension which enables the use of a centralized dictionary on an NT server. MDS provides the client applications with translation data through Internet/Intranet. Language Manager is used for remote maintenance of MDS Dictionaries.



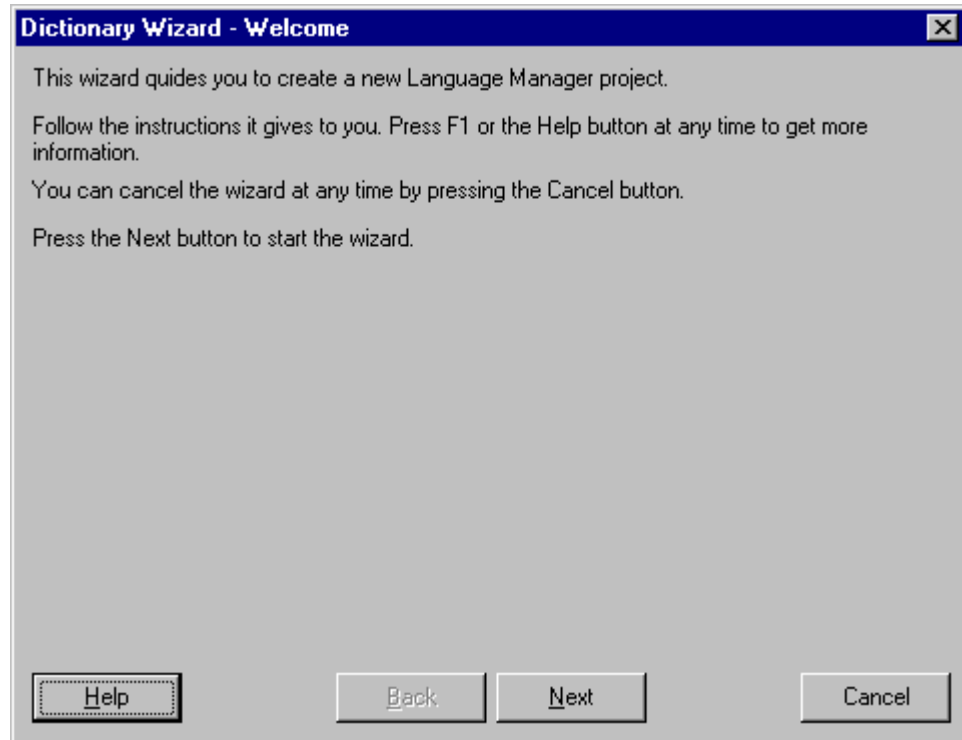
For more information on MDS, go to <http://www.multilizer.com>. There you will find documentation on MDS and a free downloadable demo version of it.

4

Creating a new project

If you start with localizing a new project, you have to create a new Dictionary project. The Dictionary includes language and locale information and translation data.

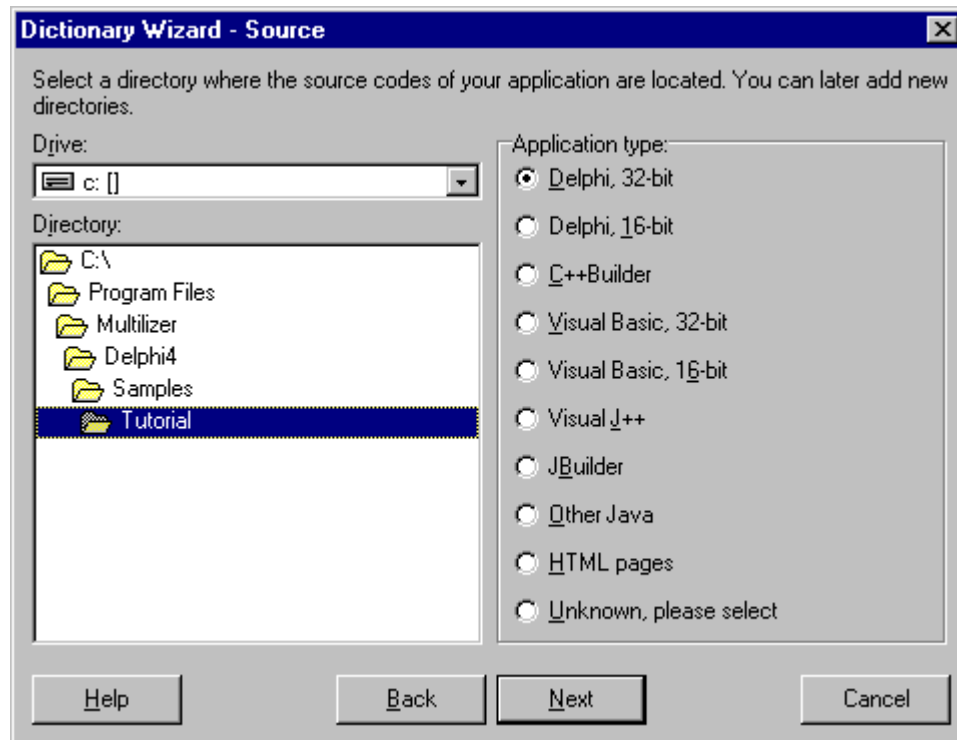
You create a new project by choosing 'New' from the File menu. A Project Wizard starts, which helps in creating the files or database tables needed by MULTILIZER components.



Starting the Dictionary Wizard

Specifying program source location and type

The first phase in creating a project is to select the source directory. You specify the directory where the source code is located. Additional source directories can be defined later.



Selecting the directory of the program source and application type

In the same view as where you choose the directory, you define the application type. Since Language Manager is used in all MULTILIZER Editions and programming environments, there are many possible application types that it detects.

The possible types are:

Value	Description
Delphi 32-bit	a 32 bit Delphi 2, 3 or 4 application.
Delphi 16-bit	a 16 bit Delphi 1.0 application.
C++Buider	a C++Builder 1, 3 or 4 application.
Visual Basic, 32-bit	a 32-bit Visual Basic 4.0, 5.0, 6.0 application.
Visual Basic, 16-bit	a 16-bit Visual Basic 4.0 application.
Visual J++	a Visual J++ 6.0 application.
JBuilder	a JBuilder application or applet.
Other Java	a Java application or applet. Use this for all Java projects but WFC and JBuilder
Resource bundle	a resource bundle of Java.
Unknown, please select	source type not recognized.

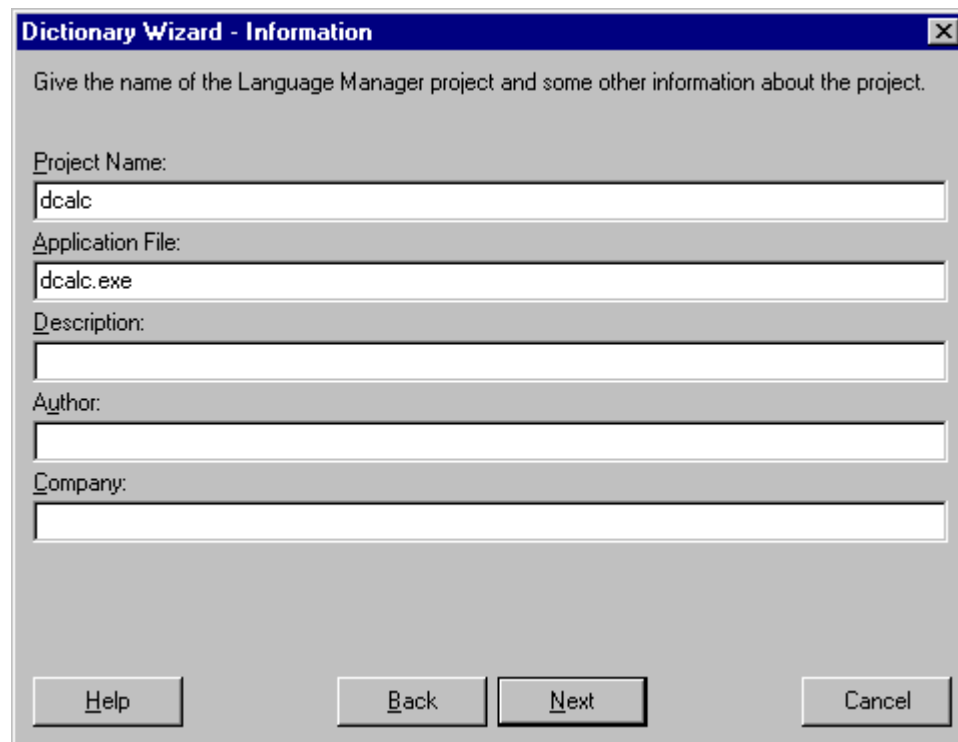


Language Manager automatically detects what kind of source is located in the directory specified. If the detection fails, you can force the type to be one of those mentioned above.

Entering Dictionary project information

The next screen is for entering information on the project. This information is useful for later identification of the dictionary project and the dictionary to be created.

You always have to enter the Project Name and the Application executable.



Entering project information

Application file information is needed for extracting the resource strings from within the source code. These strings don't exist in your source code, but they are included in the application on compiling it.

Specifying Dictionary type

The next step is to define the dictionary type. This can be one of the following:

The Dictionary type sheet lets you choose the type of dictionary.

MULTILIZER has four dictionary types:

- binary dictionary
- text dictionary
- database dictionary
- server dictionary

Binary and text dictionaries are stored as files on your computer. No database support is needed when using them. On compiling the software these dictionaries can be either

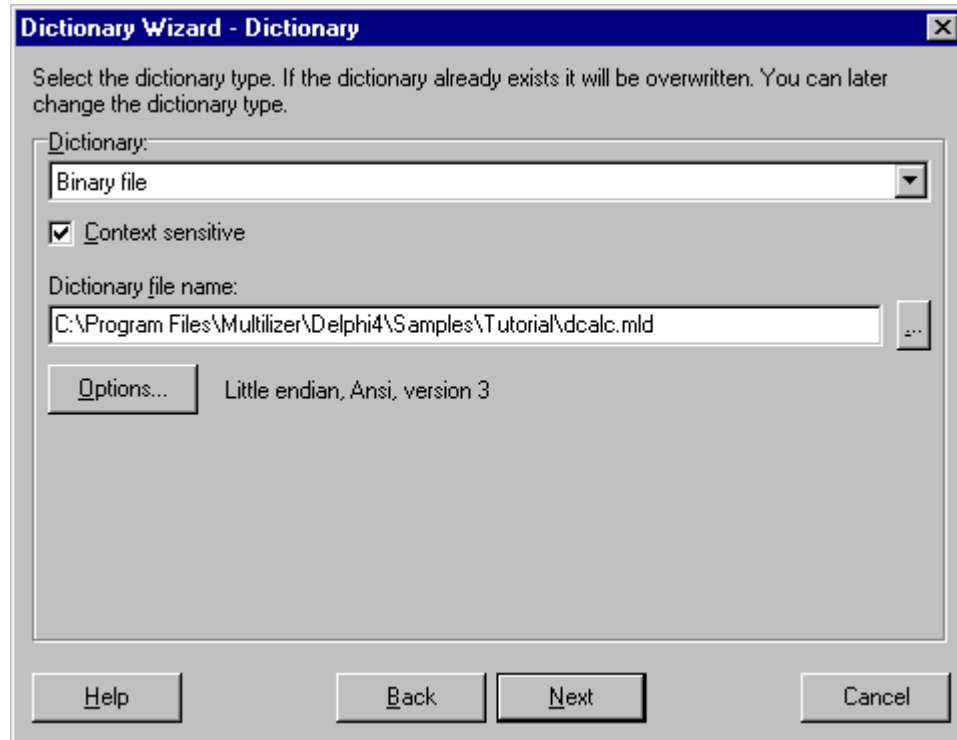
- external files, which are accessed from within the localized software, or
- the dictionaries can be embedded into your software. This feature makes it possible to create on-file multilingual applications.

Text files can be either standard 8-bit text files (ANSI) or Unicode files.

The database dictionary uses data stored in the database tables. The server dictionary uses data stored in a MULTILIZER Dictionary Server. A database can be either a desktop database or an SQL database.



You are able to embed binary or text dictionaries into the application on compiling it, making it possible to create a one-file multi-language EXE, where you can change the language on run-time.



Selecting the dictionary type



Depening on the database configuration of your system, the dictionary type to choose may vary from the picture above.

If you use an MDS Dictionary you need MULTILIZER Dictionary Server. This is an extension which enables the use of a centralized dictionary on an NT server. MDS provides the client applications with translation data through Internet/Intranet. Language Manager is used for remote maintenance of MDS Dictionaries.

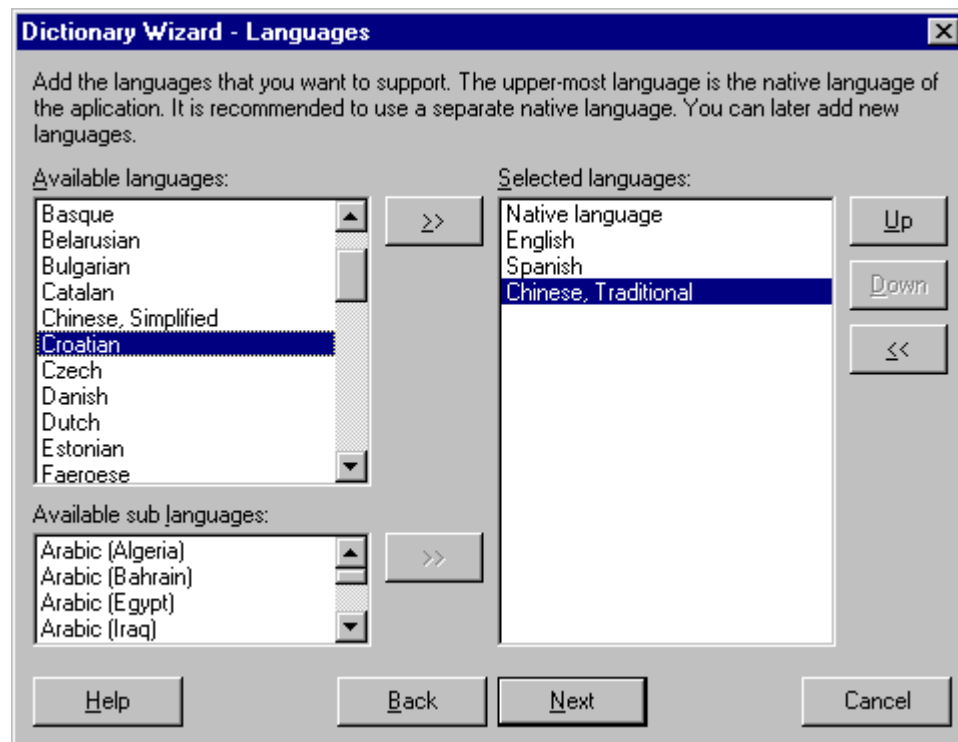


For more information on MDS, go to: <http://www.multilizer.com>. There you will find documentation on MDS and a free downloadable demo version of it.

Selecting languages used in the dictionary

After you have specified the Dictionary type, you proceed to the language selection view. In this view you specify which languages to include in your project.

You don't have to know at this phase which languages you might add in the future. Whenever you want, you can add a new language to project.



Selecting languages and sub languages.

There is one language selected by default: Native language. This is the language which is used in your software development. Even it's not necessary to use it, we strongly recommend the use of it.

There can be both languages and sub languages selected. The difference is that a sub language is country specific. E.g., British English differs from US English. Therefore you can specify which language to use. Of course, you can select both English versions for your project.

Finishing the Wizard

The next step is to scan the source files and create a new dictionary with the attributes specified earlier – press Finish. Now your application source is being scanned, in order to find the strings to be translated.

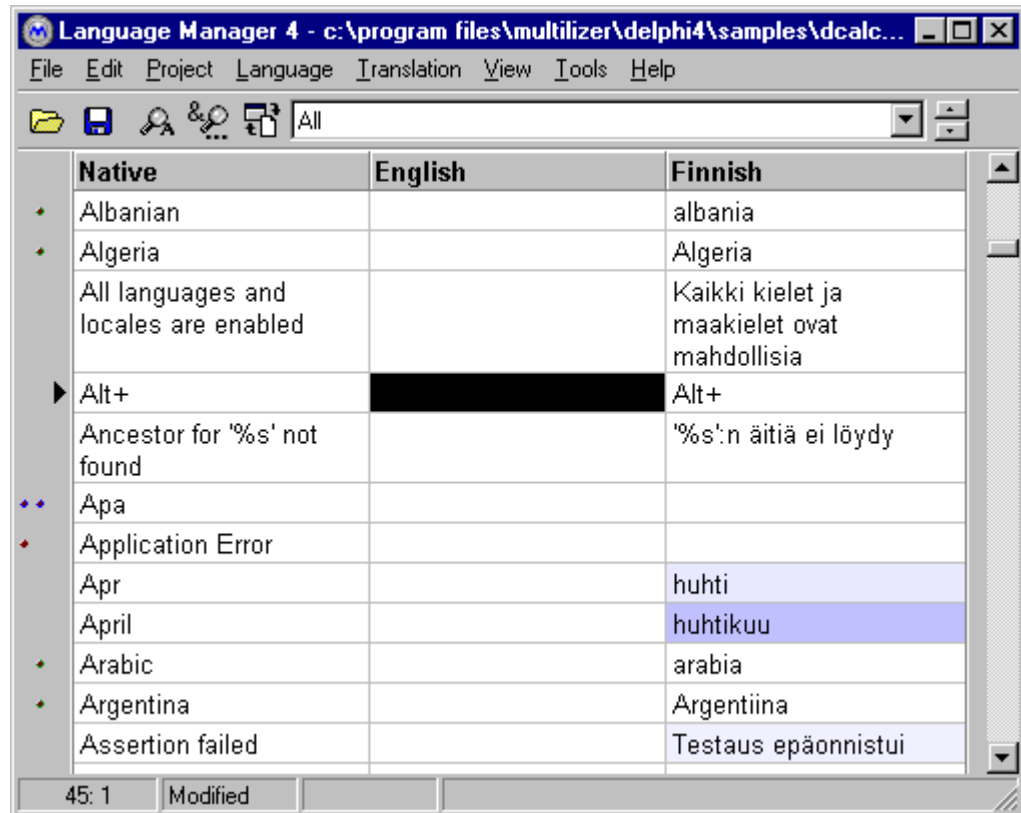
The dictionary will be created including the strings from the program source. In addition Language Manager adds information on the languages included and locale specific information.

You can now proceed to editing the translations. The following chapter familiarizes you with the Language Manager interface.

5

User interface

Language Manager is designed in a way that is as easy as possible to use. Besides automating common tasks, the remaining work is done easily with Language Manager.



The workplace of Language Manager.

The central parts of the user interface are:

- Tool bar
- Left Flag bar
- Right Flag bar
- Status bar
- Grid

The toolbar

The toolbar contains a number of speed buttons. These let you to apply the most needed functions. The buttons are as follows:



Open a Language Manager project.

Corresponds to **File | Open...** The shortcut **Ctrl+O** does the same.



Saves the current Language Manager project.

Corresponds to **File | Save**. The shortcut **Ctrl+S** does the same.



Opens the Find dialog, where you can specify a string to be searched from this dictionary.

Corresponds to **Edit | Find...**. The shortcut **Ctrl+F** does the same.

Choosing Edit | Find Again you can perform the same search again. F3 does the same.



Finds the next incomplete translation in the current language.

Corresponds to **Edit | Next Incomplete Item...**. The shortcut **Ctrl+I** does the same.



Scans all the form and/or unit files found in the source directories, adding new strings to the dictionary. For further information, see *Updating the Dictionary*, page 18.



The combo box lets you filter out the strings to be shown. If this is *All* then all the strings of the dictionary are visible. If you select any other, only the strings belonging to the selected group/form/dialog are shown.

You can, e.g., choose 'Dictionary Languages', which lets you see the language names included in the dictionary.

This feature is very useful in big projects. Furthermore, if the same string exists in several places in the program source, the hint showing the place may become too big. In this case it's very easy to select only a certain part of the strings to be shown.



Steps to the next or previous page.



Shows or edits the page information.

Left Flag bar

There is a flag bar located on the left margin of the Language Manager's workplace. This shows properties of the string extracted from the program source and its corresponding equivalents in the other languages supported.

The Flag bar is intended to quickly show at least the following:

- If all the translations for that particular word are ready.
- If the word is included in the Master Dictionary.
- If there are words in the dictionary which cannot be found in the program source.
- You can access a pop-up menu containing row-specific commands
- You get information on the native string's location in the program source.

The following flags can be found on the flag bar:



Two blue flags indicate that the native language column's string is not found in the source code. It is likely that the application doesn't use the string any more.



The green flag indicates that the string exists in the Master Dictionary.

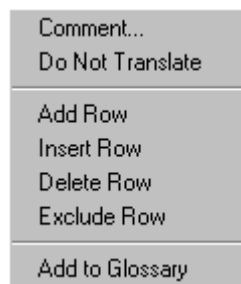


The row is incomplete. You have not filled in all the cells of the row, i.e., the translations are incomplete.

You can specify the MissingTranslation property in your application's Dictionary component in order to handle a missing translation. By default the native language string is used in case of a missing translation.

Pressing the right mouse button on the flag bar, you get a popup menu with row-specific commands. You can choose between:

- Adding/removing a comment in the *Comment bar*.
- Specifying that the row must not be auto-translated. When applying this, the row appears in gray. During the automatic translation process, this line will be skipped.
- Adding an extra line to the end of the dictionary.
- Adding an extra line to the current position.
- Deleting the current row.
- Excluding the current row.
- Adding the row's contents to the Master Dictionary



Pop-up menu for row-specific tasks



The difference between deleting and excluding a row is that excluding a row does not remove it from the dictionary. Delete the row and you lose the translations.

If you move the cursor on the flag bar, Language Manager pops up a hint window with the native string's location in the program source. (Note! Some strings may be found in several places in the source)

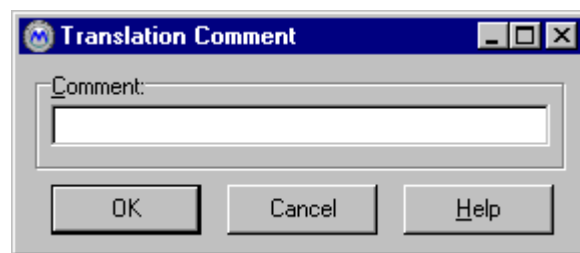
Algeria		Algeria
All languages and locales are enabled		Kaikki kielet ja maakielet ovat mahdollisia
MainForm: c:\program files\multilizer\delphi4\samples\dcalc\main.dfm, TMenuItem: AllMenu.Hint		
▶ Ancestor for '%s' not found		'%s':n äitiä ei löydy
••• Apa		
• Application Error		

A hint showing where the native string resides in the program source.

Comment bar

In addition to the Left Flag bar, there is a flag bar located on the right margin of the Language Manager's workplace. This shows custom comments made by the user. This bar is called *Comment bar*.

Custom comments can be input by pressing the right mouse button on the flag bar on the left margin. The following dialog appears.



Entering a comment for the translation

On the comment line you can write whatever comment you want. If you want to remove a comment you just need to clear the comment and press OK.

Once you have added a comment for the translation, a red dot appears on the comment bar. If you move the cursor on the comment bar, a hint appears showing the comment.

	maakielet ovat mahdollisia	
	Alt+	
	'%s':n äiti	This is a comment

A custom comment

The Status Bar

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

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

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



The Status bar

The Grid

The Grid is the area of Language Manager where editing takes place. The grid consists of columns and rows. The intersection of a column and a row is called cell.

Columns

The native strings extracted from the program source are in the leftmost column. This is called the native language column. Subsequent columns contain translations of the native language.

If you have built a new dictionary using the Project Wizard, only the leftmost column contains strings. This is due to the fact that no translations have been done yet.



TIP!

Normally, when you have started a new project and the Project Wizard has finished, you would do automatic translation. In automatic translation Language Manager looks for translation data in the Master Dictionary. If it finds translations that suit your languages and native strings, it fills in the translations. To do automatic translation, choose **Project | Translate | Using Glossaries** from the menu.

Rows

Each row contains one string extracted from the program source and the translations of it.



TIP!

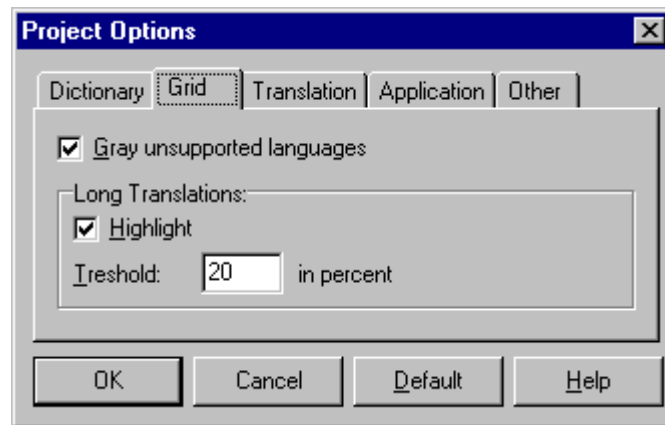
If you go to the last row and press the down arrow key, you are able to insert a new native string. This is useful if there are strings in your program source which you won't let Language Manager see, but you want to include extra string data in the dictionary.

Cells

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

Cell colors are used to indicate some central string related issues. Color coding is applied to the cell background in the following manner:

- Normally the cell color is white.
- If the string length exceeds a user-specified amount, the cell background gets blue. The more the length exceeds the native string's length the darker the blue is shown.
- If the system can't input the character used by the language, the cells of the language are grayed. For example you need the Japanese version of Windows to input Japanese.



In the Project options dialog box you can specify whether to use color coding to show cell properties.

If you don't want to use color coding or you want to customize it, you can do it by selecting **Project | Options** and setting the grid properties.

Cell Editing

To edit the contents of a cell, you can either use the cell area or you can open a window for cell editing.

Using the cell area

Strings are handy to edit directly in the cell area. You can start editing the contents of a cell by double clicking the cell with the mouse or pressing the **F2** key.

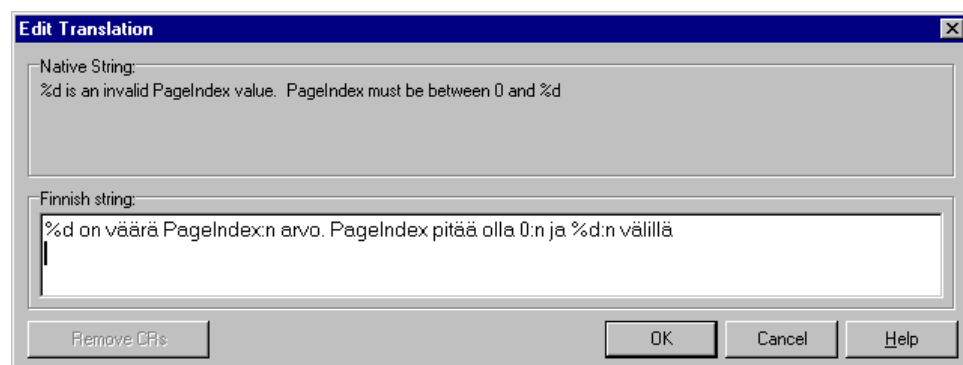
You can stop editing the contents of a cell by clicking outside the cell with the mouse or by pressing the **F2**, **UP**, **DOWN** or **TAB** key.

Pressing **ENTER** stops editing and moves the cursor to the next cell if the native cell does not contain line feeds. If it does, then pressing **ENTER** adds a line feed to the cell.

Edit Windows

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 so easily by opening an edit window. You should also use it if you need to add line breaks to the text.

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



Long strings are easily edited in an edit window.

Character composing

Typing accented characters can be a problem, especially if an English language keyboard is used. Even in non-English, language-specific keyboards, only a subset of the characters can be typed. Often that involves the use of 2 or 3 keystrokes. Another approach is to copy and paste from the Windows Character map or to type the character's code from the numeric keypad with Alt-codes.

Language Manager includes the possibility to 'compose' characters by typing very clear and easy-to-remember characters.

You get into character composing mode by pressing the right Control key. Then you enter two characters in order to obtain the special character. For example, push right control and then 'c' and ','. You easily get the 'ç', often needed in French and Portuguese.



If you don't remember these sequences, please refer to **Appendix C**. In addition you can check them in Language Manager from **Tools | Compose strings**. Here you can customize the compose character sequences, as well.

Customizing the interface

By default all languages of the project are shown on the screen. The columns are always of equal width. This means that in cases where there are many languages, the columns are narrow, which may make it difficult to see what they contain.

In order to make the columns broader, you can hide language columns not needed at the moment. This can be done in the following two ways:

1. Choose the columns to view from the View 'menu'.
2. Press the right mouse button above the column header. A pop-up menu appears and lets you choose what to do.



Columns to be shown can be chosen from a pop-up menu.

6

Adding Strings to the Dictionary

Working with dictionary files apart from the programming project provides many clear advantages. One of the strengths is the possibility to store translations in various formats for later use. In addition, you can decide whether to keep the translation data in standard text files, in MULTILIZER binary files or in database tables.

This chapter describes the use of the dictionary files in the context of adding data to a dictionary. It introduces the scanning process and including special purpose strings.

The major part of this chapter covers the methods of storing data in the dictionary. The last part covers the use of the Master Dictionary to speed up and automate part of the translation process.

While this chapter focuses on working with one dictionary, the next chapter will cover the possibilities of interchanging dictionary data between several persons and dictionaries.



You can centralize the dictionary using the Dictionary Server. The Dictionary Server is a separate product which makes it possible to share the same dictionary between many users. Users may then be anything from software developers to client applications using the data. This tool enables the possibility to work in workgroups .

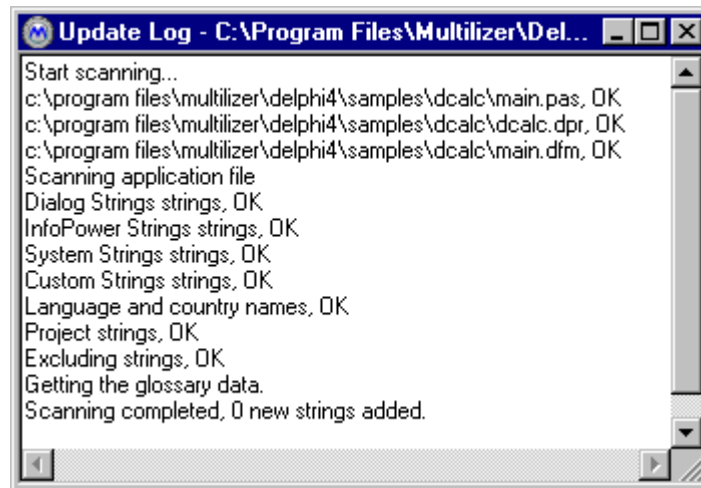
Updating the Dictionary

You have to ensure that the dictionary is up-to-date. This means that when you have made changes to the program source, Language Manager must be told to extract the new strings. This is done easily by

- pressing the scan button on the tool bar or
- choosing **Project | Update...** from the menu or
- applying the shortcut **Ctrl+U**.

Scanning

Once scanning has been started, the following window appears on the screen showing how the scanning is proceeding.

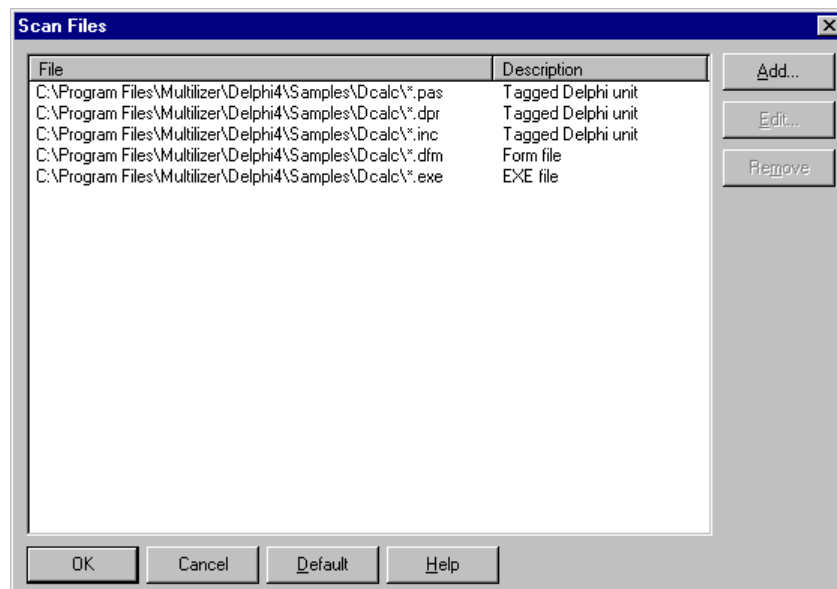


Window showing the dictionary update process.

By default Language Manager scans all the strings to be translated, using the default scanning options. The next chapter describes how you can customize the scanning of your program source.

Defining scanning options

You can define the scanning options by choosing **Project | Scan Files...** from the menu. It opens the following dialog.



Defining File Scan properties.

Using the scan options dialog, you are able to define which files shall be scanned in your program source code. For each file – or file group defined by a wildcard – you can also define file-type specific scanning properties.

New files to be scanned are added with the Add button. The following chapters give an insight on how to have Language Manager scan:

- Source code files,
- Form files,

- Program executables,
- Resource files, and
- Text files

The scanning properties of existing source file(s) can be modified with the **Edit...** button or removed with the **Remove...** button.

You can include single file(s), groups of files and whole directory structures to be scanned. You can also prevent files from being scanned.

By defining the scanning properties you can have Language Manager add almost anything to the dictionary. Besides the default scanning properties, which are sufficient for standard projects, you can customize the scanning to cover resource files and even your own text files.

These unique features make Language Manager a powerful tool in projects containing many non-standard solutions for keeping strings to be translated. The outstanding maintainability of string resources keeps the maintenance easy, even in the most complex projects.



Depending on the project type, i.e., your programming environment, it may be that not all buttons are active.

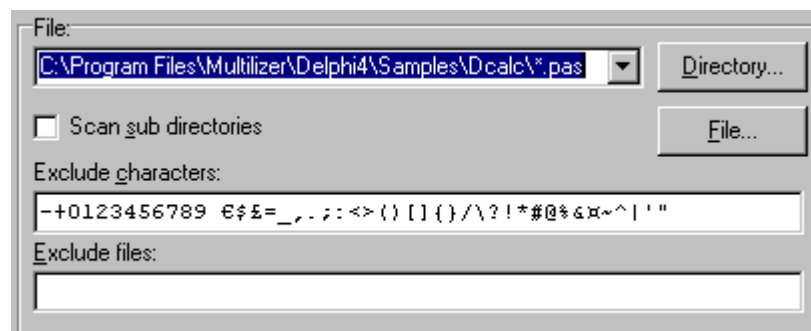
Excluding and including files to be scanned

By default, on specifying files to be scanned, you can specify any of the following:

- one file can be included,
- you can add all the matching files of the directory, or
- you can add all the matching files of the directory and its sub-directories.

into the scanning process. This is normally the most straightforward way to do it.

However, you may need to exclude files from the scanning process. Or you may not want all the strings to be included in the dictionary. For this, every dialog where you define files to be included, you are also able to define exclusions.

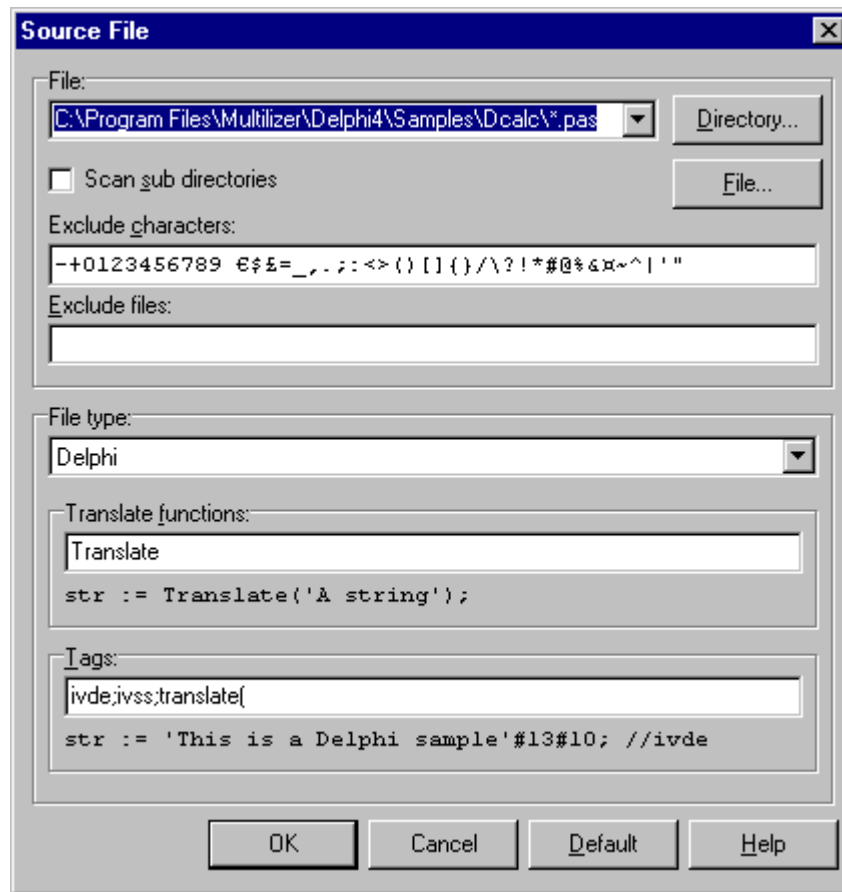


Excluding files and strings to be scanned

If you define excluded characters, strings containing only these are not included in the dictionary. By defining excluded files, you can prevent the scanning process from scanning them at all.

Source File scanning

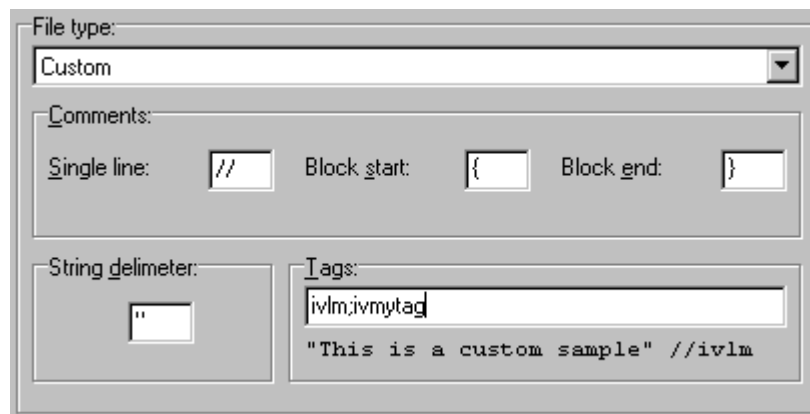
In the Source File dialog, you define the source code files of your application. You can use the **Default** button to get the default values for the properties you most likely need.



Source File dialog

By pressing the **Directory** button and choosing the directory where the source files reside, you can add a group of files to be scanned. You can even specify the sub-directories to be scanned by checking the Scan sub-directories option. The **File** button is used for adding one extra file to be scanned.

By default the File type matches the type of your project. Besides modifying the source code syntax-specific properties, you can enable custom source code type scanning here.



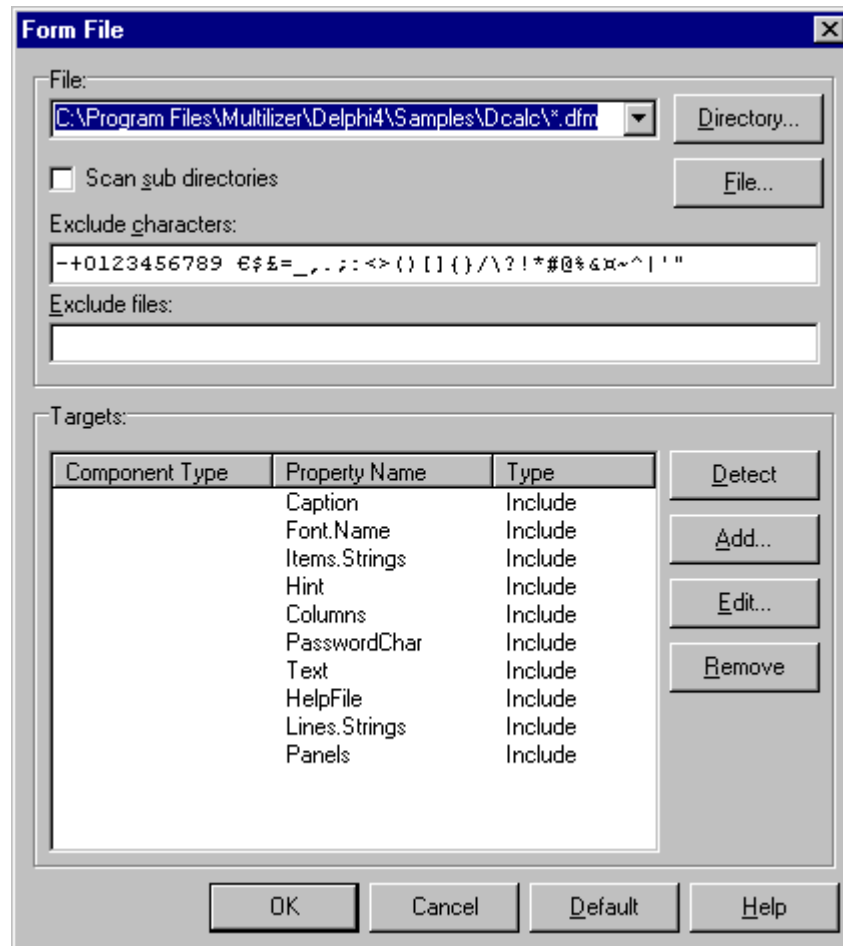
Define custom source file scanning

Remember to define the file extension at the top of the dialog.

Form File scanning

In the Form File dialog you specify which visual component properties shall be scanned by Language Manager. Specifying this information is of great importance whenever you work with 3rd party components which have non-standard string-type properties.

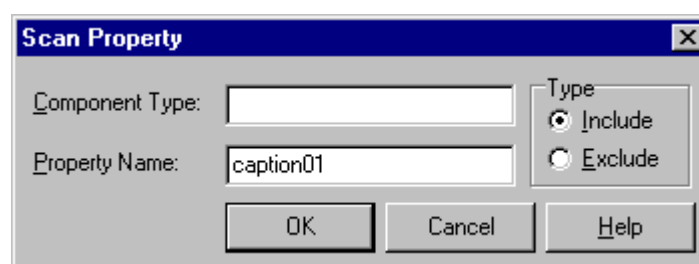
You can have Language Manager detect the components with string-type properties and have them added in the scanning process.



Form File dialog

In addition you can manually add or remove components from the scanning process. Existing targets can be edited by choosing the target and pressing the Edit button. This lets you edit the Scan properties.

Example: there are components with multiple strings with names like caption01, caption02, caption03. In such cases you have to specify the additional property names to be scanned – press the **Add** button.

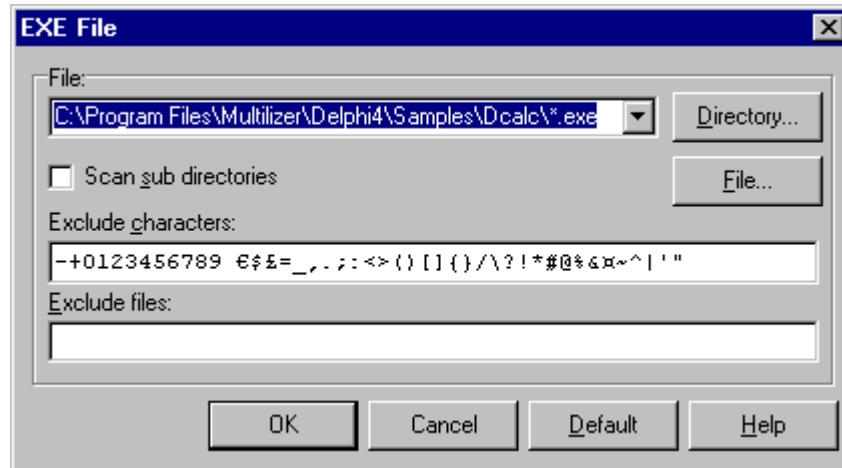


Specifying a new property to be included in the scanning process.

This example shows that Language Manager is able to scan properties of any 3rd party components.

Application File scanning

In the Application File dialog, you can specify whether you want to have the Resource Strings scanned from within the program executable. For this option you must have the software already compiled.



EXE File dialog

VCL Only

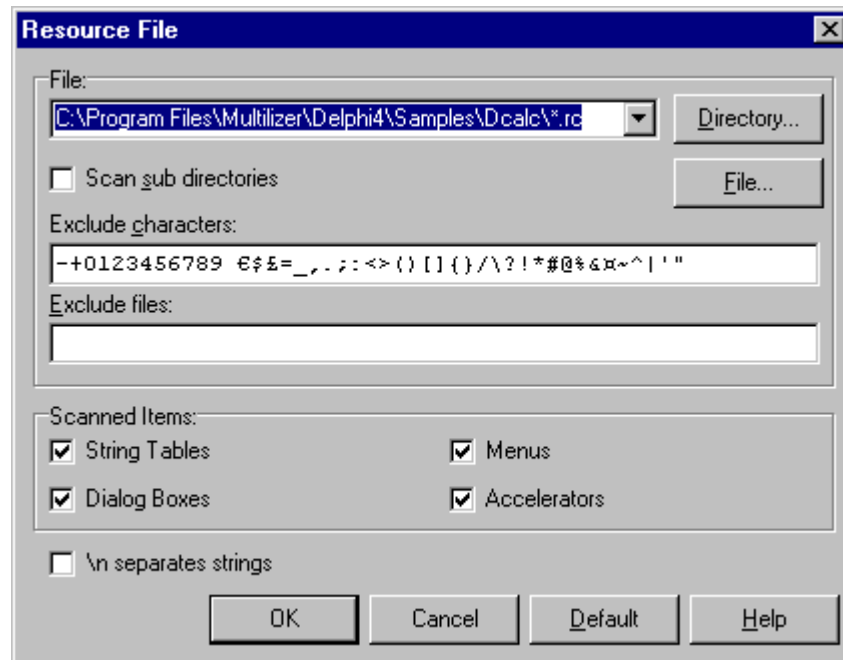
Resource Strings are used for displaying the BDE, system and exception messages and certain standard button captions.



Using this feature you are able to get everything translated in your software. You no longer have to use *lvMessageBox*; you can use the VCL standard ones, such as *MessageDlg*. These get translated without any programming – just complete the translations with Language Manager.

Resource Files scanning

Language Manager also handles scanning of resource files. The Resource File dialog lets you specify what kind of resources are scanned. The resource files contain resources used by the software. Some of the resources are of string type: strings, dialog boxes, accelerators and menus. Since they need to be translated, you can have Language Manager scan these files for you.



Resource File dialog

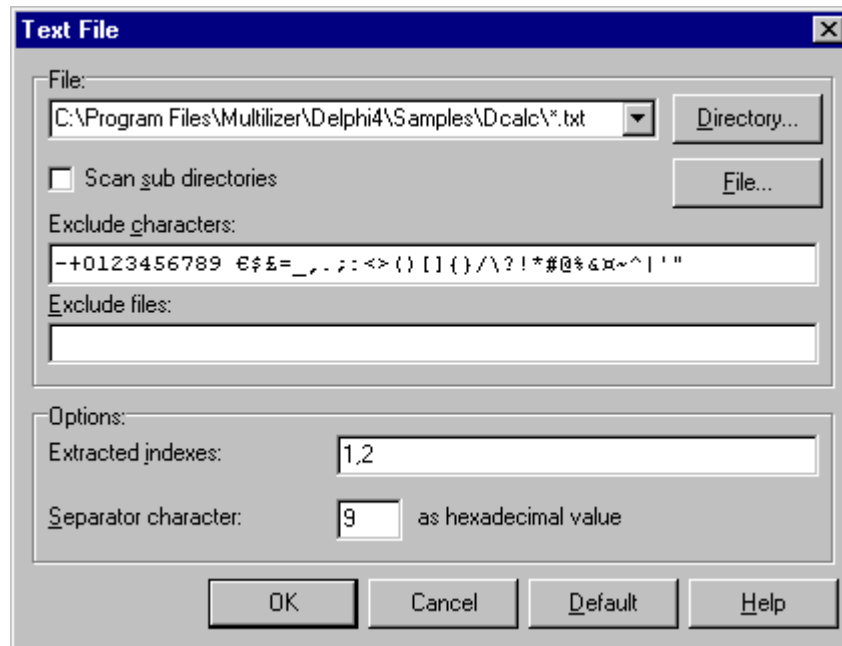
You can separately turn on and off scanning of resource items. The following table contains the possible items:

Item	Description
String Tables	STRINGTABLE strings
Dialog Boxes	DIALOG strings
Menus	MENU strings
Accelerators	ACCELERATORS strings

If '\n separates strings' is checked the \n character inside a string starts a new string. For example "First string\nSecond string" contains two strings: "First string" and "Second string".

Text Files scanning

In the Text Files dialog, you can define text files to be scanned.



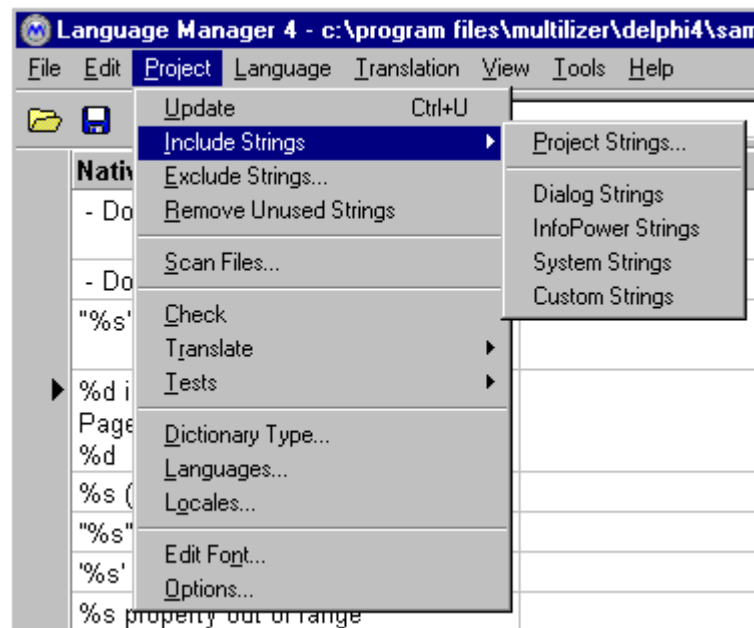
Text file dialog

If the 'Tab ends the string' check box is checked, the tab characters end the string. For example a line "Sample<tab>text" has only Sample added to the dictionary.

Including special purpose strings in the dictionary

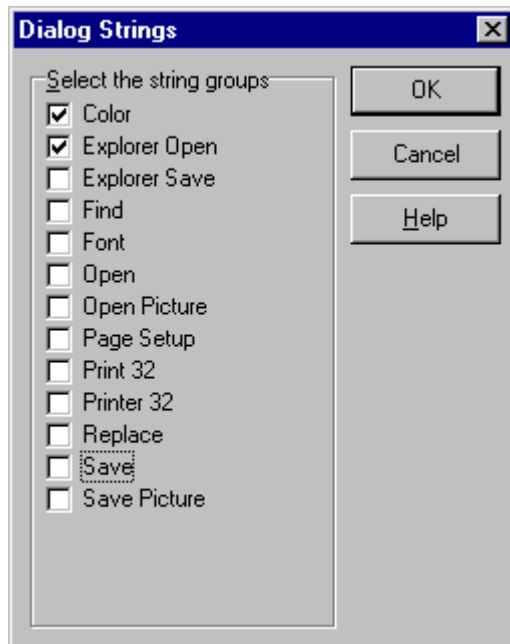
It may happen that not all the strings are added to the dictionary after the first scanning. You may also need to maintain separately a group of strings which you need to have included in your software.

For the purposes mentioned above, in Language Manager there is the possibility to add strings pertaining to a certain logical group.



Including special purpose strings in the dictionary

If you choose e.g. Dialog strings, you get the following Dialog, which lets you add all the strings encountered in the dialog that you choose.



Standard Dialog strings

The actual dialog names may vary depending on your application type.

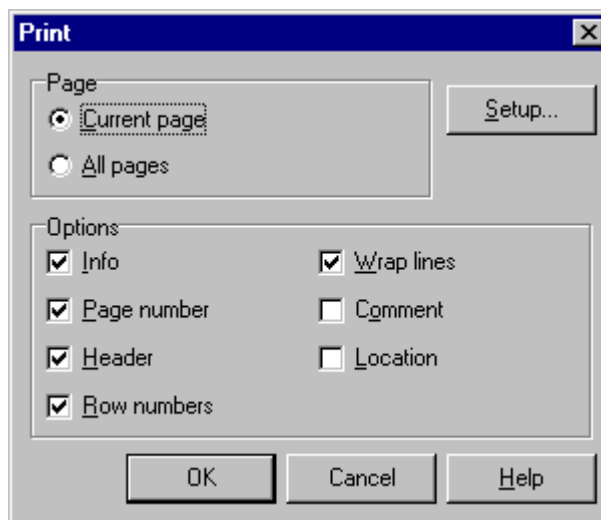
These additional included strings are defined in the Language Manager program directory in *.lms files.



You can easily define your own string groups for any third party component. Language Manager lets you then add the strings to the dictionary. To do this, you can either create a new *.lms file or edit any of the existing ones. The syntax of these files is described in the Language Manager help. It is also easy to modify an existing one and get the data from it.

Printing the dictionary

Language Manager lets the user print the contents of a dictionary. Choosing Print from the File menu brings the following dialog:



The 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 whole dictionary.

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

Using Glossaries

With Language Manager you are able to maintain glossaries. These are files or database tables that contain translation data. Glossaries are intended for storing your translations for later use. The more translations you do, the more there is translation data accumulated in the glossaries. Starting with a new project you can very easily use the data saved in the glossaries.

Language Manager contains two kinds of glossaries: Custom glossaries and Read only glossaries.

Custom glossary

The custom glossary is a glossary that contains your own glossary items. You can add new items and languages to the glossary. You can also remove items and languages from the glossary. The custom glossary strings are saved in local files or in a database tables.

By default the custom glossary is an empty local file. You can add new items and languages to the custom glossary in the following ways.

Storing translations

You might have a string in your dictionary that could be used in another dictionary as well. To make this possible, move it to the custom glossary by:

- choosing Edit | Add to Master,
- pressing Ctrl+M, or
- moving mouse on the gray area on the left side of the row and pressing the right mouse button. Choose Add to Glossary.

This adds the native string and translations from the line where you are at the moment to the Custom Glossary.

If the gray area contains a green dot, the string already exists in the custom glossary. Then there is no need to add it.

**MORE INFO**

For additional information on using the custom glossary, see the online help topic "Custom Glossary".

Read only glossaries

There can only be one active custom glossary at a time. However there can be multiple active read-only glossaries. These are glossaries that are used to translate your dictionaries.

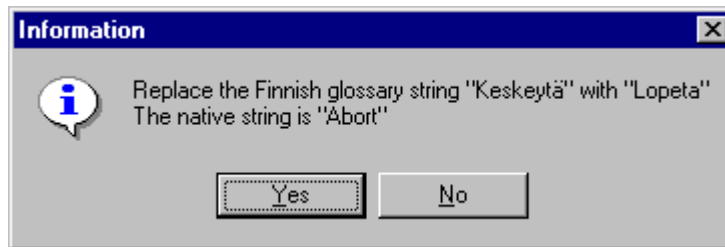
**MORE INFO**

For additional information on using the custom glossary, see the online help topic "Read only Glossary".

Retrieving translations

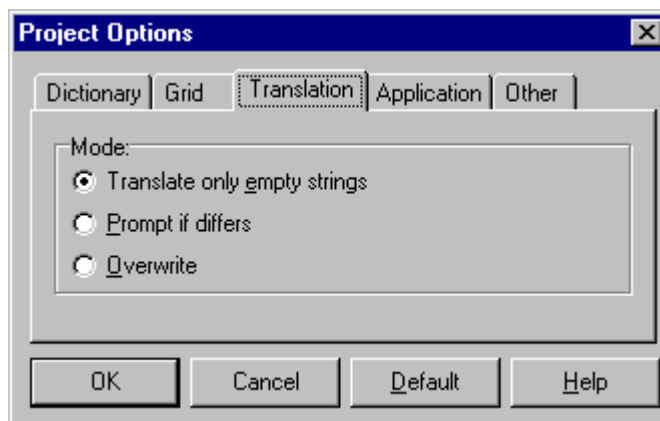
Using the Glossaries translations is easily. In a new dictionary project you just choose from the menu **Project | Translate | Using Glossaries**. Language Manager automatically searches translations for the words in the native language column.

If you already have done translations for some words in the new dictionary, Language Manager by default asks whether to replace it with the translation found in the Custom Glossary or not.



Language Manager asks if the existing translation should be replaced with the Custom Glossary translation.

To modify the settings of Glossary based translations, choose **Project | Options...** from the menu, and choose the Translations sheet in the Project Options dialog.



Choosing the Glossary based translation options.

Opening the Custom Glossary

If you want to edit the Custom Glossary, you can open it by choosing from the menu **File | Glossary**.

The Custom Glossary can be edited like any dictionary. All the functionality of editing a standard dictionary is available, with the following differences:

- No scan button (**Ctrl+U**, Update) is available. Translation entries cannot be added by scanning the program source. No hints are shown on the flag bar, either.
- The dictionary type cannot be chosen from the Project menu.

7

Inputting characters

Language Manager is a powerful text editor that you can use to edit all kinds of scripts. It automatically sets the fonts, character sets, and the input method editor (IME) to match the language you are typing in.

Most European languages use accented or diacritical characters. Typing in those characters using a Standard English keyboard might be quite hard or even impossible. To make inputting of characters as easy as possible, Language Manager offers three ways to input non-ASCII characters: Character composing, Keyboard layout changing, and Localized Windows versions

Character composing

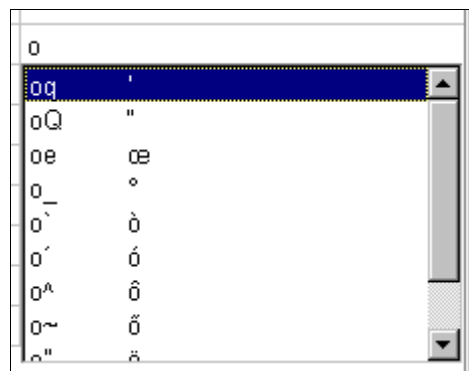
Typing accented characters (e.g. é, ä and ö) can be a problem, at least if you have an English keyboard layout. A non-English keyboard layout has been designed to input accented characters in the language of the keyboard layout. However, it is quite hard or even impossible to type the accented characters of other languages.

To make it easy to type accented and extended characters, Language Manager contains a feature called character composition. You start character composing by typing the compose key (Right-Ctrl). When in the composing mode you first type the non-accented character following the accented character.

For example let's type Ö character:

1. Enter to cell edit mode (double click a cell or press F2)
2. Press the Right-Ctrl key to enter into the composing mode. A 'composing' string appears on the status bar.
3. Press the O key.
4. Press the " key. Language Manager exits from the composing mode and replaces the typed O character with the Ö character.

After typing the first composed character (after #3) you can get a list of available characters by pressing the Down key.



If you press the Down key before you have typed any character (after #2) you are going to get a list of all accented and extended characters in the current language.

To view the character combinations of each accented character, look at the Compose Strings dialog boxes. Choose Tools | Compose strings to open the dialog box.

Character composing lets you input all the accented and extended characters of all Latin character sets (Western, East Europe, Baltic and Turkish). To type non-Latin characters you have to use either keyboard layout changing or a localized Windows version.

Character composing is a handy feature if you only need to input a rather small amount of text. If you have to input accented characters frequently you had better use Keyboard Layout Changing.

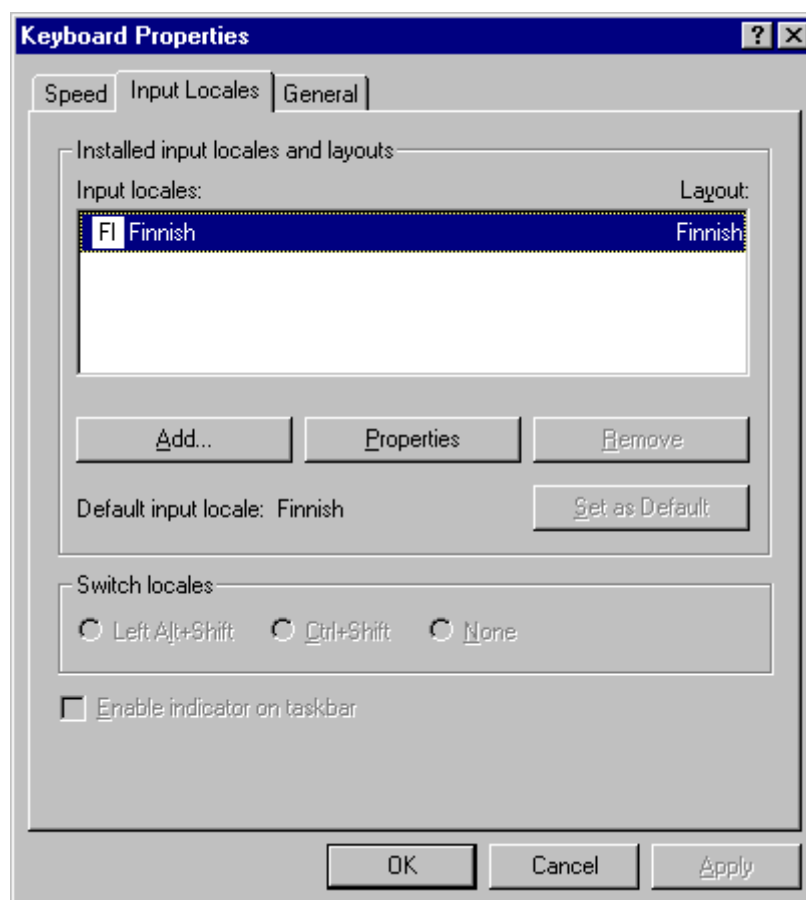
Keyboard layout changing

Windows 95, Windows 98, Windows NT 4.0, and Windows 2000 have the ability to use multiple keyboard layouts.

Each time you select a cell of the column Language Manager checks if the keyboard layout for that language has been installed. If the layout has been installed the program switches that keyboard layout on.

To add a new keyboard layout:

1. Open the Control Panel.
2. Double click the Keyboard icon. The Keyboard Properties dialog box appears.
3. Select the Input Locales sheet.



4. Press the Add button. The Add Input Locale dialog box appears.

5. Select the keyboard layout and press the OK button to close the dialog box.
6. Press the OK button to close the Keyboard Properties dialog box.

You can change the keyboard layout for all single byte character sets. To type bi-directional or multi byte characters you have to use the localized Windows versions.

Localized Windows versions

Windows 2000 can be used to edit all languages. Any western Windows NT 4.0 can edit every European language.

To edit dictionaries containing a bi-directional or a multi byte language(s) you need to run Language Manager on the localized version of Windows, or on Windows 2000.

The following table contains the localized versions that you need:

Language	Windows version
Arabic	Windows 2000, Windows NT 4.0 Arabic, or Windows 95/98 Arabic
Chinese (Simplified)	Windows 2000, Windows NT 4.0 Simplified Chinese Windows NT 4.0 Pan Chinese, or Windows 95/98 Simplified Chinese
Chinese (Traditional)	Windows 2000, Windows NT 4.0 Traditional Chinese, Windows NT 4.0 Pan Chinese, or Windows 95/98 Traditional Chinese
Hebrew	Windows 2000, Windows NT 4.0 Hebrew, or Windows 95/98 Hebrew
Japanese	Windows 2000, Windows NT 4.0 Japanese, or Windows 95/98 Japanese
Korean	Windows 2000, Windows NT 4.0 Korean, or Windows 95/98 Korean
Thai	Windows 2000, Windows NT 4.0 Thai, or Windows 95 Thai
Vietnamese	Windows 2000, or Windows 95 Vietnamese

8

Interchanging dictionaries

There are three Wizards which help in exchanging dictionary data between people working on a project. Furthermore the Wizards help in converting dictionaries from one type into another.

Working with dictionary files apart from the programming project provides many clear advantages. One of the strengths is the possibility to store translations in various formats for later use.

The following chapters describe the Wizards:

- Exchange Wizard
- Import Wizard
- Export Wizard

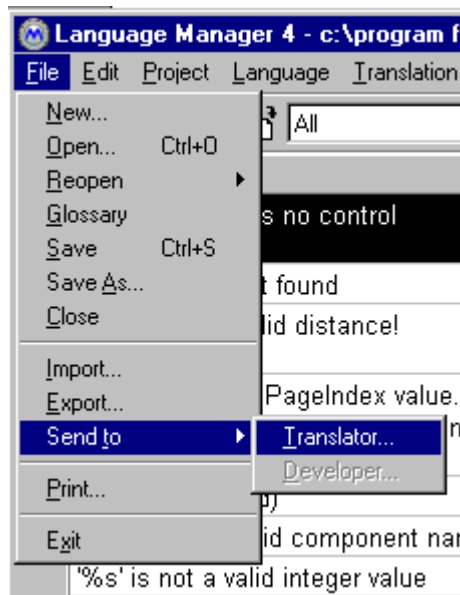
These Wizards make coping with translation data flexible. Using the features described, the actual translation work – done even by several people – can be conducted in an effective and easy manner.

Exchanging a Dictionary

The Exchange Wizard provides an easy way to send and distribute dictionary data. It lets you create a package including both Language Manager (without database support) and the dictionary.

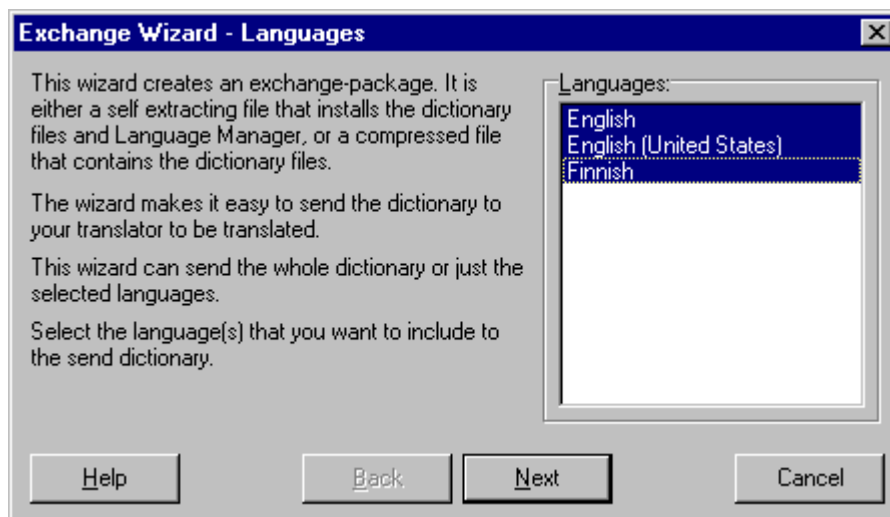
Exchange Wizard works in two ways:

- The developer uses Exchange Wizard to send the (sub)dictionary to the translator.
- The translator uses Exchange Wizard to return the (sub)dictionary back to the developer.



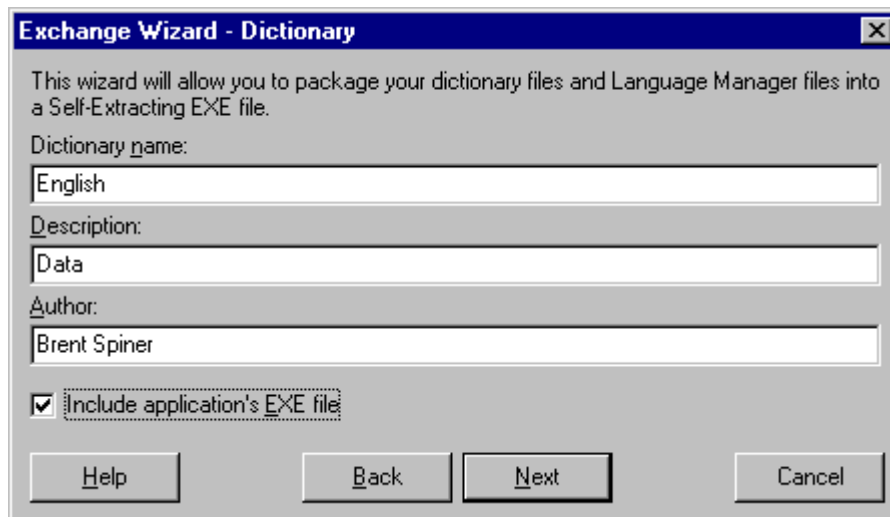
The Exchange Wizard is used in Language Manager for sharing the dictionary data and enabling efficient team work.

The first step is to specify the languages to be sent.



Selecting languages

Next step is to specify the properties for the dictionary to be deployed.



The dialog box is titled "Exchange Wizard - Dictionary". It contains the following text and fields:

This wizard will allow you to package your dictionary files and Language Manager files into a Self-Extracting EXE file.

Dictionary name:

Description:

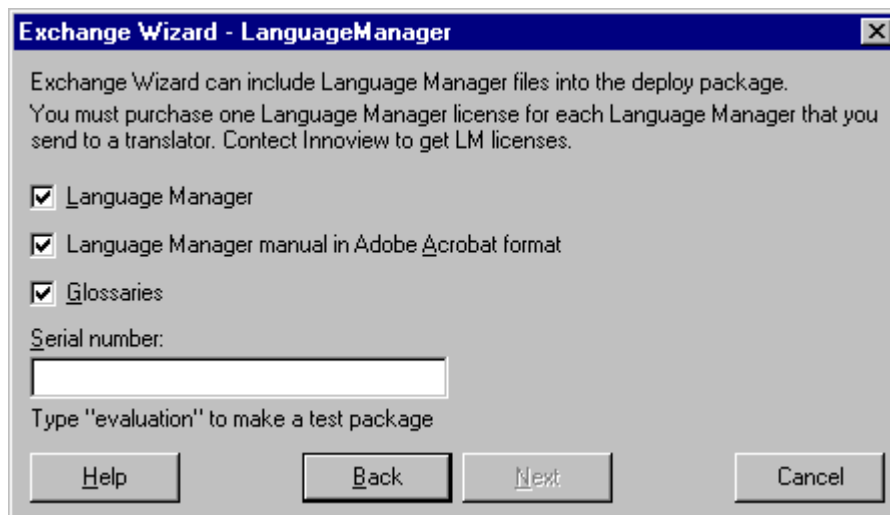
Author:

☒ Include application's EXE file

Buttons: Help, Back, Next, Cancel

Specifying dictionary properties

The next step involves specifying the contents of the package to be deployed. You can choose what to include with the dictionary.



The dialog box is titled "Exchange Wizard - LanguageManager". It contains the following text and fields:

Exchange Wizard can include Language Manager files into the deploy package. You must purchase one Language Manager license for each Language Manager that you send to a translator. Contact Innoview to get LM licenses.

☒ Language Manager

☒ Language Manager manual in Adobe Acrobat format

☒ Glossaries

Serial number:

Type "evaluation" to make a test package

Buttons: Help, Back, Next, Cancel

Entering information on the deploy package



To include Language Manager with the deploy package, you must specify a valid Language Manager serial number. These serial numbers are referred to as Language Manager translator's license, as well. So if 4 translators use Language Manager the 4 translators' licenses are needed.

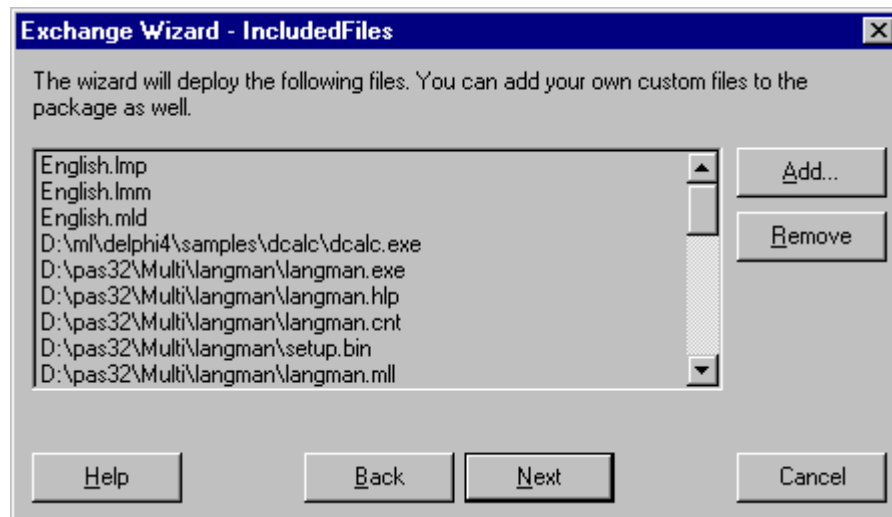


Please ask your MULTILIZER™ vendor or Innoview Data Technologies for more information about Language Manager translator's licenses.

If you don't include Language Manager with the deploy package, no serial number is needed.

Use the "evaluation" serial number to make an evaluation package. This installs Language Manager but it will expire after one day.

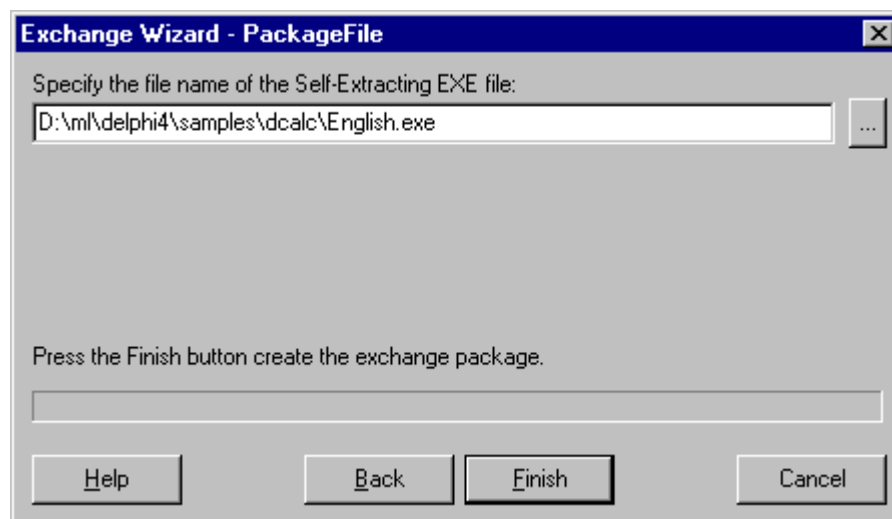
The next screen shows which files will be included in the package.



Files included in the deploy package

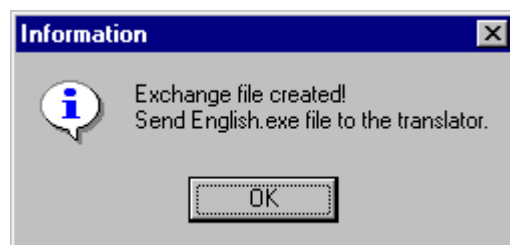
You can add any other files to the package by pressing the Add button.

In the last step, you have to specify the name for the package. Then press **Finish** to create the package.



Specifying the package executable name

Finally the Wizard informs you what file(s) you have to send to the translator. If you use 95/98 there might be several of them.



Package was created successfully

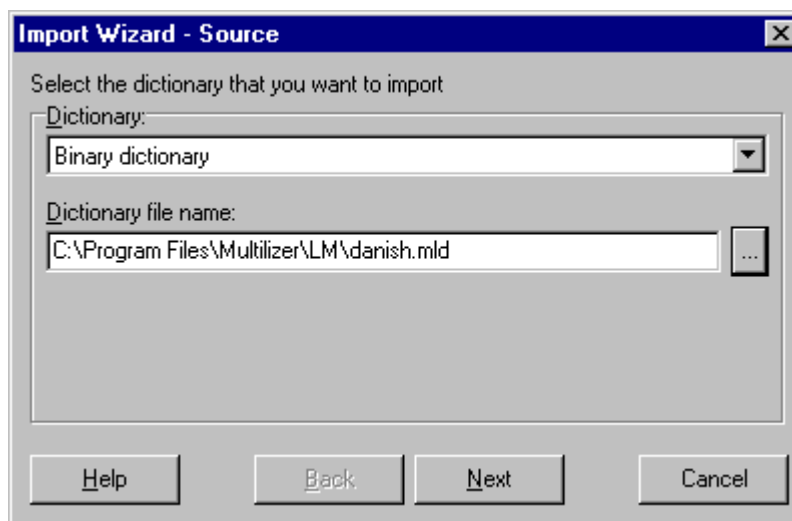
Importing a Dictionary

Using the Import Wizard you are able to import translation data into the current dictionary being edited.

Importing a dictionary is a straightforward process. However, imprecise use of the import command may cause loss of data. Therefore the user must use this command with care.

Starting the Import Wizard

The Import Wizard is started by choosing **File | Import...** from the menu. The following dialog appears.



Select the dictionary type and dictionary media

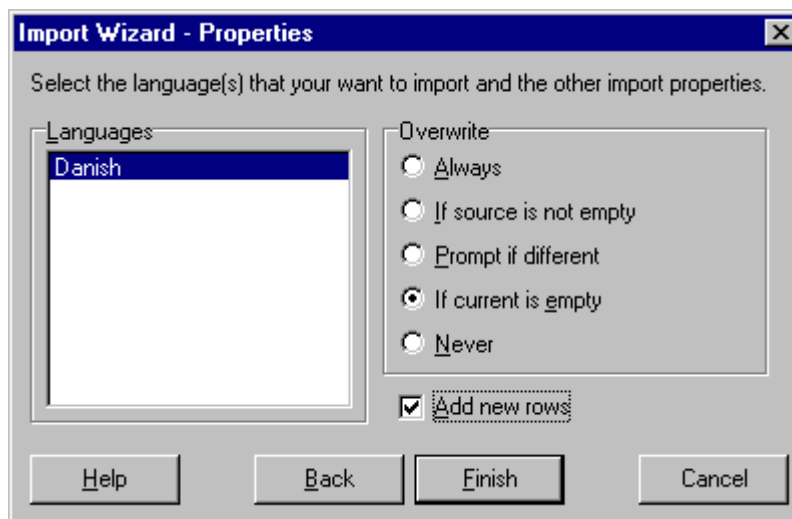


Depending of to the Language Manager version and the configuration of your computer, there may be additional dictionary types available, such as ODBC database tables and BDE database tables.

You first have to select the file(s)/database table(s) containing the translation data. When you are ready, press the **Next** button to proceed.

Specifying import properties

After specifying the dictionary, you have to specify what data and how to import it into the current dictionary. The following dialog shows the options available.



Select the language(s) and import properties

The Languages list box contains the language that will be imported. By default all the languages are selected. Deselect the language(s) that you don't want to import. (In the dialog above, there is only one language to be imported.)

Select the overwrite method. The possible values are:

Value	Description
Always	Always overwrites existing data. If both the current dictionary and the import dictionary contain the same translation (e.g. the same native string). The result dictionary will contain the cells of the import dictionary.
If source is not empty	Overwrites if the source string is not empty. Same as above but if the import cell is empty the current value is used in the result dictionary.
Prompt if different	The Wizard prompts you to accept a translation if it differs from the current one.
If current is empty	Overwrites if the current string is empty. The import cells are copied to the result dictionary only if the current cell is empty. This is the default.
Never	Never overwrites existing data. The current cells are never overwritten. Only adds new languages and new rows (see below).

Check the **Add new rows** check box if you want to add new rows to the dictionary. If this is not checked the result dictionary contains only the rows of the current dictionary even if the import dictionary contained more rows.

Press the **Finish** button to import the dictionary.

Exporting a Dictionary

Exporting a dictionary lets the user either export the whole dictionary or more usually a subset of the languages into another file.

The export command is useful if you plan to deliver supported languages to different persons doing the translations. When the translators have finished their job, they return the dictionary files and you import them back into your project dictionary.



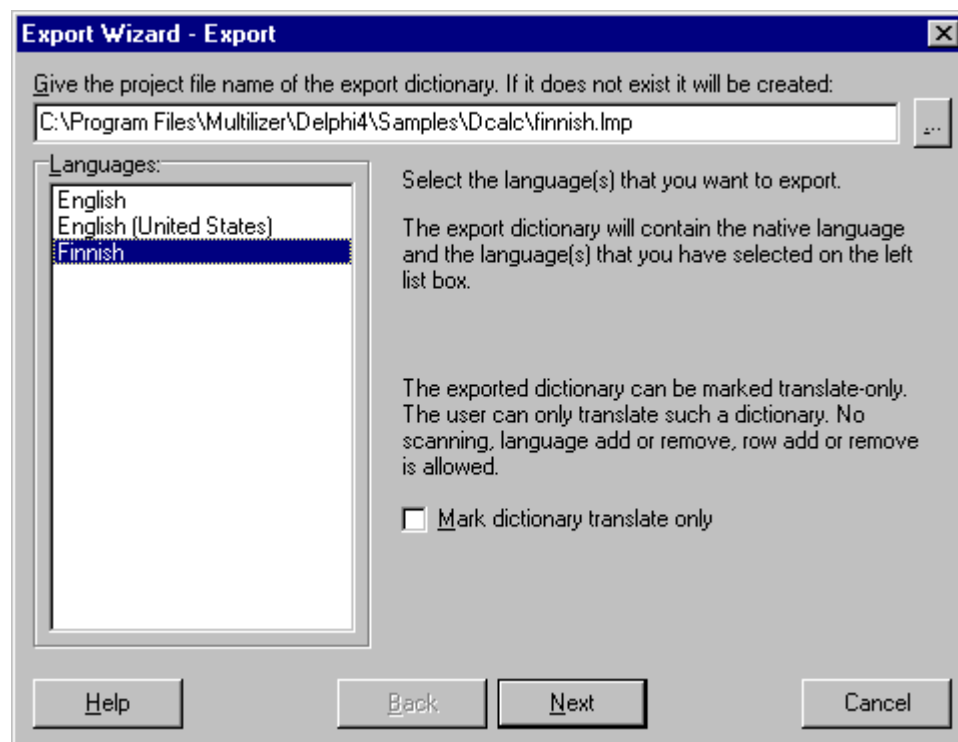
Export Wizard is designed to help split the project dictionary into sub-dictionaries. You can, e.g., specify that no rows can be deleted. Thus, there will be no data corrupted.

For distributing dictionary data to translators, you may find the Exchange Wizard more useful. It packs the dictionary and Language Manager together into a self-extracting package, which can be sent to the translator. This provides an easy way for both the developer and the translator to exchange translation data contained in the dictionary.

Starting the Export Wizard

The Dictionary is exported using the Export Wizard. You start the Wizard by choosing **File | Export...** from the menu.

In the first dialog, you have to enter the project information for the dictionary to be created. (Also, cf. *Entering Dictionary project information*, p. 6). After this the following dialog appears.



The Export sheet

The Export sheet lets you specify a location for the export dictionary and choose the language(s) that it contains.

First you have to type or browse the file name of the project file that will be written. If you type a project file which doesn't exist, a new file will be created.

Then you have to select the language(s) that you want to export.

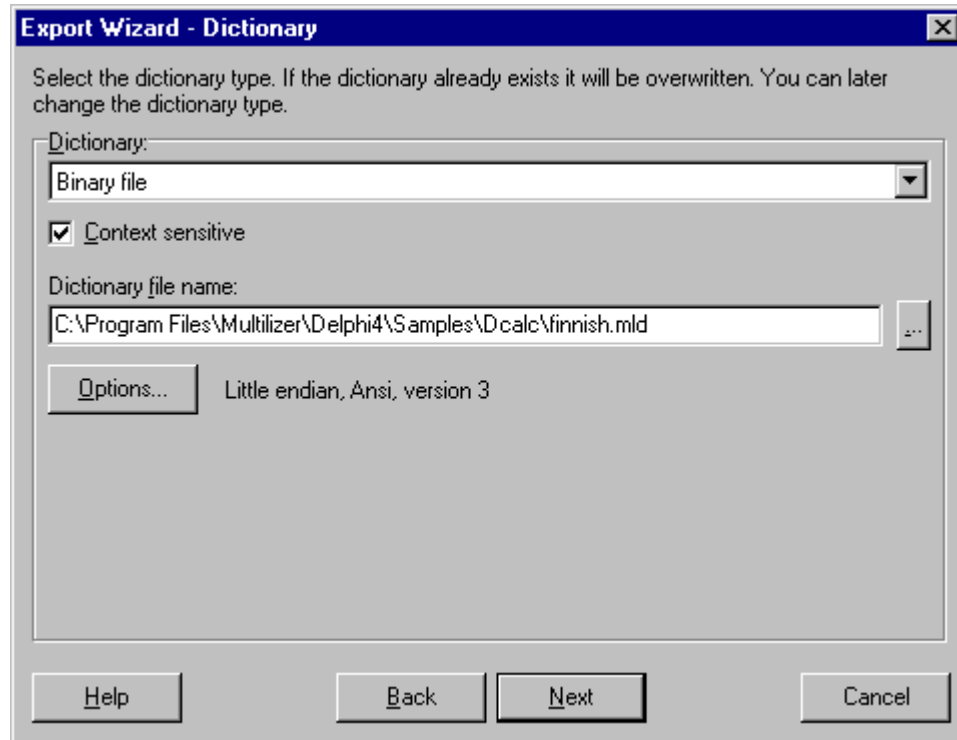
If you are going to ship the exported dictionary to a translator, and you want him to do the translations only, then check the box **Mark dictionary translate only**. This feature lets the exported dictionary be the only one translated. It is not possible to scan, update, add or remove languages, etc.

This makes sure no data is accidentally removed from the dictionary.

When you are ready, proceed to the next step by pressing the **Next** button.

Defining the Dictionary type

The next step is to define the type of dictionary to be exported. In addition to the type defined, you have to enter additional information about the location of the dictionary files.



The Dictionary sheet

The possible choices of dictionary type are:

- Binary file
- Text files
- Resource bundle
- Server dictionary
- Depending on the Language Manager version and the configuration of your computer, there may be additional dictionary types, such as ODBC database tables and BDE database tables.

For further information on dictionary types, see *Specifying Dictionary type*, page 7.

When you are ready, press the **Start** button. The Wizard informs when you are ready to export the translation data into a new dictionary. Pressing the **Start** button again, you begin the export process. Depending on the dictionary type and the size and amount of languages, it may take a while. In most cases exporting is very fast, however.

Appendix A: Language Manager Command line

Language Manager can be run either in interactive mode or in batch mode. Running Language Manager in batch mode involves the use of command line parameters.

Dictionaries cannot be created in command line mode, because the LMP file is needed. Therefore, the project and a new dictionary must be created in interactive mode before command line mode can be applied.

The command line syntax is as follows:

```
langman [/E] [/S] [/T] filename
```

Parameter	Description
/E	Quits the Dictedit.exe program after completing the specified job. Use this switch in conjunction with the /S or /T switch. Saves the project before quitting.
/S	Scan the source code of the project, adding any new string to the dictionary.
/T	Fills up every language column that has a test attached. This overwrites the current contents of the language column with the test output.
filename	Specifies the Language Manager project file (.LMP).

langman is the Language Manager's EXE file.

Appendix B: Language Manager Short cuts

There are a number of short cut keys, which speed up the editing of translations. The following table shows the short cuts.

Shortcut	Corresponding menu item	Action
Ctrl+O	File Open...	Opens a project.
Ctrl+S	File Save	Saves the project.
Ctrl+Z	Edit Undo	Undoes the last cell editing.
Shift+Ctrl+Z	Edit Redo	Redoes the last undo.
Ctrl+E	Edit Cell...	Opens the cell editing window. In this window you can easily edit long strings. When unprintable characters (line feed, tab, carriage return) are entered here, they are not shown in the cell grid.
Ctrl+X	Edit Cut	Cut.
Ctrl+C	Edit Copy	Copy.
Ctrl+V	Edit Paste	Paste.
Ctrl+N	Edit Copy	Paste native. With this command you can paste into the cell the contents of the corresponding native language cell.
Ctrl+F	Edit Find...	Find a certain string in the translations.
F3	Edit Find again	Performs the latest find command again.
Ctrl+Q	Edit Sort	Sorts the contents of the cell grid according to the native column.
Ctrl+T	Edit Next Untranslated Item	Moves the cursor to the next cell not translated.
Ctrl+I	Edit Next Incomplete Item	Moves the cursor to the next cell having an incomplete translation.
Ctrl+U	Project Update	Updates the dictionary.
Ctrl+Ins	Translation Insert Row	Inserts one row to the cell grid.
Ctrl+Del	Translation Delete Row	Deletes the row where the cursor is.
Ctrl+M	Translation Add To Master	Adds the one native string with its translations to the Master dictionary.

Appendix C: Compose character sequences

Language Manager offers the possibility to add accented characters for every Latin code page (1250, 1252, 1254 and 1257).

This is done by composing characters, thus making it possible to enter special characters with a standard US keyboard. The following tables show the default sequences for character composing.

Code page 1250 (Latin 2) covers the languages of Central and Eastern Europe: Czech, Hungarian, Polish, Romanian, Croatian, Slovak, Slovene, Serbian.

Code page 1252 (Latin1) covers most West European languages such as French, Spanish, Catalan, Galician, Basque, Portuguese, Italian, Occitan, Rhaeto-Romanic, Albanian, Afrikaans, Dutch, German, Danish, Swedish, Norwegian, Finnish, Faroese, Icelandic, Irish, Scottish, English. The lack of the ligature Dutch IJ is considered tolerable.

Code page 1254 (Latin 5) replaces the rarely needed Icelandic letters ðýþ in Latin1 with the Turkish ones.

Code page 1257 (Windows Baltic Rim) gives support for languages used in the Baltic states: Estonia, Latvia and Lithuania. Estonian can be fully supported using Windows code page 1252.

Appendix D: Inputting non-ASCII characters

The following table shows the methods available for the most common languages:

Language	Character composing	Keyboard layout changing	Localized Windows versions
Afrikaans	yes	yes	
Albanian	-	NT	Windows 95 Pan European
Arabic	-	-	Windows NT 4.0 Arabic or Windows 95 Arabic
Basque	yes	yes	
Belorussian	-	NT	Windows 95 Russian
Bulgarian	-	NT	Windows 95 Russian
Catalan	yes	yes	
Chinese (Simplified)	-	-	Windows NT 4.0 Simplified Chinese, Windows NT 4.0 Pan Chinese or Windows 95 Simplified Chinese
Chinese (Traditional)	-	-	Windows NT 4.0 Traditional Chinese, Windows NT 4.0 Pan Chinese or Windows 95 Traditional Chinese
Croatian	yes	NT	Windows 95 Pan European
Czech	yes	NT	Windows 95 Czech or Pan European
Danish	yes	yes	
Dutch	yes	yes	
English	yes	yes	
Estonian	yes	NT	Windows 95 Pan European
Faeroese	yes	yes	
Farsi	-	-	Windows NT 4.0 Arabic or Windows 95 Arabic (not fully supported)
Finnish	yes	yes	
French	yes	yes	
German	yes	yes	
Greek	-	NT	Windows 95 Greek
Hebrew	-	-	Windows NT 4.0 Hebrew or Windows 95 Hebrew
Hungarian	yes	NT	Windows 95 Hungarian or Pan European
Icelandic	yes	yes	

Language	Character composing	Keyboard layout changing	Localized Windows versions
Indonesian	yes	yes	
Italian	yes	yes	
Japanese	-	-	Windows NT 4.0 Japanese or Windows 95 Japanese
Korean	-	-	Windows NT 4.0 Korean or Windows 95 Korean
Latvian	yes	NT	Windows 95 Pan European
Lithuanian	yes	NT	Windows 95 Pan European
Norwegian	yes	yes	
Polish	yes	NT	Windows 95 Polish or Pan European
Portuguese	yes	yes	
Rhaeto-Romanic	yes	yes	
Romanian	yes	NT	Windows 95 Pan European
Russian	-	NT	Windows 95 Russian
Serbian	-	NT	Windows 95 Russian
Slovak	yes	NT	Windows 95 Pan European
Slovenian	yes	NT	Windows 95 Pan European
Spanish	yes	yes	
Swedish	yes	yes	
Thai	-	-	Windows NT 4.0 Thai or Windows 95 Thai
Turkish	yes	NT	Windows 95 Pan European
Ukrainian	-	NT	Windows 95 Russian
Vietnamese	-	-	Windows 95 Vietnamese

If the Localized Windows versions column is left empty you can use any Western Windows 95 or Windows NT 4.0 to edit the language strings.

Windows 2000 can be used to input any character set.