

LanguageExplorer

User's Guide

Version 1.3.3
Volker Simonis



Contents

1	Introduction	5
1.1	Overview	6
2	Installation	9
2.1	Installation under Windows	9
2.2	Installation under Linux	10
2.2.1	Changing the hotkey for the input method activation	10
2.3	Installation under Mac OS X	11
3	Handling	13
3.1	Loading books	13
3.1.1	Encrypted books	15
3.2	Navigation	16
3.2.1	Synchronizing the text areas	17
3.2.2	Interchanging the text areas	19
3.2.3	Aligning the text areas	20
3.3	The KWIC-Index	21
3.4	The dictionary	24
3.5	Searching	25
3.6	Regular expressions	27
3.7	Configuration	31
3.7.1	Look and Feel	31
3.7.2	Font selection	32
3.7.3	The user interface language	35

3.8	The online help system	36
3.9	System dependencies	37
3.9.1	Copying and pasting text	37
3.9.2	Input methods	38

A	Constants	43
----------	------------------	-----------

	Bibliography	47
--	---------------------	-----------

Chapter 1

Introduction

LanguageExplorer is a new program for reading texts in electronic form. However, in contrast to other, similar book readers, LanguageExplorer is specialized for displaying several versions of a text in parallel. This may be for example an original text along with its translation or several different translations of a certain text. Therefor LanguageExplorer may be characterized as an electronic synopsis¹ which offers comfortable navigation capabilities. Additionally, given a certain text position in one text, it allows to access the corresponding locations in the parallel versions of the text.

Furthermore LanguageExplorer serves as platform for the integration of arbitrary tools for text reception and analysis. Currently these are dictionaries, powerful search and indexing capabilities and tools for statistical text analysis. New features like bookmarks, user annotations and text apparatuses will be added in the next version..

Another highlight of LanguageExplorer is its ability to cope with texts in virtually any language. Besides the common Western and Eastern European languages he supports languages like Greek and Russian, but also languages written from right to left like Hebrew and languages written with ideographic scripts like Chinese and Japanese.

¹ synopsis: from Greek, “literally, comprehensive view, to be going to see together”. A comparative juxtaposition of similar text version. Traditionally used for the juxtaposition of the four gospels.

LanguageExplorer stores its texts in a modern XML-based file format. Optionally he supports strong encryption of the content he displays, thus effectively preventing illegal duplication of protected materials.

LanguageExplorer has been designed and implemented using cutting edge software technology. It offers a high degree of functionality and user-friendliness. System independence was one of the main goals during development, so today LanguageExplorer is available for the Linux, Windows and Mac OS X operating systems.

Together with LanguageExplorer, which is intended for reading and analysing texts, a second system called LanguageAnalyzer has been developed. LanguageAnalyzer allows the user to create sophisticated, linked editions suitable as input for LanguageExplorer out of simple text based sources. More information about LanguageAnalyzer is available in [LangAn]. While LanguageExplorer is targeted towards the consumer market the audience of LanguageAnalyzer are content providers like publishers and authors.

1.1 Overview

In this chapter the basic functionality of LanguageExplorer will be demonstrated based on some screen-shots. Figure 1.1 shows LanguageExplorer after loading a book. The main part of the program are the text areas which display the different versions of the text: in this example, the original German version of the novel “Die Verwandlung” by Franz Kafka together with an English and a Russian translation.

But LanguageExplorer consists of more than the menu and the text areas. In the region marked with A in figure 1.1, LanguageExplorer has a tool bar. It can be used to execute most of the commands offered by LanguageExplorer in a fast and comfortable way. Additionally, every text area has its own navigation bar (marked with B in figure 1.1) with the aid of which the books may be navigated section- and chapter-wise. While navigating, all the other text areas may be synchronized reciprocally with the actual one. Read more about navigation in chapter 3.2 on page 16.

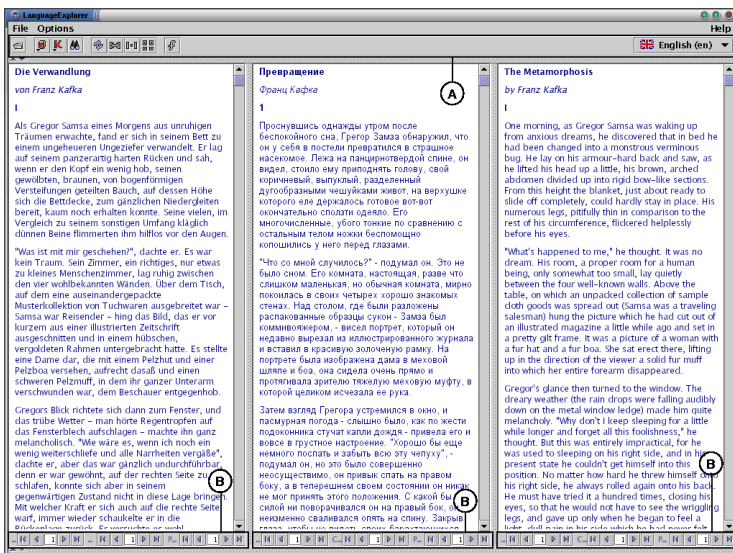


Figure 1.1: LanguageExplorer after loading a book. By clicking the left mouse button on a sentence in the left text area this sentence as well as the corresponding sentences in the other text areas are highlighted.

Figure 1.2 shows LanguageExplorer with opened dictionary (region C) and KWIC-Index² window (region D). The size of both of these windows may be adjusted by the user according to his preferences and they may be opened or closed individually. If a dictionary query is triggered or if a KWIC-Index is generated by the user, the corresponding window will open automatically to the size previously adjusted by the user.

After the basic functionality of LanguageExplorer has been demonstrated in this chapter, the next chapters will present and explain every single feature in more detail.

²KWIC-Index is an abbreviation for “KeyWord In Context”-Index. It denotes an index which not only contains every occurrence of the key word, but also a certain amount of the text before and after the key word. KWIC-Index generation is described in-depth in chapter 3.3 on page 21.

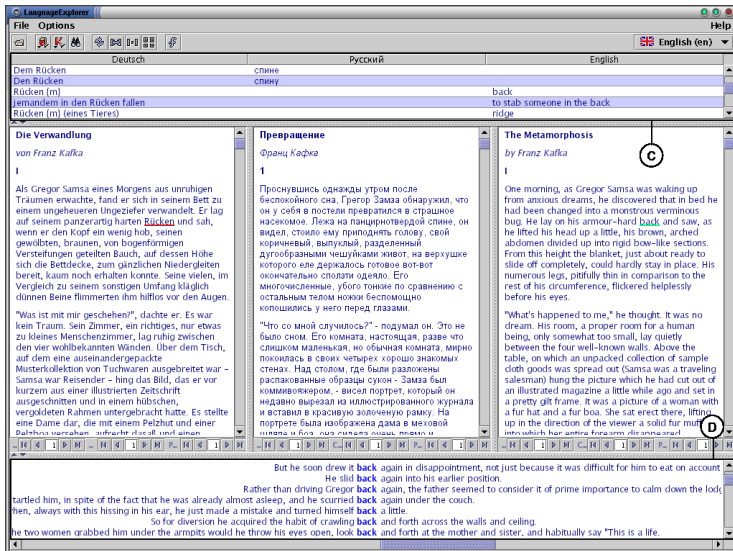


Figure 1.2: LanguageExplorer with opened dictionary and KWIC-Index window. The KWIC-Index visible in the region marked with D in the figure was produced by simultaneously pressing the Shift key and the left mouse button on the word “back”. The dictionary (visible in the part C of the window) was opened by simultaneously pressing the Ctrl key along with the left mouse button on the same word.

Chapter 2

Installation

This chapter covers the installation of LanguageExplorer. Because there are graphical installers available for all the platforms supported by LanguageExplorer the installation is usually a matter of a few minutes. Therefor the next sections will mainly focus on the peculiarities of the different platforms.

2.1 Installation under Windows

Insert the LanguageExplorer CD-ROM into the CD-ROM drive and Choose Run... from the Start-Menu. Type the command `D:\windows\setup.exe` into the appearing text field. Notice that it may be necessary to replace `D:\` with the real name of your CD-ROM drive. Thereafter follow the instructions given by the installation program. By default LanguageExplorer will be installed into the folder `C:\Program Files\LanguageExplorer` however the target folder may be changed by the user. Please be aware that under Windows NT, Windows 2000 or Windows XP Professional you may need Administrator privileges in order to install LanguageExplorer into the default `C:\Program Files` folder.



After successful installation there will be a new LanguageExplorer menu entry in the Programs sub menu of the Start-Menu. Under

this new menu the entry `LanguageExplorer` can be used to start `LanguageExplorer` and the entry `Uninstall` to remove `LanguageExplorer` from the system.

2.2 Installation under Linux



Insert the `LanguageExplorer` CD-ROM into the CD-ROM drive and mount it. The following instructions assume that your CD-ROM drive is available under `/mnt/cdrom`. Start the program `/mnt/cdrom/linux/setup.bin` and follow the instructions given by the installation program. Depending on which target directory you choose for installation you may be required to have root-privileges.

After the installation completed successfully `LanguageExplorer` can be started with the command `/opt/LanguageExplorer/LanguageExplorer` where `/opt/LanguageExplorer` may have to be replaced with the actual installation path chosen during installation. With the `Uninstall` program, which is located in the same directory, `LanguageExplorer` can be removed from the system.

2.2.1 Changing the hotkey for the input method activation

As described in chapter 3.9.2 on page 38 `LanguageExplorer` supports input methods for the input of characters not available on the keyboard. Such an input method may be selected from the input method menu which can be activated by pressing a certain hotkey combination. By default this is the `F4` key. However, this hotkey may be changed by setting the environment variables `INPUTMETHOD_SELECTKEY` and `INPUTMETHOD_SELECTKEY_MODIFIERS`. By appending the line `export INPUTMETHOD_SELECTKEY=VK_F8` to the end of the `.bashrc` configuration file, the hotkey can be changed to `F8`. The file `.bashrc` is located in the users home directory. The environment variable `INPUTMETHOD_SELECTKEY` can be set to the values `VK_F1` to `VK_F12` and `VK_A` to `VK_Z` corresponding to the keys available on the keyboard.

Additionally, the second environment variable `INPUTMETHOD_SELECTKEY_MODIFIERS` may be set to the value of a modifier key, which has to

be pressed together with the key defined before, in order to activate the input method selection menu. The actual values for the three modifier keys can be `SHIFT_MASK`, `CTRL_MASK` or `ALT_MASK`. Setting this variable can also be omitted, in which case pressing the hotkey defined before will be enough to activate the input method selection menu.

2.3 Installation under Mac OS X

Insert the LanguageExplorerCD-ROM into the CD-ROM drive. In the folder `macosx` of CD-ROM click on the archive `setup.sit`. This will expand the installer program and create the application setup in the folder you chose. By executing setup the actual installation process will be started.



By default LanguageExplorer will be installed into the application folder which may require administrator privileges. However an arbitrary installation folder can be selected during the installation process. After completing the installation, LanguageExplorer can be started by clicking the LanguageExplorer icon on the desktop.

Chapter 3

Handling

This chapter will give a brief description of every single function available in LanguageExplorer. Functions are grouped together into sections based on their subject, where every section starts with the description of the most important functions for a given area. Subsections contain the description of special auxiliary functions.

3.1 Loading books

After starting LanguageExplorer, the first thing to do, before any meaningful work will be possible, is opening a book. This can be achieved by choosing Open Book from the File menu or by clicking the Open Book button (see right margin) on the tool bar.



Like every menu entry available in LanguageExplorer, the Open Book menu entry may be reached by using a keyboard shortcut. For the Open Book menu entry this so-called accelerator is the combination of pressing the Control key together with the O key on the keyboard. It will bring up the open book dialog shown in figure 3.1.



The open book dialog is a default file dialog extended by a custom accessory component tailored specially for LanguageExplorer (see region C in figure 3.1). While region A of the dialog shows the actual folder, region B displays all the available files in that folder. If a file

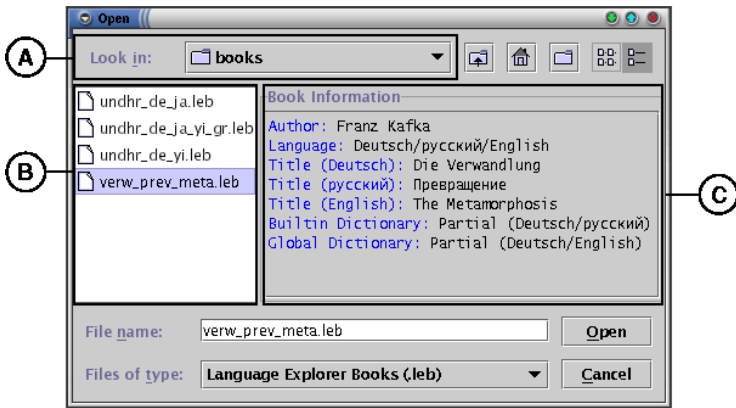


Figure 3.1: The open book dialog. After the file `verw_prev_meta.leb` has been chosen in the region marked with **B**, the accessory component visible in region **C** displays the bibliographic data of the selected book.

is chosen which is in LanguageExplorer book format, the accessory component displays the bibliographic data of the corresponding book. It consists of the author's name, the languages of the different book versions, and the titles of each version in the corresponding language.

One additional information displayed in the accessory is the availability of dictionaries for the selected book. LanguageExplorer supports two kinds of dictionaries: global ones and builtin dictionaries. While global dictionaries are available to all books, builtin dictionaries are packed together with the books into the LanguageExplorer book files. They can be used only by the corresponding book and they usually contain only the words occurring in that book. If both dictionaries are present for a certain book, LanguageExplorer uses a two step algorithm when looking up a word in the dictionary where the builtin dictionary will always be favored against to the global one. More information on the dictionary function may be found in section 3.4 on page 24.

Finally, a book file may be opened by double clicking on the corresponding book file or by pressing the Open button for an already selected book file.

If the text areas contain some strange character glyph or don't display any characters at all after the loading of a new book, this indicates that the actual font is not capable of displaying that text. It may be necessary to select a new font by using the LanguageExplorer font selection dialog which is described in section 3.7.2 on page 32.



3.1.1 Encrypted books

As already mentioned in chapter 1, LanguageExplorer also supports encrypted books. They have the same file suffix like usual, unencrypted book (namely .leb) and they are displayed in the same fashion like usual books in the open file dialog. However when opened for the first time, a so called authentication dialog as shown in figure 3.2 is presented to the user.

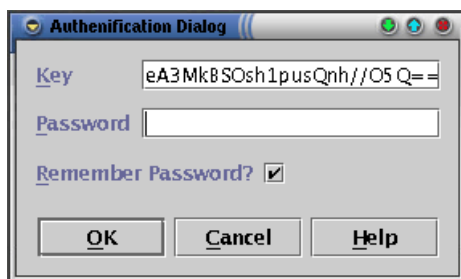


Figure 3.2: The authentication dialog with a key entered by the user.

It prompts the user for a key and a password for the selected book. This key/password combination is usually user and book dependent and was created by the publisher of the book for every user who bought that book. If you didn't receive your personal key and password combination for an encrypted book when buying it, please contact your dealer or the publisher of the book.

If the “Remember Password” check box is selected when entering the password, LanguageExplorer will store an encrypted version of the password in the personal preference file of the actual user in order to avoid the password dialog the next time the same book will be

loaded again. Because the key for every encrypted book is stored by LanguageExplorer automatically, it has to be entered only when loading an encrypted book for the very first time.

3.2 Navigation



After loading a book as described in the previous chapter, LanguageExplorer looks as shown in figure 1.1 on page 7. By dragging the drag bar which is located between the different text areas, the size available to each of them can be customized. This makes sense if one text area contains a more condensed version of a text than the other ones for example. By adjusting their width, the text areas can be usually customized in such a way to hold approximately the same amount of information per window.



Pressing the left mouse button on an arbitrary sentence in one of the text areas will highlight that sentence and all the corresponding sentences in the other text areas as well. It must be noticed that in the other text areas more then one sentence may correspond to the sentence selected first. Under certain circumstances it may also be possible that there is no corresponding sentence in a particular text version in one of the other text windows. Pressing the right mouse button in one of the text areas will remove the highlighting in each of them again.



The cursor keys (see left margin) as well as the PageUp and PageDown keys can be used to navigate the text inside the text areas. While the cursor keys scroll the text line by line the PageUp and PageDown keys (see left margin) may be used to scroll the text page wise, where a page always corresponds to the currently visible text in the corresponding text area. Page wise scrolling is done in such a way that there will be always at least one line of overlap between the page which was displayed last and the new one.



The actions just described can be initiated with the mouse as well. For it, the mouse has to be pressed on the scrollbar (see left margin) located on the right side of every text area. In doing so, clicking the small arrows of the scroll bar corresponds to the line by line scrolling done with the cursor keys while just clicking inside the scrollbar area

is equivalent to the page wise scrolling done with the PageUp and PageDown keys. By dragging the scrollbar with the mouse to a fixed position, it is possible to directly navigate to the text position which corresponds to the relative location of the scroll bar. Independently of the navigation method used, the scrollbar position always signals the relative position of the displayed page in relation to the whole text.



Figure 3.3: A picture of the navigation bar. The text area belonging to this navigation bar just displays the first section in the second chapter in the first part of its book.

As a last possibility the navigation bar (see figure 3.3) located at the bottom of every text area (see region B in figure 1.1) may be used for a structural navigation of the text. By clicking the corresponding arrow buttons with the mouse, the text may be navigated section, chapter or part wise back and forward. It is also possible to jump to a certain of these structures by entering its number into the appropriate text field.

Additionally it is possible to jump to the very first and the very last element of the before mentioned structures (e.g. the first or the last section of a chapter) with the help of the Begin and the End buttons (see right margin). Similarly to the scrollbars, the navigation bars are always synchronized with their corresponding text area. They always show the element which is displayed in the upper left corner of the text area, no matter which means of navigation is used.



3.2.1 Synchronizing the text areas

One of the main features of LanguageExplorer is its ability to show different versions of a text in parallel where always the corresponding part of each version is visible. Usually the synchronization is done automatically. Even if navigating in one of the text areas as described in the previous section, the other text areas are always updated to show the corresponding parts.

However sometimes this synchronization may be not necessary or even hindering. For example when searching in one of the text areas

(see section 3.5 on page 25) it may be helpful to temporarily disable the synchronization. And indeed this is possible in LanguageExplorer. Every text area may be individually synchronized or unsynchronized with the other ones.

Synchronization for two text areas



By clicking the left synchronization button on the tool bar (region A in figure 1.1 on page 7) the left text window will be unsynchronized from the right one. This means that the right window will not follow any navigation in the left window. Notice that the synchronization buttons are so called toggle buttons. Clicking the left button once again will reconnect the left text area to the right one such that all movements done in the first one will be followed by the second one respectively. The state of the button is indicated by the small check mark in the lower right side of the button. If the check mark is present, the corresponding window is connected to its sibling window. If the check mark is absent as shown in the right icon on the left margin, then the navigation in the corresponding text area is independent of the second one.



The hot key `Ctrl-L` or the menu entry `Options→SyncLeft` may be used instead of the synchronization button located on the tool bar to configure the synchronization behavior of the left text area.



The right window may be synchronized with the left one in the same manner like the left window with the right one. The user may choose between the right synchronization button from the tool bar (see left margin), the menu entry `Options→SyncRight` and the hot key `Ctrl-R`.



Synchronization for several text areas



The synchronization buttons on the tool bar automatically switch their appearance in the way shown on the left margin if a book with more than two version of a text is loaded. Because now it is not possible anymore to represent the synchronization status of every single window by its own button, a different approach was taken. The left synchronization button has the duty to synchronize a single window

with all the other windows, while the right button may decouple a window from the other ones.

Clicking on the left synchronization button changes the cursor to the shape shown on the right margin. After the cursor has changed, it is possible to synchronize an arbitrary text area with all the other ones by simply clicking with the mouse into that text area. After clicking, the mouse cursor changes back to its default shape. If the mouse will be clicked outside of a text area, it will also resume to its default shape and no action will be taken at all. Notice that after a book has been loaded all the text areas are synchronized by default.



Clicking on the left synchronization button changes the cursor to the shape shown on the right margin. Subsequent clicking with this mouse cursor into a text area decouples the movements in that window from all the other windows. As with the left synchronization button, clicking into any other part of the application than a text area leads to no action at all. After the first click, the mouse cursor changes back to its initial shape.



For books with several text versions the same hot keys and menu entries for text synchronization are available like for two version books. The menu entry Options→Synchronize Window and the hot key Ctrl-L have the same effect like pressing the left synchronization button whereas the functionality of the right synchronization button is also covered by the menu entry Options→Unsynchronize Window and the hot key Ctrl-R.

3.2.2 Interchanging the text areas

Right after a book has been loaded into LanguageExplorer the different versions of the book are displayed from left to right in the text areas in the same order in which they are stored in the book file. This is also the order in which the dictionaries appear in the dictionary view (region C in figure 1.2 on page 8) of LanguageExplorer. The order of the different text versions may be changed however by the user while the order of the dictionaries will be automatically updated to always reflect the text area order.

Interchanging with two text areas



If a book consists only of two version, interchanging their text areas can be done simply by clicking the swap button on the tool bar (see left margin). Alternatively the menu entry **Options**→**Swap Windows** or the hot key **Ctrl-S** may be used.

Interchanging with several text areas



If a book contains more than two different version of a text, pressing the swap button on the tool bar does not automatically interchanges two text areas but instead changes the mouse cursor to the shape shown on the left margin. Now interchanging two arbitrary text areas is simply a matter of subsequently clicking with the mouse into the two windows. Notice that after successfully clicking into the first window the mouse cursor will slightly change again into the form shown on the left margin.

Clicking with this changed mouse cursor on any other region than a text area will abort the interchanging operation and reset the mouse cursor to its original form. The menu entry **Options**→**Swap Windows** as well as the hot key **Ctrl-S** may be also used to start the interchanging operation for several text areas.

3.2.3 Aligning the text areas

Usually text layout is done in every text area, independently of the other text areas. However LanguageExplorer offers the possibility to align the text in all text areas section wise. This will give all corresponding sections in all text areas the same vertical extent. It may be useful for example to get a quick overview of parallel text versions. Especially for synopses where there are no analogous parts for some structures of a given text in the parallel versions, it may help to identify the gaps faster.



The default setting after starting LanguageExplorer is normal, not aligned text layout. By pressing the align text button on the tool bar, this may be changed by the user at any time. The align text button is a toggle button. Its state is displayed by a small check mark in its lower

right corner. If this check mark is present, the sections of the different text versions are aligned, otherwise they are laid out normally.

The align text action may also be reached from the menu entry Options→Align View and by pressing the hotkey combination Ctrl-A.



3.3 The KWIC-Index

One of the most helpful features provided by LanguageExplorer is its ability to create arbitrary KWIC-Indices on the fly. As explained in chapter 1, KWIC-Index is an abbreviation for “KeyWord In Context”-Index. It denotes an index which not only contains every occurrence of the given key word, but also a certain amount of text before and after that key word. Usually the index is sorted alphabetically based on the suffix of the key word. The advantage of such an index is the ability to see at once the different context in which the key word appears in the text.

With LanguageExplorer the KWIC-Index for a word can be created by holding down the Shift key and pressing the left mouse button on the desired word in the text. Thereafter the KWIC-Index window as shown in part D of figure 1.2 on page 8 will open and display the generated index. For systems which already define the mentioned key combination, an alternative way for generating KWIC-Indices is available. Simultaneously pressing the Alt and the K key on the keyboard will augment the mouse cursor with a small K in its lower right corner (see right margin). Clicking a word with this mouse cursor will now generate a KWIC-Index of the corresponding word as well. After the KWIC-Index has been generated or after the mouse cursor leaves the original text window, the cursor will be restored to its default shape.



The generation of a KWIC-Index automatically opens the KWIC-Index window. However this window may be closed and reopened at any time by using the KWIC-Index button on the tool bar. The content of the KWIC-Index window will be conserved until a new index for another word will be created. Similarly to the Synchronization buttons described in chapter 3.2.1 on page 17 the KWIC-Index but-





ton has a small check mark on its lower right corner which indicates whether the KWIC-Index window is opened or closed. Opening and closing this window may also be performed with the hot key Ctrl+K or by executing the Options→KWIC menu entry.

Another characteristic of the KWIC-Index button compared with the other buttons of the tool bar described until now is the small arrow on the lower left side of the button. It indicates that a context menu is reachable from this button by pressing (not clicking) it for a while. As shown in figure 3.4 the context menu pops down right under the button and allows further customization of the KWIC-Index creation process.

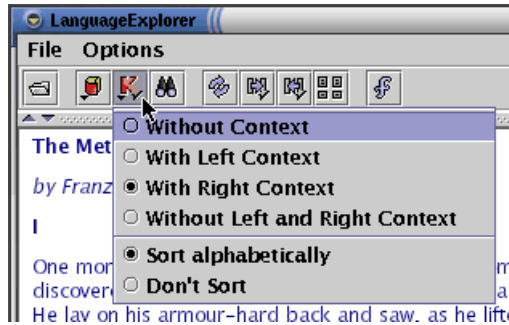


Figure 3.4: Opening the KWIC-Index context menu.

In the upper part of the context menu the user may choose how the KWIC-Index will be created out of the key word selected by the user. The default is to use just the plain word as keyword. It is however possible to create a KWIC index not only for the simple word which has been selected, but for all the words which begin, end or contain the selected word. This can be achieved by selecting the options “With Right Context”, “With Left Context”, and “With Left and Right Context” respectively. For example a KWIC-Index for the word “in” with the option “Without Context” would contain just the word “in”. Together with the option “With Right Context” it could also contain the word “inside”, with the option “With Left Context” it could additionally contain the word “within” and finally if the option

“With Left and Right Context” had been chosen, all the words which contain “**in**”, like for example “**running**” or “**window**” would appear in the index.

In the lower part of the context menu it is possible to choose how the entries of the index will be sorted. Alphabetic sorting means that the entries of the index will be sorted alphabetically with respect to the trailing context of the key word. It must be taken into account that key word suffixes which can occur with the option “With Right Context” are counted as trailing context when sorting. So for example a sorted KWIC-Index with right context for the word “**in**” would contain the sentence “**in**adequate clothing...” sorted before the sentence “**in** both cases...”.

In LanguageExplorer KWIC-Indices can also be created from the Find-Dialog. It offers more sophisticated possibilities like for example ignoring the case of a words or creating KWIC-Indices for arbitrary patterns described by regular expressions. More information on this can be found in section 3.5 on page 25 the Find-Dialog is described.

Once the KWIC-Index has been generated, it contains a single line for every occurrence of the key word. In this line, the keyword will be highlighted and centered so all the key words will be displayed one beneath the other. Notice that highlighting will be done only for the original key word and not for possible suffixes or prefixes of the keyword which may be present because of the various context options.

Navigation in the KWIC-Index window is the same like in the usual text windows (see section 3.2 on page 16) with the only difference that clicking with the left mouse button on a sentence in the KWIC-Index window will highlight that sentence in the text window out of which the KWIC-Index has been created. Additionally, the corresponding sentences in all the other windows will be highlighted as well and all the sentences will be made visible in their windows. All this happens independently of the synchronization settings for the different windows.

In addition to the usual means of navigation, the KWIC-Index window supports the left and right cursor keys to move the whole content of the window to the left or to the right. The size of the





KWIC-Index window is customizable in the same way like the size of the different text areas: by dragging the corresponding drag bar (see left margin) with the mouse to the desired position. Clicking the small arrows on the left side of the drag bar is another possibility of opening and closing the whole window.

3.4 The dictionary

As already mentioned in the introduction, one feature of Language-Explorer is its ability to integrate and use different dictionaries. It is possible to use general dictionaries which will be available to all the books in the corresponding languages or special dictionaries which are integrated into the books and usually contain only the vocabulary used in them. If dictionaries exist for a book at all, and if they are global or builtin may be determined at load time by using the accessory component of the File-Dialog (see figure 3.1 on page 14).



If at least one dictionary is present, it is possible to query it for a certain word by simply clicking with the left mouse button on that word while simultaneously pressing the **Ctrl** key. As with the KWIC-Index generation there is a second way to query the dictionary. Pressing the **Alt-D** key combination will change the cursor by adding a small **D** to its lower right corner as shown in the picture on the left margin. Now querying the dictionary is a simple matter of clicking the desired word with this mouse cursor.



If at least one entry will be found in the dictionary for the selected word, the dictionary window in the upper part of LanguageExplorer will open automatically and display the matching results. By using the dictionary button from the tool bar it is possible to open and close the dictionary window as desired. The buttons functionality, which conforms to that of the the KWIC button described in the last section, may also be reached from the menu entry **Options→Dictionary** or by using the keyboard shortcut **Ctrl-D**.

The size of the dictionary window may be adjusted in the same way like the size of the KWIC-Index window: by moving around the corresponding drag bar. The arrows on the left side of the drag bar can be used as an alternative for opening and closing the window. The

only difference during navigation in the dictionary window compared to the other LanguageExplorer windows is the fact that clicking with the mouse has no effect in this window.

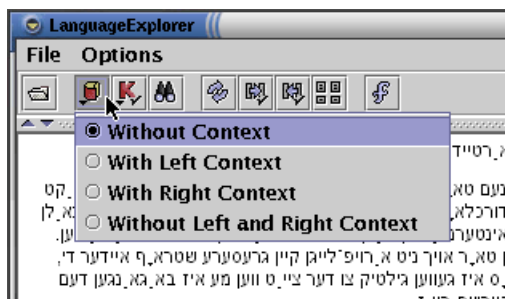


Figure 3.5: Opening the dictionary context menu.

Pressing the dictionary button for a while will open a context menu which allows some customization of the dictionary look up process. As can be seen from figure 3.5 it not only resembles the KWIC-Index context menu, it also has the same options concerning the context of the word to query. The only difference compared with the KWIC-Index generation is the fact that dictionary look up is always case insensitive.

3.5 Searching

The find dialog (see figure 3.6) is currently the most complex dialog supported by LanguageExplorer. It can be used to search the text of the loaded book for arbitrary strings or regular expressions¹. Instead of scanning for individual occurrences of the search item it is also possible to generate a KWIC-Index (see section 3.3 on page 21) which contains all the appearances of that item.

¹Regular expressions are search patterns which may contain control characters with a special meaning during searching. More information about regular expressions can be found in section 3.6 on page 27.

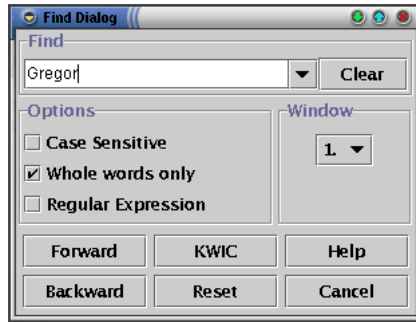


Figure 3.6: The Find-Dialog.



Clear

The Find part of the find dialog contains a text field for entering the desired word or expression to search for and two buttons. The arrow button may be used to open a pull down menu with the history of the last few search terms while the clear button can be used to clear the text field. A search item will be entered into the history list of the pull down menu only after it was searched at least one time.

In the Options part of the dialog it is possible to choose how to search for the search item. The “Case Sensitive” check box selects whether the search will be case sensitive, the “Whole words only” check box selects whether the search will only find the search item as a single word and finally the “Regular Expression” check box selects whether the search item should be interpreted as a regular expression.

Finally the window which will be searched for the search term entered by the user may be specified in the Window part of the dialog by simply selecting it from the corresponding pull down menu.

Forward

Backward

After all the search options have been specified, the search process may be started with the buttons located in the lower part of the dialog. It is possible to search forward in the corresponding text area as well as backward. The search process always starts in the upper left corner of the visible part of the text area for the forward search and in the lower right corner of the visible part of the text window for the backward search. Thereafter searching continues relative to the last occurrence of the search item.

However several peculiarities have to be taken into account. The find dialog is a so called “non-modal dialog” with the consequence that it is possible to navigate in any of the LanguageExplorer windows while the dialog is displayed, create a KWIC-Index or even look up a word in the dictionary. If no sentence has been marked in the corresponding search window before the search is resumed, the search will continue as described above. If however a sentence had been selected in between, forward searching will continue at the beginning and backward searching at the end of the last selected sentence respectively.

The Reset button can be used to reposition the visible part of the actual text area to the position valid before the find dialog was called or before the target window in the find dialog was changed for the last time. The Cancel button quits the find dialog, however without repositioning the current view position.

Finally, the KWIC button can be used to create a KWIC-Index of the search item. Because the search item can be interpreted as a regular expression, the KWIC-Indices generated this way can be much more complex than the ones created in section 3.3. If the KWIC-Index is generated for a regular expression, the whole text string that matches the expression will be taken as key word. And because of the properties of regular expression, these key words may well be different text strings for the same regular expression. Sorting is done based on the suffix which follows the text string that was matched by the regular expression and based on the settings made in the KWIC-Index context menu (see section 3.3 on page 22). Notice that it is possible to generate a KWIC-Index which is case insensitive with respect to the key word by simply unselecting the “Case Sensitive” check box.

A rectangular button with a light gray background and a thin black border, containing the word "Reset" in a small, black, sans-serif font.A rectangular button with a light gray background and a thin black border, containing the word "Cancel" in a small, black, sans-serif font.A rectangular button with a light gray background and a thin black border, containing the word "KWIC" in a small, black, sans-serif font.

3.6 Regular expressions

Regular expressions are search patterns which can contain special control characters. These special characters are called meta characters. They must be quoted with a preceding \ character to treat them as usual characters. There are a lot of different idioms for regular expression which usually differ in the kind of the meta characters and the

extensions they add to the classical regular expressions. Language-Explorer uses a syntax similar to the one known from Perl regular expressions [PerlRegEx] with some extensions for Unicode processing [UniRegEx]. Following, inside the quotes, all the meta characters available in the LanguageExplorer flavor of regular expressions: “()[]{}^\$.|?*+”.

The following table lists the most important meta characters and explains their semantics. Finally, the section will be ended by some examples. More informations about regular expressions can be found for example in J. Friedels book “Regular Expressions” [RegExp].

Pattern	Matches the following text:
<i>Single letters and characters</i>	
x	the character “x”. “x” may be any character except a meta character.
\x	the special character “x” where “x” has to be a meta character (e.g. “\.” for the dot sign “.”).
\uhhhh	the Unicode letter with the hexadecimal value hhhh (e.g. “\u0416” for the Russian letter “Ж”).
<i>Character classes</i>	
[abc]	one of the characters “a”, “b” or “c”. A simple character classes.
[^abc]	any character except “a”, “b” or “c”. A negated character classes.
[a-z]	all the characters between “a” and “z”. A simple character range.
[a-m[v-z]]	all the characters between “a” and “m” or between “v” and “z”. The union of two character classes.
[a-o&&[l-z]]	all the characters between “l” and “o”. The intersection of two character classes.
[a-z&&[^l-o]]	all the characters between “a” and “k” and between “p” and “z”. The subtraction of two character classes.
<i>Predefined character classes</i>	
.	any single character.

...to be continued on the next page ➡

Pattern	Matches the following text:
<code>\p{InBlock}</code>	a character in the Unicode block “ <i>Block</i> ”. “ <i>Block</i> ” can be for example “Greek”, “Cyrillic” or “Arabic” ² .
<code>\P{InBlock}</code>	any character except the ones defined to be in the Unicode block with the name “ <i>Block</i> ”.
<code>\p{IsCat}</code>	any character with the Unicode category “ <i>Cat</i> ”. For example <code>\p{IsLu}</code> for uppercase letters ³ .
<code>\P{IsClass}</code>	any characters except the ones with the Unicode category “ <i>Cat</i> ”.
<i>Logical operators and quantifiers</i>	
<code>XY</code>	the regular expression <code>X</code> followed by the regular expression <code>Y</code> . The simple concatenation.
<code>X Y</code>	the regular expression <code>X</code> or the regular expression <code>Y</code> . The simple alternation. The regular expression <code>he she</code> for example matches “he” as well as “she”.
<code>(X)</code>	the regular expression <code>X</code> . The parenthesis are used to delimit a capturing group (see next operator). They also override normal operator precedence. While the expression <code>r(unn) (ead)er</code> for example will match all the words containing either “runn” oder “eader” the pattern <code>r(unn ead)er</code> will only match the words “runner” and “reader”.
<code>\n</code>	the text corresponding to the <i>n</i> -th capturing group. Every text that matches the part of a regular expression enclosed by parenthesis is called a capturing group. Capturing groups are stored during pattern matching from left to right and numbered from 1 to 9. The expression <code>\1</code> for example matches exactly the same text that was previously matched by the first capturing group.

...to be continued on the next page ➡

²Block may be any Unicode block name with the white space characters removed from the name. Table A.1 in appendix A lists all the valid Unicode block names

³The Unicode character categories are listed in table A.2 in appendix A.

Pattern	Matches the following text:
$X?$	the regular expression X once or not at all. The expression $s(ing)?$ for example would match “s” and “sing” but not “singing”.
X^*	the regular expression X zero or more times. The expression $s(ing)^*$ for example would match “s”, “sing” and “singing”.
X^+	the regular expression X one or more times. The expression $s(ing)^+$ for example would match “sing” and “singing” but not “s”.
$X\{n\}$	the regular expression X exactly n times. The expression $s(ing)\{2\}$ for example would match only “singing” but not “s” or “sing”.
$X\{n,\}$	the regular expression X at least n times. The expression $s(ing)\{1\}$ for example would match “sing” and “singing” but not “s”.
$X\{n,m\}$	the regular expression X at least n times but not more than m times.

Even if regular expressions seem to be quite complicated to understand at first glance, it may be nevertheless useful to learn how to use them. As a motivation, the following paragraphs contain some interesting examples.

The regular expression `[\p{InCyrillic}&&[\p{IsLl}]]` matches all the Cyrillic lower case characters. It is the intersection of the set of the Cyrillic characters with the set of the lower case characters.

The regular expression⁴ `(_\p{IsL}+)(_\p{IsL}+){2,3}\1_` matches every repetition of an arbitrary word which is separated by at least two but no more than three other words (e.g. “..to pay attention to..” or “...he felt as if he...”). In the example the first parenthesized part `(_\p{IsL}+)` matches a space character followed by at least one letter. This corresponds to a word. Notice that because of the fact that the expression is parenthesized, it will be stored as the first capturing group. The second part of the original regular expression `(_\p{IsL}+){2,3}` therefore matches at least two but not more than three single words.

⁴In this example the `_`-character will be used instead of the usual space character in order to increase the readability.

Finally the last part `\1_` matches the first capturing group, that is the first word which has been matched, followed by a last space character.

3.7 Configuration

This chapter will discuss the various configuration properties offered by LanguageExplorer which don't apply to special functions but to the program and its user interface as a whole. For convenience, most of these configuration options will be stored persistently between subsequent execution of the program, so they will have to be adjusted only once.

3.7.1 Look and Feel

LanguageExplorer offers the possibility of changing the Look and Feel of an application at run time. Different Looks and Feels are provided and every Look and Feel may be used with different color themes. The Look and Feel, as well as the current color scheme can be changed by invoking the Look and Feel sub menu of the Options menu as shown in figure 3.7.

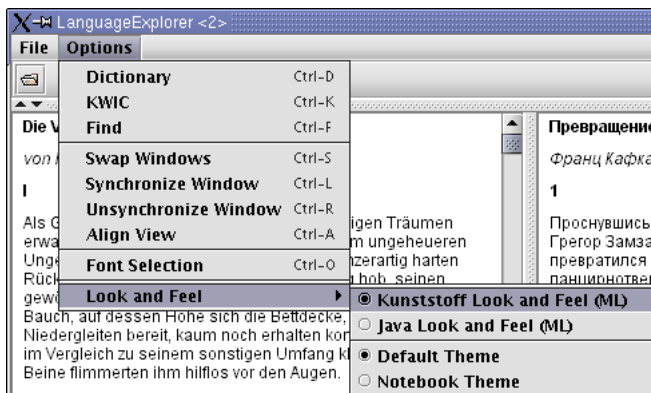


Figure 3.7: Setting the Look and Feel and the color scheme.

The user is advised to try the available Look and Feels and color themes and choose the combination which is most convenient for him. As other settings, the Look and Feel settings are preserved in the personal preferences files between different LanguageExplorer sessions.

3.7.2 Font selection

The font dialog shown in figure 3.8 offers the possibility to select the fonts used to display the LanguageExplorer books on the screen. Font selection is usually made based on several criteria first of which is the personal taste of the user. However usability should be taken into account as well, and fonts which are readable well on the screen should be preferred.

The most important aspect when speaking about font selection is the question which character glyph are supported by the given font and if a font is capable of displaying all the characters available in a book. This is not a trivial task taking into account that LanguageExplorer books may contain arbitrary UNICODE⁵ characters [Unicode]. The UNICODE standard defines about 60.000 characters today. Starting with the well known Latin characters defined in ASCII, it also defines, among others, the letters for the Arabia, Hebrew, Cyrillic, Indic, Thai or Ethiopian scripts but also Chinese, Korean and Japanese ideographs. Unfortunately, there exist few fonts which contain all the characters defined in UNICODE. Therefore LanguageExplorer offers the possibility to select different fonts for every single text window. That way it will be possible to read different versions of a book in parallel even if there is no single font available which contains all the needed characters. Different fonts which contain only the characters needed for a single version will suffice.

In the Window part, in the upper left corner of the font selection dialog the window for which the font selection will be done can be selected. It is possible to select a single window here or to select all the windows in order to set the same font for all windows simultaneously.

⁵UNICODE is a consortium which developed a character encoding system for most of the languages used in the world today. This coding system has been approved as an international standard under the number ISO/IEC-10646.

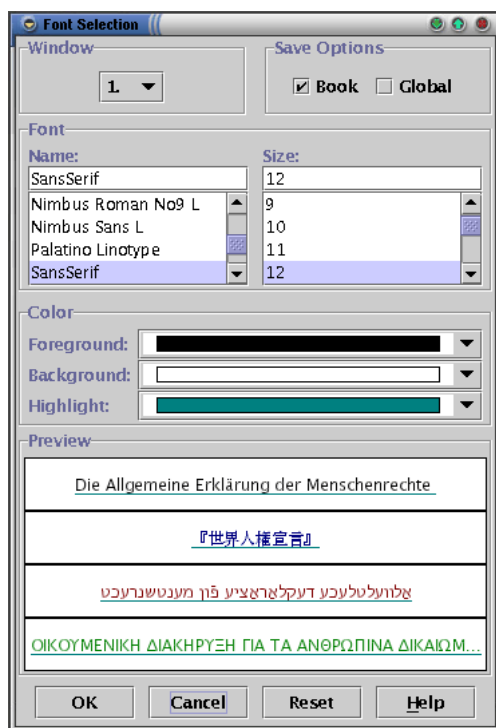


Figure 3.8: The font selection dialog.

Basically it is preferable to use the same font for all windows because this leads to more balanced presentation. However in the above mentioned case where a font doesn't cover all the required characters, different fonts have to be used. The Preview part of the font dialog displays the title of every text version of the current book, each in its own text field. These text lines may be used to check if the desired font supports the characters needed by the corresponding text version.

It is also possible to choose the window for which the font should be changed by simply clicking into the text field with the corresponding title. Clicking into the Preview area outside of any of the text

fields will select all the windows for font change.

Finally, a new font can be selected in the Font part of the font selection dialog. Clicking on one of the displayed font names will select the font and update the Preview panel in order to reflect the font change. Depending on whether a single window or all the windows have been selected for update, only one or all the text fields will change. The same holds true if a new font size will be selected in the Font panel.

There are small editable text fields above the font name and font size selection lists. They can be used to manually enter the desired font name or font size. For the font name, it is sufficient to enter the first unique letters a name in order to select it. While the new input for the font name has to be present already in the name list in order to be acceptable, it is possible to enter size values not offered in the size list. Such new values will be inserted into the list.

The Color part of the font dialog offers pull down menus for the selection of the foreground, background, and underline color respectively. Any changes made in this panel will be reflected immediately in the Preview panel as well.

A small rectangular button with the word "Reset" in a sans-serif font.

The Reset button can be used to undo the changes made so far in the font dialog box. Pressing it will only reset the settings changed in the font dialog since the dialog was opened. If a single window has been selected in the Window part of the dialog, only changes for that particular window will be undone, otherwise, all the font attributes for all the windows will be reseted to their initial values.

It is possible to make the actual changes persistent between different LanguageExplorer executions by selecting one of the options in the “Save Options” part of the font dialog. If neither of the two check boxes is selected, the changes will be effective only for the current LanguageExplorer session. They will be lost when LanguageExplorer will be started the next time. With the Book option, the actual settings will be saved for the current book. If at any later time the book will be reloaded, the current font settings will be immediately applied to the corresponding text windows. Using the Global option when leaving the dialog will save the current settings as the default LanguageExplorer settings which will be loaded every time at program start up and for books for which there exist no font settings

until now.

The save options just mentioned apply only if the dialog is left by pressing the OK button. This will store the font settings in the desired way and update the text windows to reflect the changes as well. All the windows will be updated simultaneously in the way displayed by the preview panel of the font dialog, no difference which window was selected in the dialog when the OK button was pressed.

Leaving the font dialog with the Cancel button discards all the changes done so far and leaves the text areas of LanguageExplorer unchanged.

OK

Cancel

3.7.3 The user interface language

One of the nice features of LanguageExplorer is its ability to switch the language of the user interface elements at run time, without the need to restart the whole program.



Figure 3.9: The LanguageExplorer locale chooser.

Switching the user interface language at run time can be easily done with the locale chooser shown in figure 3.9. The locale chooser is a pull down menu which can be opened by clicking the small arrow on its right side. In the closed state it displays the current language while it offers a list of available languages in the open state. LanguageExplorer is fully localized⁶ for German, English, Russian

⁶Localization is the process of adapting a program to conform to the language, formatting rules, and cultural nuances of a specific region of the world.

and French. If switching to a language not fully supported by LanguageExplorer until now, all the string resources not localized will be displayed by using their English default values.

3.8 The online help system



LanguageExplorer comes with a fully fledged and comfortable online help system with searchable index (see figure 3.10). The whole user's manual is available in electronic form during program execution. It can be accessed at any time through the menu bar (Help→Tutorial) or by pressing the F1 key. Additionally, most of the LanguageExplorer dialogs have an auxiliary Help button which was not mentioned until now. Pressing such a Help button will automatically open the help system and jump to the corresponding place in the manual where the description of the dialog is located.

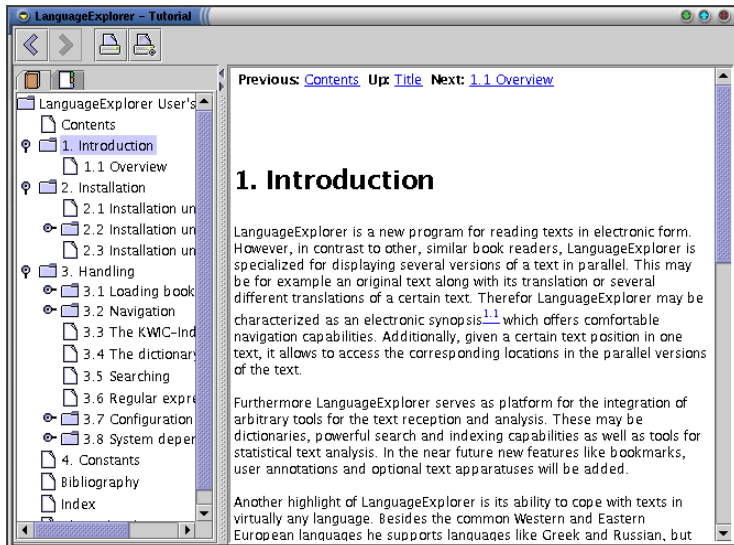


Figure 3.10: The online help system of LanguageExplorer.

3.9 System dependencies

LanguageExplorer has been developed from the very beginning to be platform independent. Due to the significant differences of the target platforms supported by LanguageExplorer it is unavoidable however, that minor differences in the handling may occur. This chapter is devoted to explaining and working around these system dependencies.

3.9.1 Copying and pasting text

While copying and pasting text between other applications and LanguageExplorer works as expected under Windows and Mac OS X, there are some peculiarities to be considered under Linux. While the before mentioned systems have just one clipboard, the Linux's X Windows system⁷ has two of them: a primary clipboard and a secondary clipboard. Selecting text with the left mouse button under the X Windows system automatically copies this selected text into the primary clipboard. Thereafter it can be pasted by pressing the middle mouse button. The problem with this kind of clipboard is that every text selection automatically replaces the old content of the clipboard with the new selection.



That's why X Windows additionally supports the secondary clipboard. Like under the Windows, text is not implicitly inserted into the clipboard by simply selecting text. Instead this has to be done explicitly. However how this is achieved varies between applications. Nowadays most X Windows applications support the `Ctrl-C` and `Ctrl-V` hot keys respectively for copying and pasting text.

LanguageExplorer supports only the secondary clipboard together with the `Ctrl-C` and `Ctrl-V` hot keys under the X Windows system. Therefore it is not enough to simply select text in another application with the left mouse button in order to paste it into LanguageExplorer. Instead the desired text has to be moved into the secondary clipboard. If this is not supported by the source application, the standard X Windows tool `xclipboard` (see figure 3.11) may be used to help.



⁷X Windows is the graphical windowing system of Linux and virtually any Unix based operating system. For more information see <http://www.x.org>.

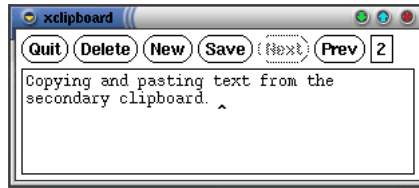


Figure 3.11: The X Windows helper application `xclipboard`.

Using `xclipboard` is quite simple. Executing `xclipboard` on the command line opens the window shown in figure 3.11. Text may now be selected in an arbitrary application with the left mouse button and pasted into the `xclipboard` window with the middle button. Pasting the text into the `xclipboard` program automatically enters this text into the secondary clipboard. Now it can be pasted into `LanguageExplorer` by simply pressing the `Ctrl-V` hot key.

Pasting text from `LanguageExplorer` into a Linux application which does not support the secondary clipboard also works well with the `xclipboard` application. Copying text into the clipboard in `LanguageExplorer` by using the `Ctrl-C` hot key, automatically inserts that text into the `xclipboard` window. Thereafter it can be selected with the left mouse button, thus implicitly inserting it into the primary clipboard, and subsequently pasted into arbitrary other applications by pressing the middle mouse button.

`xclipboard` is also useful because it supports a history of the last few entries of the clipboard. More information about `xclipboard` can be obtained at the command line by typing the command `man xclipboard`.

3.9.2 Input methods

Especially when working with texts in different languages the problem arises that not all letters can be typed with the keyboard attached to the computer because it usually offers only keys for one language. Therefore several different systems have been developed in the last years which allow not only the input of letters not present on the key-

board, but also the input of ideographs for languages like Chinese or Japanese. These systems are commonly called input methods. Input methods range from simple systems which implement a new keyboard mapping for the input of Cyrillic or Greek characters on a Latin keyboard to highly complex programs which allow the comfortable and fast construction of thousands of different ideographs with a usual computer keyboard.

LanguageExplorer not only supports the generic input methods offered by the native operating system, but also custom input methods specific to LanguageExplorer. Because the invocation of these input methods is system dependent, they will be discussed in the following subsections. Basically, every input method belongs to a top level window and all the widgets inside that window. However different top level windows may well have different input methods associated with them. So it would be possible for example for the open book dialog to use the default system input method while the search dialog uses a Cyrillic input method.

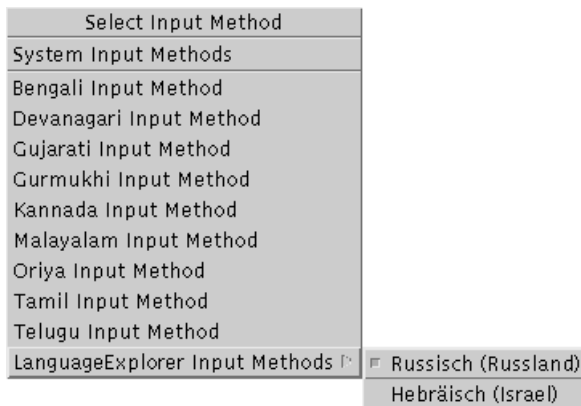


Figure 3.12: The input method selection menu under Linux.

Input method invocation under Linux



To activate a different input method for a top level window under the Linux operating system it is necessary to first click into that window in order to give the window the input focus. Thereafter the F4⁸ function key can be used to bring the input method selection menu on the screen (see figure 3.12).

While the first line denotes the default system input method, the last line of the menu which reads “LanguageExplorer Input methods” opens a sub menu with the input methods specific to LanguageExplorer.

Input method invocation under Windows



The Windows operating system offers a standard way to open a input method for an application. If the application supports input methods, its context menu (as shown in figure 3.13), offers an additional menu entry for the input method selection menu.

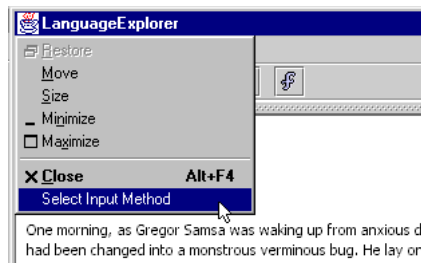


Figure 3.13: The default context menu of LanguageExplorer under windows gives access to the input method selection menu.

Finally the input method selection menu looks exactly the same like the one shown for the Linux operating system shown in figure 3.12.

⁸F4 is just the predefined default key for calling the input method selection menu. This key may be configured as described in section 2.2.1 on page 10.

Input methods under Mac OS X

Under Mac OS X LanguageExplorer currently only supports the system input methods provided by the operating system. They are invoked through the keyboard menu of the application. Notice that the keyboard menu will be visible only if there is more than one input method available. It is possible to install additional system input methods by choosing the “Keyboard Menu” tab from the “International” section of the “System Preferences” window.



Using the LanguageExplorer input methods

After a certain LanguageExplorer input method has been selected for a top level window, a small helper window as shown in figure 3.14 will be displayed in the lower right side of the screen while the top level window has the keyboard focus. This helper window displays the language of the associated input method in its title bar and a picture of the new keyboard bindings. The bindings may change if certain modifier keys (e.g. the Shift key) is pressed on the keyboard, but they will always display the characters currently available.

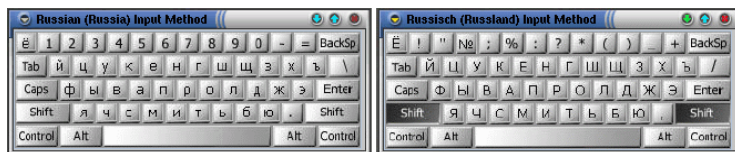


Figure 3.14: The help windows displayed by the LanguageExplorer input method for Russian. On the left side the new default keyboard configuration, on the right side the keyboard layout valid when holding down the Shift key.

As long as an input method is valid for a window, any keyboard action will result in the input of the corresponding characters shown in the helper window instead of the characters visible on the real keyboard. Switching back to the original keyboard layout is just a matter of selecting the system input method for the corresponding top level window.

In `LanguageExplorer` input methods are especially useful in the find dialog if searching a text version written in a language that contains letters which are not directly accessible from the keyboard.

Appendix A

Constants

Predefined character blocks in Unicode 3.0	
BasicLatin	Latin-1Supplement
LatinExtended-A	LatinExtended-B
IPAExtensions	SpacingModifierLetters
CombiningDiacriticalMarks	Greek
Cyrillic	Armenian
Hebrew	Arabic
Syriac	Thaana
Devanagari	Bengali
Gurmukhi	Gujarati
Oriya	Tamil
Telugu	Kannada
Malayalam	Sinhala
Thai	Lao
Tibetan	Myanmar
Georgian	HangulJamo
Ethiopic	Cherokee
UnifiedCanadianAboriginalSyllabics	Ogham
Runic	Khmer
Mongolian	LatinExtendedAdditional
GreekExtended	GeneralPunctuation
SuperscriptsandSubscripts	CurrencySymbols

..to be continued on the next page ➡

Predefined character blocks in Unicode 3.0	
CombiningMarksforSymbols	LetterlikeSymbols
NumberForms	Arrows
MathematicalOperators	MiscellaneousTechnical
ControlPictures	OpticalCharacterRecognition
EnclosedAlphanumerics	BoxDrawing
BlockElements	GeometricShapes
MiscellaneousSymbols	Dingbats
BraillePatterns	CJKRadicalsSupplement
KangxiRadicals	IdeographicDescriptionCharacters
CJKSymbolsandPunctuation	Hiragana
Katakana	Bopomofo
HangulCompatibilityJamo	Kanbun
BopomofoExtended	EnclosedCJKLettersandMonths
CJKCompatibility	CJKUnifiedIdeographsExtensionA
CJKUnifiedIdeographs	YiSyllables
YiRadicals	HangulSyllables
HighSurrogates	HighPrivateUseSurrogates
LowSurrogates	PrivateUse
CJKCompatibilityIdeographs	AlphabeticPresentationForms
ArabicPresentationForms-A	CombiningHalfMarks
CJKCompatibilityForms	SmallFormVariants
ArabicPresentationForms-B	Specials
HalfwidthandFullwidthForms	Specials

Table A.1: LanguageExplorer supports the character block names defined in Unicode 3.0 when constructing certain regular expressions (see section 3.6 on page 27). Notice that these names omit the space characters which are used in the Unicode standard as word separators (e.g. “BasicLatin” is defined as “Basic Latin”).

The character categories defined Unicode 3.0	
Category	Explanation
<i>Characters</i>	
L	Letter.
Lu	Uppercase letter.
Ll	Lowercase letter.
Lt	Title case letter.

..to be continued on the next page ➡

Category	Explanation
Lm	Modifier letter.
Lo	Any other letter.
<i>Numbers</i>	
N	Number.
Nd	Decimal digit.
Nl	Letter number.
No	Any other number.
<i>Symbols</i>	
S	A symbol.
Sm	A mathematical symbol.
Sc	A currency symbol.
Sk	A modifier symbol.
So	Any other symbol.
<i>Punctuation marks</i>	
P	A punctuation mark.
Pc	A connector.
Pd	A dash.
Ps	An opening punctuation mark.
Pe	A closing punctuation mark.
Pi	An initial quote.
Pf	A final quote.
Po	Any other punctuation mark.
<i>Separators</i>	
Z	A separator.
Zs	A space separator.
Zl	A line separator.
Zp	A paragraph separator.
<i>Combining marks</i>	
M	A combining mark.
Mn	A nonspacing mark.
Mc	A spacing combining mark.
Me	An enclosing mark.
<i>Other characters</i>	
C	Any other characters.
Cc	Control character.
Cf	Format character.
Cs	Surrogate character.
Co	Private use character.

..to be continued on the next page ➡

Category	Explanation
Cn	Not assigned character.

Table A.2: The character categories defined Unicode 3.0. In Unicode every character is assigned a general one letter category value. Each category may be subdivided into several, non-overlapping sub-categories which can be identified by a second letter in the category name. For more information consult the Unicode standard [[Unicode](#)].

Bibliography

- [LangAn] Volker Simonis “*LanguageAnalyzer User’s Guide*”
- [PerlRegEx] Perl 5 “*Perl Regular Expressions*”,
<http://www.perldoc.com/perl5.6/pod/perlre.html>
- [RegExp] Jeffrey E. F. Friedl “*Mastering Regular Expressions*”,
O’Reilly and Associates, 1997
- [UniRegEx] The Unicode Consortium “*Unicode Regular Expression Guidelines*”, Unicode Technical Report #18,
<http://www.unicode.org/unicode/reports/tr18>
- [Unicode] The Unicode Consortium
“*The Unicode Standard 3.0*”, <http://www.unicode.org/>

