

What is *Corpus Presenter* ?

Introduction

The present program allows you to view the text files of a corpus and to search through them as you wish. The structure of the corpus is visible from the tree on the left-hand side of the screen. By moving in this tree you can view the various files which are associated with the nodes of the tree (each node contains a descriptive reference to a particular file). You can also load files directly without using a tree and still avail of the sophisticated retrieval options which the program puts at your disposal.

Corpus Presenter is designed to present files of different medium types: text files, images (maps, pictures, etc.), databases (e.g. bibliographies, glossaries) and sound files (e.g. language samples). The program recognises these types automatically and presents them appropriately. Image files are normally in the *Windows* Bitmap (.BMP) or the JPEG Image File (.JPG) formats (though other common formats such GIF [graphic image format], TIF [tagged image format] or WMF [*Windows* metafile] are also accepted). Databases should be in *dBASE* (.DBF) format and audio files in the *Windows* Wave (.WAV) or .MP3 format. For text files, two types are automatically recognised: RTF and HTM files. A HTM file is in the *Hypertext Language* format and can be read and edited by most advanced word processors and by internet software. An RTF data file is in the *Rich Text Format* and can equally be read without difficulty by the majority of commercially available word processors.

In addition, a corpus may contain plain text files (so-called ASCII files). Indeed this is frequently the default case: very often no formatting specific to any word processor is included to ensure that the texts can be read on any computer system. *Corpus Presenter* can of course handle plain texts equally well. Such texts can, if necessary, be edited using the supplied *Corpus Presenter Text Tool* which can process ASCII and RTF files. There is a supplied database processor *Corpus Presenter Quick Database* which can be used where necessary. To process HTM files, use the supplied word processor *Corpus Presenter Word Processor*.

Apart from presentation, the main operation which users will be interested in is searching texts. There are particularly flexible search functions integrated into *Corpus Presenter*. This is discussed in detail in the section *Retrieving information* below.

All the information necessary for the correct presentation of different medium files is contained in a single control file, called a data set file, with the extension .CPD (= *CorpusPresenterData*). This is the file you are requested to select when the program starts or when you choose to open a new data set, see section *Structure of a data set file* towards the end of the present text. This should be consulted if you wish to design your own presentations to run under *Corpus Presenter* (you can, of course, design your own data set files with the program *Corpus Presenter Make Tree*). It is important to stress the flexibility of the program: it can display virtually any set of files in a structured fashion and allow sophisticated retrieval operations to be carried out. A data set file can also be generated in seconds by just marking files on the directory lister level and clicking on the button *Make dataset*.

To get acquainted with *Corpus Presenter* quickly, then F4 on the main level. This opens a window with the most common commands. By double clicking on an icon you can activate an option.

This help file is a normal RTF file and can be printed with the *Corpus Presenter Word Processor* or with any other advanced word processor. A PDF version of the present file – called *Corpus_Presenter_Help.pdf* – is also available.

Bear in mind that there are other help files included in the *Corpus Presenter* suite. The *Corpus Presenter Guide* provides structured information on the options of the entire program suite in a specially written program. The *Frequently Asked Questions* file provides information in question and answer format and is useful when beginning work with *Corpus Presenter*. There is also a quick help function which can be activated within *Corpus Presenter* via the first option in the help menu. An initial help text is available in the latter menu as is a set of troubleshooting tips. There are links to all these help files from the *Corpus Presenter* folder on your *Windows* desktop.

Changes in Version 9.0

Four major changes have been made to the main program *Corpus Presenter* in the latest version:

1) The directory interface has been improved and you now see all the drives on your computer in a tree on the left. Other components on this level have also been improved, e.g. the file list on the right can be made much faster by simplifying the display. Toggle to the quick display mode (hotkey: Shift-Ctrl-F5) and you will find that it only requires a split second to load and display a few thousand files. Two additional features should be highlighted:

(i) You can now generate a DataSet file (the small control file which *Corpus Presenter* uses to display texts in a tree on the left of the main screen) by just selecting files on the directory lister level and clicking on the button *Make dataset* at the bottom of the screen. Once you do this, the selected files are entered into a new dataset file. This is now shown and can be selected for processing by double clicking it and all the retrieval options of *Corpus Presenter* are at your disposal.

(ii) You can now convert files on this level. Click on the button *Convert files* and a window opens offering a range of options and showing the files from the current directory which are eligible for conversion. You could, for instance, convert XML files to text files and use them straight away for search tasks.

(iii) You can generate statistics and wordlists for any file on the directory listing level, just click on the appropriate button at the bottom of the screen. The statistics and word list modules have been improved and now return percentages for word types. Furthermore, you can double click on any return and see the locations in the source text at which it was found.

2) The simple search function, accessed by pressing Ctrl-F, has been enhanced. You can now specify whether the program is to search through all loaded texts, starting at the current one and moving downwards in the list. You can also say whether the search is to be carried out automatically. The results of a search are deposited in the Windows clipboard and can be retrieved via Ctrl-V or Paste in any text processing software.

3) On both the Quick Locate and the Text Retrieval levels you can now store any returns to disk in a format which allows re-loading later (this was restricted to multi-line grid returns on the Text Retrieval level up to this). There is no limit to the sets of returns you can save to disk. If you re-load a set of returns with the corpus of text files from which they were generated (this will normally be the case for most users), then the GoTo

function allows you to jump to the point in the relevant texts where finds were made when the set of returns was originally generated.

4) You can now automatically comb through returns on both the Quick Locate and the Text Retrieval levels and have the results stored to the Windows clipboard. If you have a very large set of returns, say several hundred, it might be useful to search for text within such a set, rather than going through it manually. The function gains additional usefulness in combination with the previous option, so that you can search through sets of returns which were generated in a previous work session.

Some small changes have also been made to display in various parts of the program (it now remembers user-specified sizes of returns tables). There are also one or two changes in file handling (so that the program remembers every alteration the user might make to any file). On the retrieval level you can now look for two strings with zero intervening items, i.e. for strings which are adjacent to each other in a text. Please note that you can now choose to load HTML files as plain text (click on button „Setting“ on the directory listing level). This leads to quick retrieval of text and may be a solution to problems loading complex HTML files in their native format.

Raymond Hickey
March 2006

Changes in Version 9.1

The option of reloading word lists, which have been generated on some previous occasion, has now been added to the *Search* menu. Once a list has been loaded, you can go to any word in it and have all occurrences displayed along with the contexts in which they occur.

The results of searches can now be exported to a HTML file. This option will organise finds as a central column on a line with text to the left and right of it. The option can be found on all levels of the program which involve searching texts.

Raymond Hickey
June 2006

Changes in Version 10.0

The major change in Version 10, compared to Version 9, is the option on all search levels of storing returns as a database which can then be used for chart generation in Microsoft *Excel*. The way this works is as follows: assume that you search for a certain structure across a set of 100 texts and that a percentage of these show the structure, but to varying extents. The database which *Corpus Presenter* stores to disk contains the names of those texts in which the structure was found and the number of times it was found in each text. Such a database can then be loaded into Microsoft *Excel* and via the command *Insert, Chart* (after selecting all the rows and columns of the database with Ctrl-A) you can generate a chart in which the occurrences of the structure you searched for are shown in chart form.

Corpus Presenter can furthermore gather up to eight sets of returns for a group of

texts and transfer these to a database (for later chart generation). The advantage of this is you can see whether the occurrences of more than one structure run parallel across a set of texts, either ascending or descending. You can also check whether two or more structures run in opposite directions, e.g. whether the increase of one structure correlates with the decrease of another.

Some other options have been added, especially those allowing users to store any set of returns on disk and re-load these for editing in a later work session.

There is also a new utility *CP Make Tree* which makes the design and editing of tree structures for *Corpus Presenter* even easier.

Raymond Hickey
January 2007

Changes in Version 11.0

Version 11 of *Corpus Presenter* provides a number of additional features which further increase the flexibility of the program. For instance, there is now a function *Search via word list extract* in the *Search* menu. Via this option you can create a customised extract of forms from a fuller list which you generate yourself – probably via the *Make word list* option. You can use this extract for a search across those texts from which the fuller list was generated. This guarantees that the forms are found and returned.

Another new feature concerns the exporting of returns to a Microsoft Excel table for chart generation. This feature now includes the option of generating not just raw returns from a search but also percentages reflecting the relative frequencies of strings in texts. Along with this there is an option of producing the frequency of finds per thousand words.

The interface for the *Quick locate* function has been simplified considerably so that only the essential options for a search are now visible with the extra options available in an additional window.

In addition, a flexible image viewer has been integrated into *Corpus Presenter*. Furthermore, some smaller changes have been made to existing functions and options.

The supplied word processor – *CP WordPro* – has been greatly expanded and can now load and save Word 2003 and 2007 files directly. Other programs, notably *CP File*, *CP Text Tool* and *CP Find Text* have also been expanded considerably.

At the same time the number of separate programs has been reduced vis à vis earlier versions to facilitate quick orientation within the software suite.

Raymond Hickey
February 2009

Changes in Version 12.0

Version 12 of *Corpus Presenter* constitutes a major upgrade of the program. It has been expanded in a number of ways, for instance with respect to the exporting of data and the interface to well-known commercial products. First and foremost is the export of data from both basic and advanced search levels to Microsoft Excel. On either level you can choose to export the contents of a returns grid to an Excel table. This is done

automatically after you specify and confirm the output table file which is to be written during the export process. As was the case with the Excel export function in Version 11, it is possible to generate charts from the data which is exported from *Corpus Presenter*. To export the contents of any returns grid, click on the button Export to Excel on the *Basic Search* level or choose the option 'Export returns to Excel' in the 'Returns' menu on the *Advanced Search* level.

The *Word List* module has been completely restructured, both expanded in its functions and simplified in its interface. Furthermore, the statistics module has been integrated into the options of the *Word List* module. This is now menu-driven with a tool bar of icons for the main functions. The export options for the word lists have been expanded, allowing you to now export the contents of a returns grid to a Microsoft Excel table.

One of the main expansions of the *Word List* module is to interface with the 'Keyness' module. With the latter you can test for differences between word lists generated from any two corpora, e.g. that of a group of authors and that of a single author. By these means you could, for example, ascertain whether there are stylistic differences between a particular author and others of his era. *Corpus Presenter* produces statistics which show the differences between any two corpora and the program can export these to Microsoft Excel where these differences can be displayed as a chart.

The ability to examine texts for lexical clusters, previously only contained in *Corpus Presenter Text Tool*, has now been expanded and integrated into the main program, *Corpus Presenter*.

The functions pertaining to file listing and loading have been drastically improved so that you can now load a large number of files – say 5,000 – in a second or two (the exact time required depends on the speed of your computer).

The context-sensitive help function has been greatly expanded, offering immediate, visually effective help, for all the major modules of the main program, *Corpus Presenter*.

Many of the utility programs which accompany *Corpus Presenter* have been further improved, e.g. the *Find Text* and the *Make Tree* utilities. With the latter you can interactively design a CPD (dataset control file) which *Corpus Presenter* uses to display the files of a corpus in tree form on the desktop.

Version 12 of *Corpus Presenter* has been successfully tested on a number of computers running under the *Windows 7* operating system.

Raymond Hickey
January 2010

PS: To make the labelling in *Corpus Presenter* easier to grasp the *Quick Locate* option has been renamed *Basic Search* and the *Text Retrieval* option is now termed the *Advanced Search* option.

Update information on *Corpus Presenter* can also be found in the internet at the following addresses: www.uni-due.de/CP (the dedicated website for *Corpus Presenter*).

Command summary

1 File

1.1 *Open a data set...* This option displays the files of a directory (the default directory of *Corpus Presenter* or that which you were in when last working with the program). You can choose a control file to start working with a new data set or press ESCAPE to return to the one you were just processing.

Shortcut: **Ctrl-O, F5**



A *data set file* is a small text file (without any formatting) which contains all the information necessary to display a corpus correctly in tree form from within *Corpus Presenter*. A *data set* is the collection of files (texts and possibly images and sound files) which is displayed in tree form on the left of the screen.

1.2 *Create/edit a new data set (file for tree)* It is possible to design your own corpus from within *Corpus Presenter*. This is done via the current option which loads the supplied program *Corpus Presenter Make Tree* (see help text for this program). Once you have designed a data set file for a corpus and saved this to disk, you can load the corpus using this file.

Bear in mind that you can generate a data set file with the program *Corpus Presenter Make Tree* which will make a data set from any group of selected files in a directory listing.

Shortcut: **Ctrl-N**

1.3 *Load text file(s) directly* If you do not have a data set file for a corpus or if you simply wish to process one or more text files directly, then you can do so with the current option. You choose a file from a directory listing and this is then displayed on the right of the screen. If you have selected more than one file (by pressing the Ctrl-key and marking items in the file list) the first of the group is displayed. There are as many nodes in the tree as files which you selected. In each case the node label is the full path of the particular file. Note that you can load HTM, RTF and plain ASCII files directly from within *Corpus Presenter*.

Shortcut: **Ctrl-L**

1.4 *Load associated text* It is possible to specify the name of a text which you can load and have displayed in a window when you press F6 or select the current item in the *File* menu. This option is useful if there is some text file which you would like to have access to from within *Corpus Presenter* at the touch of a button. You can choose a new file via the option *Get file from disk* in the menu bar of the text box which opens. *Corpus Presenter* will remember this file and load it again when you activate this command.

Shortcut: **F6**

1.5 *Information on current set* Displays a windowful of information on the data set you are currently processing.

1.6 *Nodes and associated files in set* An important part of the work with *Corpus Presenter* is determining what files are to be linked with what nodes in the tree which is the visual representation of the current data set. The present option allows you to check online what files are associated with what nodes. To alter the settings, edit the control file directly and store this to disk. Now re-load the control file by choosing a new data set. The changes are reflected in the links between nodes and files at the points you altered.

Shortcut: **F3**

1.7 *Display node information file* This option is will attempt to open a text file associated with the file linked to the current tree node. To show how this works, consider the following case: you have a node in a corpus tree called *My drama* and linked to that node (via the information in the data set file which controls the display of the tree) is a file called *my_drama.rtf*. Now when you activate the current command, *Corpus Presenter* looks to see if a file called *my_drama_info.rtf* exists in the same directory and if this is the case it will load it and display it in a text window. The upshot of this is that for each of the files associated with nodes in a corpus tree you can have an additional file with information on the node tree. This additional file must have the same name as the node file but end in *_info.rtf*.

1.8 *Directory lister* With this option you simply activate the directory lister. This is a level on which you can view, copy, move, duplicate and delete files. A loaded corpus is not affected and the options *No Load* and *Retrieve* are deliberately disabled.

Shortcut: **F2**



A *directory lister* is a module contained in many programs of the *Corpus Presenter* suite. It not only allows you to select a file for processing, but also to view, copy, rename or delete any files on any drives of your computer.

1.9 *Copy tree to text buffer* Puts a copy of the current tree into an internal buffer which can be retrieved in any text editor, such as *Corpus Presenter Text Editor* via CTRL-V.

1.10 *Exit program* Unloads *Corpus Presenter* after user confirmation.

Shortcut: **Alt-F4, Alt-Q**

2 Edit

2.1 *Undo* Allows you to undo the last deletion.

Shortcut: **Ctrl-Z**

2.2 *Redo* Permits you to cycle through the last formatting moves.

2.3 *Select all* Highlights all of the current text.

Shortcut: **Ctrl-A**

2.4 *Copy* Deposits the currently selected text in the *Windows* buffer.

Shortcut: **Ctrl-C**

2.5 *Paste* Empties the contents of the *Windows* buffer at the current position in the text.

Shortcut: **Ctrl-V**

2.6 *Cumulative clipboard*

You will have noticed that *Windows* only allows you to deposit one item of data, such as a piece of text, in the general clipboard at any one time. The present command allows you to keep depositing stretches of selected text in an internal clipboard and so collect text cumulatively. When the cumulative clipboard window is open you can retrieve text from the *Windows* clipboard or you can choose to copy part or the whole of the cumulative clipboard text to either the text editor or the storage window assuming that you have called it from either of these modules.

Shortcut: **Ctrl-Y**

2.7 *Cut* Moves the currently selected text into the *Windows* buffer.

Shortcut: **Ctrl-X**

2.8 *Delete* Deletes the selected text.

2.9 *Show text file information* Here a window with information on the text associated with the current node is displayed.

Shortcut: **Shift-F9**

2.10 *Hyperlinks for current text* Each text of a corpus can have a set of hyperlinks associated with it. The way this works is as follows. For each text there can be a file with the same name but the extension ‘.hpl’. *Corpus Presenter* will read the contents of this file then you activate the hyperlinks option (via the right mouse button popup, the command button bar or the relevant option in the Internet menu on the top of the screen).

A small window appear with a grid in it. The grid is populated by those references contained in the hpl-file associated with a text. There are two columns in the grid, one is a text describing the source of a hyperlink and the second is the actual reference to the hyperlink. Typically, a hyperlink file will contain an internet address – a URL – which is activated when the user clicks on the row containing it. A grid row may also contain a reference to an image file which is associated with a corpus or indeed a further text file.

A hyperlink file is structured as follows: two lines are needed for each item, the first line is the description and the second the actual reference, like the following example:

Corpus Presenter website
<http://www.uni-due.de/CP>
 A picture of J. M. Synge

John_Millington_Synge.jpg
 Metatext for current corpus text
 synge_metatext.rtf

Shortcut: **Shift-Ctrl-F12**

3 Search

3.1 *Basic search* Allows you to search for a string in one or more texts which are contained in a data set. Obviously, only searches through text files are permissible. This option initiates a dialogue in which you specify what string you wish to look for. This window also allows you to specify the range of a search, to gather finds in a list and store these to disk.

In addition to a straight search, this option allows you to store the finds in a list which can then be consulted after a search (it is retained in memory until a fresh search is made).

You can enter a search string directly or use an input list just as with word lists and complex retrieval tasks. Returns can be collected if required and can be then transferred to the line grid on the retrieval level if necessary. The list of returns allows you to select all items or only a subset. With the latter you decide what items to select by either pressing the Shift key and then either the Up or Down Arrow key (to mark contiguous items) or by pressing the Ctrl key and then either the Up or Down Arrow key (to mark non-contiguous items).

Pressing the Escape key will cancel the location procedure.

Using wild cards during a search

The wildcards * and ?, common in other software, can be used here to increase the flexibility of searches. The question mark stands for a single unspecified character while the asterisk stands for several of these. A few points must be borne in mind when using wildcards to ensure that results are accurate.

- 1) Do not begin a search string with a wildcard, i.e. '*r' is illegal, but 'h*r' or 'h*' is fine (do not enter the inverted commas!). If you are interested in, say, word endings, trying a search from the retrieval level (press Ctrl-L on the main level).
- 2) A space, tab stop, page break, carriage return or line feed cannot be an unspecified character, and hence cannot be captured by an asterisk. The entry 'h*r' will return 'harder' (if present in the text(s) scanned) but not 'had their' as there is a space between 'h' and 'r'.

For more demanding retrieval tasks, choose the option *Advanced search* in the *Search* menu or just press Ctrl-L, see *Retrieving information* below for more information.

Bear in mind that the *Basic search* and the *Advanced search* options can be used to do counts for strings in texts and return percentages. This might well be useful if you wish to know how the distribution of a certain form or structure varies across different texts, be they different in type or from different periods. Choose the option *Only count finds* to have this option at your disposal.

By clicking on the third button on the left (a square with a horizontal line through it) you can toggle between the split screen and full screen mode. When the former is active then moving in the list of returns will automatically show in the bottom half of the screen the position in the text where the find was made. You can copy the find and its immediate context by pressing F4. This way you can collect various finds in a text window and afterwards copy these to your word processor, for example.

The split screen mode is also available on the *Advanced search* level.

Shortcut: **Ctrl-B**

3.2 *Advanced Search* Because retrieval operations are central to *Corpus Presenter*, there is a special, in-depth discussion devoted to these, see *Retrieving information* at the end of this section.

Shortcut: **Ctrl-D**

3.3 *Find string in files* This is the simplest type of search which allows you to look for any string in the currently loaded text or in any files in the tree on left. The option is quick and easy but quite powerful, for instance, if you choose an automatic search through all files you can comb through your whole corpus quickly with the results deposited in the Windows clipboard from where they can be retrieved via the Paste option of word processing software, including the internal text editor of *Corpus Presenter*.

Shortcut: **Ctrl-F**

3.4 *Find string in tree* If there are many files in your corpus, you may wish to search for a string contained in one of the node labels of the tree on the left of the screen. The present function will do this for you. Each time you search, the program starts at the next node following the present one so that you can proceed through the whole tree if you wish.

Shortcut: **Ctrl-T**

3.5 *Make a word list* You can generate lists of words from the text files of a corpus. At a maximum, you can create a word list of all words in all text files of a corpus. This would take some time for a large corpus and is unlikely to be the aim of most users, but can be done on occasions of course. Instead users are probably interested in creating a list of selected words in a corpus. For this reason, one of the first options in the input window which opens on selecting this command is *Input word list*. Here you specify a plain ASCII file which consists of a list of words, one on each line. Such a word list can be easily created with *Corpus Presenter Text Tool*. The next item to remember, and which is concerned with restricting the words used for a list, is a stop word list. Essentially, this is a list of words (again in the form of a plain ASCII file) which are to be excluded from word list generation. For instance, if you choose to make a word list of an entire text, then it is unlikely that you want to have statistics on the occurrence of such common words as *a, the, on, at*, etc. These and similar words can be excluded by putting them into a stop word list and then specifying it on this level of the program.

To generate a word list *Corpus Presenter* examines a file or files, extracts all

words and places each of these on a separate row in the grid which you see in the word list window. Each word is only entered once into the grid. The number of times a word occurs in a file is recorded in the frequency column.

When saving the results in the grid to disk you can choose to have these deposited in a plain text file or in a database (for further processing with one of the supplied database editors). You can also just select rows in the grid and then store only the contents of the selected rows (an extract of the entire grid so to speak). To select contiguous rows, hold the Shift-key depressed and mark the rows by moving with the up or down arrow key. To select non-contiguous rows, hold the Ctrl-key depressed and move from row to row with either the up or down arrow key. To select a row, press the SpaceBar (without releasing the Ctrl-key) or click on the row with the mouse.

Generating a word list is a somewhat slow process as the entire text must be combed through for each word. But it is something you can initiate and leave the computer to work away while you do something else. Note the option of generating a word list for the checked files of the current corpus tree. This option can be used to ensure that all the files you want to encompass are included in the operation.

Examples of a word list, based on input forms with the legal wild cards * and ?

Note that the question mark stands for a single unspecified character and the asterisk for several unspecified characters. The results given here can be repeated by selecting the text `RIDERS.CIE` (Synge's *Riders to the Sea*) which is contained in the test corpus supplied with *Corpus Presenter*.

Input form	Word(s) returned from text <code>RIDERS.CIE</code>
<code>b?g</code>	<i>big</i>
<code>gr??e</code>	<i>grace, grave</i>
<code>g??e</code>	<i>Give, gone</i>
<code>he*d</code>	<i>head, heard, he'd</i>
<code>ho*</code>	<i>holding, holds, hole, Holy, hook, hooker's, horses, hour, house, house-six, How</i>

When you are deciding how returns are to be displayed you can choose between a plain list (which just includes the word and the frequency) or a grid. The latter is much more flexible and you can decide how many of five fields are to be included. The first one, *Word*, is obligatory, but the others can be determined by the user. If you choose to have the field *Location* then the search is liable to be slowed up if there are a lot of finds for each word. The reason is that the program now records the location in each text of all the finds. When looking for rare forms, this option can be very useful. So use prudently.

Range of files for a word list

The range of a stop word list covers the four possible types: 1) *From first file*, 2) *Branch only*, 3) *Just current text*, 4) *Checked files*. These four ranges are also available when retrieving data from files in your corpus. The first three options refer to a section of the tree on the left of the screen, i.e. the entire tree, a branch or a single node. But there are cases where the files to be encompassed cannot be referenced by a section of a tree. In such cases what one does is to change the tree display to a list-type display (this can be done via the option *Tree or list display* in the *Display* menu, shortcut: F11). Now you can check files of your corpus as you wish. When the selection is made you activate the word list window again and proceed. The remaining parameters are discussed briefly below.

Keep punctuation Determines whether punctuation, found typically after a word, is included in the list generated.

Heed case of words If this is off (the default state) then lowercase and uppercase spellings of a word are regarded as the same.

Put file name in output If information on the file from which words stem is important then you should click on this parameter. If you include this information then you can later extract the words stemming from a particular file.

Get input word list Here you select an input word list from a directory listing.

Get stop word list The same command, this time for a file containing stop words.

Save as word list There are two basic means of saving word list returns to disk. The present one is the simplest: it will just save them as a plain ASCII file which can be loaded afterwards with any word processor or text editor.

Save as database The second means of saving is as a database. This is a table with information arranged in the form of rows (records) and columns (fields). You cannot load a database directly with a word processor or text editor. Instead you must use a database manager like *Corpus Presenter Database Manager*. It may require some getting use to but the advantage is one of much greater flexibility in the manipulation of data as you can access individual fields and records and extract data from these and arrange it afresh.

The database generated will have seven fields which correspond to the label in the word list window on the right of the screen.

- | | |
|--------------|------------|
| 1) WORD | 2) REVERSE |
| 3) FREQUENCY | 4) PERCENT |
| 5) FILE_NAME | 6) LABEL |
| 7) LOCATION | |

Statistics Assuming that a search has been done, this option will display the returns for the files examined, i.e. the number of unique forms, the number of words and the number of files.

Comment character There may well be lines which you do not want to have examined during word list generation. If these are preceded by an unambiguous symbol then you can enter this as comment character here and have such lines ignored.

Window This command will expand the word list and use the entire window for display. The option is a toggle so that clicking on the button again will revert to the previous state.

Shortcut: **Ctrl-W**

3.6 *Show word locations* This option provides a comfortable means of viewing the passages in any text which contain words from a list. The way one uses this function is as follows: first generate a word list from any text or texts using the *Make Word List* (see previous command but one). Save the list generated to a disk file making sure that you

include information from all fields when save the list to disk (the information on locations of words in files is essential for the current function to work properly).

Once a list has been made, start the present option and choose the option *Load word list* (on the top of the window). Select the list from the directory listing you are presented with. The list is loaded and the grid now displays information on each word form. Click on any word (or choose the option *Show tokens* on the top) and the screen changes. The list you see on the left contains all the occurrences of this word in the texts from which the word list was generated. Move up and down the list to view the locations of this form in the texts in which it occurs. You can copy out anything you wish from the text displayed on the right.

Shortcut: **Ctrl-S**

3.7 *Keyness in texts*

This term has gained a specific meaning in recent years by which it refers to the extent to which a text or texts show a specific stylistic profile when compared with another set of texts. Specifically, the term has been used when analysing the lexical profile of an author or author(s) in direct comparison with that of another group. By this means it has been possible to show how the style of an author or authors is distinctive vis a vis others of his/her time. This distinctiveness is the 'keyness' of the texts by the individual(s) in question. 'Keyness' is quantified by measuring the positive or negative difference in the lexis of author(s) vis a vis that of those in a reference corpus with which the former are compared.

Shortcut: **Ctrl-K**

3.8 *Analyse lexical clusters* Virtually any text will show recurring word patterns which may not be obvious at first sight. Here text processing software like the present program can be useful. With the current option *Corpus Presenter Text Tool* will comb through any text and break it down into segments consisting of between 1 and 8 words and then lists these segments with their frequencies. Such a breakdown is useful when analysing the style of an author or when examining typical collocations in a language. The following illustrates the principle with chunks of three words at a time.

This is the principle of Lexical Cluster Analysis

```

I think this would be an interesting project.
I think this
  think this would
    this would be
      would be an
        be an interesting
          an interesting project

```

There are many options at your disposal with the current command, for instance you can view clusters alphabetically or by frequency and you can display the context in which each occurrence was found. Results can be saved in a user-specified manner as a text or a database.

Shortcut: **Ctrl-U**

3.9 *View returns storage* There is a text editor window incorporated into *Corpus Presenter* which functions as storage space when you are collecting returns from various types of retrieval. This storage area can be accessed from any level of the program and you can add, delete, copy, move, paste any text to or from it as well as carry out certain basic types of formatting tasks. The text in the storage area can be stored to disk as a Rich Text Format file (with the extension `.RTF`) and hence be loaded into virtually any commercially available word processor on any operating system.

Shortcut: **Ctrl-R**

3.10 *Subsection of file names* A further means of restricting a search is to enter here a section of a file name which must be present for a text with this file name to be searched through in any retrieval operation. For instance, if you have a subset of files in your corpus which have the opening characters `20c_` for 20th century texts, then you could specify this as a subsection of file names necessary for searches to be carried out.

3.11 *Markup codes* Parts of texts can be excluded from retrieval operations. Such text is usually enclosed in markup codes, often used for comments, such as editor or compiler notes. You can either specify a code which indicates a comment line (this must be the first character on the line in question) or you can determine codes, an opening and a closing sequence, which encloses text to be excluded from searches.

Shortcut: **Ctrl-F9**

3.12 *Delimiters, punctuation* This option opens a window in which the various user-determined delimiters, for words and sentences, are shown. The symbols which have been specified as punctuation for *Corpus Presenter* can also be seen. In this window you can edit the various symbols to suit your needs. The settings here are used in the various search routines and are maintained across work sessions until perhaps changed later.

Shortcut: **Shift-Ctrl-F9**

4 Database

4.1 *Load a database* There is an inbuilt database editor in *Corpus Presenter* which offers users most of the options which they require for daily work with databases. This means that you do not need to load the *Corpus Presenter Database Manager* every time you wish to make a small alteration to a database in a corpus. The various options of the internal database editor are explained in detail in the section *How to use the database module* below.

Shortcut: **F4**

4.2 *Start external database editor* This option will load the supplied database editor *Corpus Presenter Quick Database Editor*. When you conclude the work session with the latter program an automatic return to *Corpus Presenter* is made.

Shortcut: **Shift-F4**

4.3 *Make a new database* With this option the supplied program *Corpus Presenter Make Database* is loaded and you can create a new database with it, starting from scratch or using a model database as you do so (see description below).

4.4 *Locate string in database* If the corpus you are processing contains databases then you can search for strings in these also. The function for this is to be found in the search menu. The way it works is as follows: *Corpus Presenter* looks for a database in the list of files for your corpus. When one is found the program reads in the structure and displays the locate function window with the fields of this first database shown in a list on the right. You may choose a field to search through, or you can demand that all fields of a database are combed through. The best results are achieved if all databases have the same structure.

4.5 *Extract records from databases* It is possible to extract any subset of records from the databases of the currently loaded data set by specifying a filter condition which the records are to meet to be included in the extraction process. The filter is constructed in the same manner as when you are processing a single database. You must furthermore enter the name for a database which is to receive the extracted records. This can be a new database or an existing one. If the database already exists then you can choose between overwriting it or just appending records at the end of the database file which you would do if you wish to preserve the previous contents. If the current data set does not contain any databases there an error message is issued. If the data set contains media files of different types (texts, graphics, sound files, databases) then only the databases are affected by the current option. By definition databases are those files with the extension DBF.

4.6 *Output all databases as text* The purpose of the current option is simply to put all the information in a data set at the disposal of users in text form. The output file which is generated is always a Rich Text Format file (to ensure that formatting is preserved) with a name which the user specifies. On completion you can load the output with your own program (see option *Your program* in 8. below).

5 Internet

5.1 *Corpus Presenter website* Here the internal browser is used to load the dedicated website for *Corpus Presenter* at <http://www.uni-due.de> Much information relevant to the understanding and use of *Corpus Presenter* can be found on that site.

5.2 *Browse on web* This option will activate an internal web browser with which you can navigate through the internet without exiting *Corpus Presenter*. There is a history function and a list option with which you can maintain a list of the web site addresses you might want to visit.

5.3 *Internet file editor* There is a text editor – *Corpus Presenter Internet Editor* – supplied with the current program which is intended specifically for editing internet files which are formatted according to the *Hypertext Markup Language* specifications. For instructions on how to use the editor, consult the online help contained in the program.

6 Display

6.1 *Set screen size* Here you can set the size of your screen by choosing from a selection of different values.

6.2 *Quick text display and retrieval* Flexibility of display and retrieval is a major concern in *Corpus Presenter*. To this end there is a powerful editor which can handle files of all types, including HTM/HTML files used as standard on the Internet. In order to increase the speed of text retrieval there is a special fast mode in *Corpus Presenter* which will vastly improve performance. This mode can be turned on here by checking the option *Use fast retrieval text mode* (these options can also be set from the retrieval level and when specifying parameters for a text search, see below for details). Technically, what happens here is that the ASCII mode for text strings is employed as opposed to the default Unicode mode which *Windows* normally uses.

Unicode strings consists of two bytes per text character. The advantage of this is that alphabets and syllabaries like those used in Russian, Chinese or Japanese can be displayed and processed under *Windows*. The trade-off here is that processing speed with Unicode strings is considerably slower because all strings are double the length of the text they represent and a lot of internal conversion takes place to hide this fact from the user to whom a piece of text is simply a piece of text. Now if you are processing a corpus with texts in any West European language (English, French, German, etc.) or even one with Old and Middle English symbols, then Unicode encoding is not necessary. So what *Corpus Presenter* can do is to convert any text to a single-character string type (technically called a byte array) for a retrieval task. This speeds up information retrieval considerably (anything from 50% to several hundred percent) and should be used where possible. The difference in retrieval times is especially noticeable with large texts, e.g. with a 500KB text the difference is nearly 1000% (sic!). However, if you are using search strings with special symbols (for Old and Middle English) or with diacritics from West European languages (such as umlaute from German or vowels with accents from French) then case insensitive searches will not always function correctly. In such cases you should not choose the fast retrieval mode.

Note also that there is an extended file interface included in all the major programs of the current suite which offers features not present in the standard interface offered by *Windows*. Specifically, the extended file interface shows you a tree of the current drive and so makes it easier to recognise where you are located in terms of folders. By default the internal file interface is used. If you choose 'No' here the standard *Windows* interface will be used until possibly changed at a later date.

The values for the parameters in this window are stored in the initialisation file for *Corpus Presenter* and are thus retained between work sessions.

Shortcut: **F8**

6.3 *Back colour for text display* The current option allows you to select a colour for the background during text display. The exact range available depends on the capabilities of the graphics adapter in your computer. Most computers nowadays have adapters which can handle at least 24 bit true colours which means that you can specify an exact shade of colour for display.

Shortcut: **Shift-Ctrl-F8**

6.4 *Set text font* With this option you can specify the font and size to be used for displaying plain text files. If your text files are in RTF format then the fonts/sizes specified within each text file will be used.

6.5 *Tree or list display* There are two basic means for displaying the structure of a corpus with the present program. Either you use a multi-level tree (default) or an indented list. The second option shows the labels in the upper half of the screen with levels indented. The corpus text files are displayed in the lower half of the screen. You can toggle displays to and fro. Note that with some options, the display reverts to the default tree mode automatically.

Shortcut: **F11**

6.6 *Check all files* With this option you can check all files in the current list.

Shortcut: **Shift-F11**

6.7 *Font for list display* Here the font to be used for files in list form can be specified.

6.8 *Remove checks from file list* This option is the mirror image of the previous one and will remove all checks after user confirmation.

Shortcut: **Shift-F11**

6.9 *How many files are checked?* This option will simply display the total number of checked files.

6.10 *Copy checked files to new corpus* This option will examine the list of files and copy those which are checked to a new directory. By these means you can create a subset of corpus files and then perhaps carry out some operation on these files. This option is particularly useful if you simply wish to experiment on a part of a corpus without affecting the integrity of the corpus as a whole.

Shortcut: **Ctrl-F11**

6.11 *Toggle full/divided screen* When a text file is associated with a node you may find it appropriate to use the entire screen to display the text. This command is a toggle and so switches back and forth between the two display types.

Shortcut: **F12**

7 Tree

7.1 *Collapse tree to left* When *Corpus Presenter* is loaded, only the nodes on the first level of the tree are displayed (for the sake of clarity). If you have expanded the tree and wish to reduce the number of levels shown then you should choose the current option.

Shortcut: **Alt-Shift-L**

7.2 *Expand tree to right* The opposite movement is the expansion of a tree. By this is meant that more and more levels are displayed when you choose the present option. Note that you can achieve the effect of the present and the previous options via dedicated keys or the toolbar on the top of the screen.

Shortcut: **Alt-Shift-R**

7.3 *Determine tree font* Here you can determine what font and what font size are to be used for displaying the tree on the left of the screen.

7.4 *Colour for tree* Allows you to choose, from the possible set of colours on your computers, one which is used for display of the tree or list of files (depending on display mode).

8 Miscellaneous

8.1 *Run List Processor* This option will load the program *Corpus Presenter List Processor* which allows you to manipulate lists in various ways. The program has three lists which can be used for the intake or depositing of data. The first two are input lists and you may copy information into the lists by either loading a file from disk or by copying suitable data from the *Windows* clipboard.

8.2 *Open internal text editor* There is a small text editor included in *Corpus Presenter* with which you can take notes while working with corpus files. The commands of the editor are a subset of the supplied word processor *Corpus Presenter Word Processor*. This text editor – termed here a *jotter* – can be used as a repository for the results of a retrieval operation, for instance if you choose to export results to an RTF text table on the retrieval level. From here the data can be copied to your own word processor by simply selecting text and depositing it in the *Windows* clipboard.

Shortcut: **Ctrl-J**

8.3 *Run word processor* There is a supplied word processor with *Corpus Presenter* which can handle RTF, HTM/HTML and plain text files and which is loaded at this point. You can use the *Corpus Presenter Word Processor* to process the files of a corpus if you wish. It is also useful for editing the results of searches which you may choose to save to disk.

8.4 *Load external text tool* The text editor of the *Corpus Presenter* suite – *CP Text Tool* – is loaded with this command. The editor has a range of options which make it useful for editing non-formatted texts, i.e. ASCII files. It is also ideal for editing large corpus files directly. The text editor can, however, handle both plain ASCII texts and Rich Text Format files.

8.5 *Start file manager* Here a supplied utility of the current program suite is loaded. With this you can also carry out most housekeeping tasks such as incremental backups of your files to an external disk and can view, copy, move and delete files flexibly, to mention just a few of the more obvious options of this program.

Shortcut: **Ctrl-F4**

8.6 *Calculation options* Activates the internal calculator in *Corpus Presenter* which has a variety of options. Any calculations done here can be stored in the Windows clipboard and imported into the internal text editor if required.

8.7 *Date and time* Simply displays the current date and time in the centre of the screen.

8.8 *Calendar* A calendar is displayed here with which you can check on days, months and years. By default the value for today is highlighted.

8.9 *Sound player* Under Microsoft *Windows* you can listen to sound tracks which are deposited in special files, usually with the extension WAV or MP3. Such files can be processed directly by the sound recorder which is part of the standard group of *Windows* programs, i.e. with the *Windows Media Player*. There is also an inbuilt sound file player in *Corpus Presenter*.

You can specify which sound player you wish to use by selecting either the *Windows Media Player* or the inbuilt audio playback module (in which case you leave the input line empty).

8.10 *Show wallpaper* An image can be displayed by *Corpus Presenter* which fills the screen and hides all other windows. This is technically called a wallpaper image because it covers the entire surface of the screen. The effect is like a screen saver and may be useful if you wish to conceal the present work surface.

Shortcut: **F7**

8.11 *Get wallpaper file* With this option you can specify what file to use as a wallpaper image.

8.12 *Opening image* When *Corpus Presenter* is loaded an image file is displayed, a so-called splash file. This can be determined interactively by you choosing an image file of your own (one which reflects the contents of your corpus). The file chosen here overrides that in the control file, should these be different.

8.13 *Your program* This option allows you to load a program of your choice. When this terminates an automatic return to the outset is made.

Shortcut: **F9**

8.14 *Set your program* The program you wish to run via F9 can be specified at this point. The name and full path of the program are stored in the initialisation file so that you do not need to repeat this procedure for each work session.

8.15 *Shell down to DOS* This option will switch directly to MS DOS, which will be apparent either by the entire screen twitching and displaying a blinking cursor on a black background or by a window opening with the same cursor and background. You can now execute any DOS command you like. To return to *Corpus Presenter* you type in 'exit'.

Shortcut: **Ctrl-F1**

9 Help

9.1 *Help text* Here you can load the help file. By means of the table of contents you can jump to the section which is currently of interest to you.

Shortcut: **F1**

9.2 *Initial help text* To facilitate the task of getting acquainted with *Corpus Presenter* a short help file has been included which can be displayed when the program is loaded. It can also be loaded here and you can additionally specify whether it should continue to be shown on load-up and whether an opening image should also be displayed.

9.3 *Load and keep help* This option will do the same but will keep the help file loaded so that you can consult it during work with *Corpus Presenter* without having to load it afresh each time.

9.4 *Manual for corpus* There are three important text files which should accompany any corpus (they are specified as items 3, 4 & 5 in the CPD control file). This the first and is the basic manual for your corpus. The file must be an RTF file and is displayed when the current option is chosen.

Shortcut: **Shift-Ctrl-F1**

9.5 *Goto corpus directory* A *Windows Explorer* window is opened with this command and the directory where the present corpus is located is automatically logged in.

Shortcut: **Shift-Ctrl-F6**

9.6 *Explore home directory* Similar to the previous command but in this case the home directory of *Corpus Presenter* is logged in.

Shortcut: **Shift-Ctrl-F9**

9.7 *About this program* Displays a small text about *Corpus Presenter*. This window also contains information about the current release of *Corpus Presenter* (presently Version 12 January 2010) and about the author.

9.8 *Removing Corpus Presenter* Here a window is displayed explaining the three short steps necessary to remove the entire *Corpus Presenter* suite from your computer.

NOTE Certain options are only available when the tree on the left is visible. Equally, other options can only be accessed when the text window fills the entire screen. In addition, note that there is a toolbar at the top of the screen (if you have left it visible). To find out what commands the tools activate, place the mouse cursor over a tool and leave it rest for a second. You are then shown a 'tip text' which described the associated function briefly.

Retrieving information (advanced search)

It is fair to say that linguists, after perhaps a period of browsing in a corpus, will be primarily interested in retrieving information from the corpus in question. For example, if you have looked at a comprehensive corpus like the *Helsinki Corpus* then you will probably be interested in seeing if any information of relevance to your own research interests can be gleaned from examining this corpus with appropriate software. You might want to check up on the attestation of a certain syntactic construction, on the occurrence of morphological forms or on the frequency of some lexical items. If the software you are using is *Corpus Presenter* then you will have to move to the retrieval level to achieve any or all of these goals.

The retrieval level is reached by selecting the option *Advanced search* in the *Search* menu (shortcut: Ctrl-D). The set of functions offered here allows you to locate virtually any string or strings in any of the texts of a corpus. For this to work properly certain items of information and certain parameter settings are required. The most important of course is the search string itself, or strings if you choose to carry out a double string search. Either or both of the search strings are entered in the boxes provided for this purpose in the retrieval parameters window. Below you will find a discussion of all the relevant options of the retrieval level and a discussion of the effects of specific settings.

Search parameters

A variety of options which can be set in various ways for text retrieval. The settings determine the behaviour of *Corpus Presenter* during a searches on this level. A parameter is set to 'on' by clicking in the small box in front of the text describing it. If set, a tick appears in the box. When this box is empty the parameter is not set.

- 1) *Heed case during search* If this parameter is not set then uppercase and lowercase letters are treated in the same manner, that is no distinction is made between capital and small letters. This also applies to any special symbols chosen from the list on the right.
- 2) *List negative finds* This option should be used with prudence as it will return all contexts which do not match the search parameters. For a corpus of any size, the results would be enormous and your computer would run out of memory at some stage. The option has only been included for those cases where users really know what they are doing and definitely require negative finds.
- 3) *Double string search* This type of search requires two strings, a first one which represents the left-hand section of a syntactic frame and a second one which is the right-hand part. A typical example of a frame would be a phrase or part of a sentence. For instance, if you wish to search for occurrences of *do* plus *have* in historical texts of English, say in the Helsinki Corpus, then one might enter the following.

Syntactic frame search
 Left-hand string *do*
 Right-hand string *have*

This would return finds like *do have*, *do certainly have*, etc. You can furthermore specify whether either or both strings are entire words or only a part of a word, see below.

Special symbols Many corpora contain historical and/or foreign language texts and so you may well wish to search for a string/strings with special symbols in it. The list on the right of the screen allows you to access such symbols. Double click on any symbol and it is entered into the current string. Note the two option buttons below the symbols list. These allow you to specify which of the two input strings a chosen symbol is deposited in.

History When a search is carried out, the string/strings you entered is/are deposited in the history list which appears on the right of the screen. This list can be saved to disk and retrieved at a later point. You can also have more than one history list. When *Corpus Presenter* is loaded the list used last is automatically retrieved and its contents fill the history array.

4) *Use input list 1; Use input list 2* Instead of simply entering a single item for string one or two you may wish to enter a number of forms. It may well happen that the string/word you are looking for – above all in historical corpora – occurs in more than one form. For instance, if you were looking at modal verbs in Middle English then you might want to treat *mai*, *maie*, *may*, *maye*, etc. as instances of ‘may’. This is done by creating an input list with all the possible spelling variants of ‘may’ and then using this for string one. The same applies to string two; and of course you could have a combination of input forms for string one and for string two.

Using input lists will slow down the performance of *Corpus Presenter* slightly as it must check on not just two forms for a double string search but on the multiplication of the number of input forms in list one by the number in list two. With modern, faster computers this should not be an issue, however.

5) *Find across sentences* A syntactic context which you specify for a frame search will probably occur within a sentence. For this reason you are given an option here which is ‘off’ by default. If you wish to deliberately search for a frame which straddles two sentences then set the current parameter to ‘on’.

The set of delimiters for sentences can be edited by the user (see the appropriate text box). For instance, if you were dealing with Spanish texts you would want to include the inverted exclamation or question mark symbols as possible sentence delimiters.

6) *Allow spaces between strings 1 and 2* A frame search normally aims at returns consisting of several words, i.e. a phrase. However, it is equally possible to search for a word using a frame. For instance, if you wished to find all instances of negated adjectives in a text then you could enter a frame consisting of *un* and *able* and specify that intervening spaces are not allowed by removing the tick from the box for the current option. Such a search would return such tokens as *unacceptable*, *unbearable*, *unthinkable*, etc. You could also use an input file list for the beginning of such words. If you had a list with *un*, *in*, *im*, *il* and second list with *able*, *ible* then you would also find *indescribable*, *impossible*, *illegible*, etc.

7) *Number of intervening items* If you are preparing a two string search then this parameter is of relevance as it determines how much material can occur between the first and second string for the context to be registered as a successful find. You can

furthermore specify whether the intervening items are words or characters. The latter would be significant if the search strings are intended to be part of a single word for a successful find to be returned. If this parameter is set to 0 then the left and right sections of the frame must be immediately adjacent. The maximum number of intervening items (characters or words) is 64.

8) *Amount of context returned* When searching for strings, *Corpus Presenter* can return the context in which it occurred. You can determine how much of this is shown by specifying how many words to the right and left of the string are to be returned.

9) *Return whole sentence containing find* This can be useful to see what sentences embody a structure which you might be searching for. Bear in mind that a sentence is defined as a syntactic structure which is bound by a sentence delimiter. You can determine the set of such delimiters by editing the appropriate input line on this level of *Corpus Presenter*.

10) *Delete returns from previous search* Unless you wish to accumulate returns in a large composite list, you should select the current option. This will ensure that any previous list is deleted before starting a new search. However, if you wish to retain the returns from the previous search then tick the box here. This option only applies to Only applies to *RTF Text File* and *Line List* returns. Make sure you do not alter the manner in which returns are shown and you do not exit the advanced search level. You may, however, open the parameters window and change something, such as a string for the search, or choose a new input list. If you wish to, you can save returns in a single- or multi-line grid and reload at some later point.

11) *Find delimiters* By default these consist of a left and right angular bracket. You can, however, enter any symbols you like which you feel might visually set off a search string in a return context.

12) *String position in word* This is a simple parameter which determines whether the units used for a search operation are entire words or only sections or indeed whether this consideration is relevant at all for a search. Basically you can specify that a string is to be treated as an entire word, the beginning or end of a word or specify that it may occur anywhere in a word, i.e. that its status as part of a word is immaterial for the pending search.

Bear in mind that *Corpus Presenter* uses mechanical means for determining if a string is a word, i.e. it looks to see if the string is preceded by a tab stop or a blank or is the first item on a line and then checks to see if it is followed by a blank, tab, comma, full-stop, colon, semicolon or is the last item on a line. The set of word delimiters can be determined by the user editing the list provided in the relevant text box in the search parameters window. This list is stored to disk and re-read in later work sessions.

The possibilities here can render a search, and hence the returns, more accurate. For instance, if you wished to search for the perfective construction of Irish English as in *She's after selling the car* you could enter *after* as String1 and *ing* as String2 and specify that the position of the latter is at the end of a word. This would ensure that in a sentence like *She's after bringing the dog* only the final *ing* is returned as a valid find for String 2.

On the other hand you could choose the setting *Beginning of word* in a case like that discussed above under frame search. If you specified that *do* was only to be returned if

found at the beginning of a word then cases would be registered like *don't* which would allow for negated forms of *do* among your retrieval results.

13) *Intervening items* The left and right of the frame can be separated by a specifiable number of intervening items (characters or words). If this is set to 0 then the left and right sections of the frame must be immediately adjacent. To allow simple adverbs in the above example you would set the type of intervening item to words and the number to 1. The maximum number of intervening items (characters or words) is 64.

14) *Save profile* All the parameters you may have specified for a certain search can be saved with the current option. The suggested extension for a search profile file on disk is *.SPR* which you are advised to keep to in order to recognise such a file in future.

15) *Get profile* Here you can load a set of previously specified parameter settings from disk. The dialogue window which opens by default lists all files with the extension *.SPR* which is that expected by *Corpus Presenter* for search parameter files.

16) *Grid options* There are a number of ways for *Corpus Presenter* to deposit returns from a retrieval run. The simplest is as a plain text which can be copied directly via the normal *Cut* and *Paste* keys of Windows. The next is as a line list which is slightly more structured in that each return occupies a separate line in the list. The most flexible type of output for retrieval returns is doubtlessly a grid. This is a lattice of rows and columns. There is one row per find. The number of columns depends on the settings for the parameters activated by clicking on the button *Grid options* (see next two options for details).

a) *Multi-line grid* This type of repository allows for the context to contain more than one line. For instance, you might find it useful to have several lines before and after the find for a string or strings. This is possible with the current option. There is one important restriction, however: a multi-line grid cannot be saved as a database as the fields of a database can only accept single lines of text.

Possible columns Apart from the column *Text section*, which is obligatory and which cannot be unchecked, there are a number of other columns which you can add to a multi-line grid by checking them. The columns *Location*, *File name*, *Node label* will include the numeric position of a find in a text, the name of the text file from which it derives and the label for the node in the tree which it occupies. In addition you can add up to 4 user columns. Here you can enter information which you might want to add to that automatically returned by *Corpus Presenter*.

Marking rows in grids In both the multi-line and the single-line grids you can mark rows discontinuously by holding down the Control key and clicking on a row with the left mouse button. Bear in mind that selected rows can be copied to text at any time by choosing the option *Copy to text window* which is visible after a set of returns are displayed on this level.

Editing options When you edit a cell in a multi-line grid you will notice that a button appears in the top right-hand corner. If you click this the contents of the first column of the current row appear in a small text window. You can edit a stretch of text, store macros to disk (click button in bottom right-hand corner), load a new file, etc.

b) *Single-line grid* Here only an amount of the context to the left and right of a find is returned (with a positive find). These returns can be used as the input to a database directly, for instance you could process the results with the database editor in the *Corpus Presenter* suite.

Possible columns Apart from the column *Find* which is obligatory and which cannot be unchecked, there are a number of others which you can insert for a single-line grid by checking them. The left and right flank for a find (trimmed, i.e. with trailing or preceding blanks removed, or not) can be included and you can specify whether the delimiters, set in the main search parameters window, should also be included. The columns *Location*, *File name*, *Node label* will include the numeric position of a find in a text, the name of the text file from which it derives and the label for the node in the tree which it occupies. In addition you can add one or two user columns. Here you can enter information which you might want to add to that automatically returned by *Corpus Presenter*.

Sorting a single-line grid One inherent advantage of the single-line grid is that it allows one to sort any field in either ascending or descending order. All you do is click on a column heading and the entire grid is sorted according to this column. The sort option is a toggle: clicking a heading once will lead to an ascending sort on that column, clicking again will cause the data to be sorted in descending order.

c) *Separators between output fields* There are different ways of separating the fields for each return (assuming that the repository type is a text). Four common options are offered here, along with the choice of having no separator. These various types are a matter of personal taste; it is best just to try them out and see how you find them.

d) *Output file in Corpus Presenter Table Editor format* There is a supplied program for editing tables generated on the advanced search level, namely *Corpus Presenter Table Editor*. If you choose to save returns in this format then you can edit these separately with the table editor and export them from there to a text editor. One of the advantages of this is that the table editor can handle several tables at once. If you have several return sets you can edit these as a group later independently from *Corpus Presenter*. For this option one must use the *Multi-line grid* for returns.

Exporting rows from a single- or multi-line grid can be sensitive to three settings. Essentially, you can specify that only selected rows are exported, that only the text in returns, i.e. no information about location and filename, and that only those columns of returns which are currently visible are exported.

When you are viewing returns the large buttons at the top of the screen change. On the right-hand side is one labelled *Goto text*. If you click on this then the returns window is hidden and the text window beneath shows the text where the currently highlighted find was made. The find itself is selected (shown as white lettering on a black background). You can bring the returns window to the foreground again by clicking on the button *Last results*.

Cocoa parameters

One means of specifying various items of information about a corpus text is to mention these in a header at the beginning of each file. A system which is quite widespread among corpora is the *Cocoa* parameter set. This consists of up to 32 parameters with typical

settings for certain file types. For instance, the texts of the *Helsinki Corpus* are all encoded with a *Cocoa* header in which information is given about a following text. The settings can be used in *Corpus Presenter* to determine what files are examined during a retrieval operation. The way this is done is outlined in the following.

To determine a setting you copy a *Cocoa* parameter from one of the text files of your corpus, say <V PROSE> or <X MALE>, into the *Windows* clipboard by marking the line and pressing CTRL-C when you are in a text program such as the *Corpus Presenter Text Editor*. Now move to the current window, click the text line at the bottom of the screen and retrieve the contents of the clipboard with CTRL-V. You then double click the position in the settings list on the right-hand side of the screen where this parameter belongs. Repeat this procedure for as many *Cocoa* settings as you require for the impending search. Settings which are empty will be ignored.

```

1   <B = 'name of text file'>
2   <Q = 'text identifier'>
3   <N = 'name of text'>
4   <A = 'author'>
5   <C = 'part of corpus'>
6   <O = 'date of original'>
7   <M = 'date of manuscript'>
8   <K = 'contemporaneity'>
9   <D = 'dialect'>
10  <V = 'verse' or 'prose'>
11  <T = 'text type'>
12  <G = 'relationship to foreign original'>
13  <F = 'foreign original'>
14  <W = 'relationship to spoken language'>
15  <X = 'sex of author'>
16  <Y = 'age of author'>
17  <H = 'social rank of author'>
18  <U = 'audience description'>
19  <E = 'participant relationship'>
20  <J = 'interaction'>
21  <I = 'setting'>
22  <Z = 'prototypical text category'>
23  <S = 'sample'>
24  <P = 'page'>
25  <L = 'line'>
26  <R = 'record'>

```

NOTES The location procedure only applies to text files, not to images, sound files, databases (see below for this) or to dummy text files which are place holders for branch nodes without a specific file reference, e.g. a superordinate node to a group of texts.

The location function in this program is a facility which permits easy searching of texts. There are other programs which are more powerful in this respect and which allow searching for syntactic contexts. Such software usually requires that all the texts of a corpus are indexed prior to operation and is essentially different in design and purpose from *Corpus Presenter* which is intended to present all information in any chosen corpus in an easy-to-grasp, intuitive manner.

Range of files for a search

The range for a search can be one of five possible types: 1) From first file, 2) Branch only, 3) Just current text, 4) From the current text to the end of the tree, 5) Checked files. These five ranges are also available when retrieving data from files in your corpus. The first four options refer to a section of the tree on the left of the screen, i.e. the entire tree, a branch or a single node. But there are cases when the files to be encompassed cannot be referenced by a section of a tree. In such cases what one does is to change the tree display to a list-type display (this can be done via the option *Tree or list display* in the *Display* men, shortcut: F11). Now you can check files of your corpus as you wish. When the selection is made, you then activate the word list window again and proceed.

Files affected by a search

Recall that the most common file ending for the texts of a corpus are: 1) .ASC (Ascii files), 2) .TXT ([Ascii] text files), 3) .RTF (Rich Text Format), 4) .HTM(L) (Hypertext Markup [Language]). These are the types of files which can be dealt with directly by *Corpus Presenter*. It may happen that your corpus has a special extension for its own files, or as in the case of the *Helsinki Corpus*, there is no extension. All such cases involve plain Ascii files and can of course be processed here. If your corpus is mixed, then those files which are recognisably not text files will be ignored. This means that if you carry out a text search on the files of the supplied corpus, then the images and sound files are ignored. To be precise, the following files types are ignored during a text search:

.DBF <i>dBASE</i> database	.JPG image file
.BMP image file	.PCX image file
.GIF image file	.TIF image file
.ICO image file	.WMF image file
.WAV audio file	

The first type of file — *dBASE* database — is a special type and can be the object of a search with *Corpus Presenter*. For this one must use the option *Locate string in database* in the *Database* menu. Consult also the section *How to use the database module* below.

Note that many nodes in a tree may contain a reference to a file `DUMMY.RTF`. This is an empty file which is used as a placeholder only and is ignored in all retrieval operations.

By choosing the second item in the *Returns* menu or by clicking on the third button on the left of the toolbar (a square with a horizontal line through it) you can toggle between the split screen and full screen mode. When the former is active then moving in the list of returns will automatically show in the bottom half of the screen the position in the text where the find was made. You can copy the find and its immediate ontext by pressing F4. This way you can collect various finds in a text window and afterwards copy these to your word processor, for example.

The split screen mode is also available on the *Basic search* level.

Advanced search options

1 Search

1.1 *Search through texts* This command begins a search using the parameters entered by the user (make sure to check these if you do not recollect what they are). You can interrupt a search by either pressing the Escape key or click on the *Stop* button at the top of the screen.

Shortcut: **Ctrl-T**

1.2 *Statistics from last search* Assuming that at least one search has been done, this option will display a window with the information on the last retrieval operation.

Shortcut: **Ctrl-L**

1.3 *Corpus information* This option will display information about the corpus dataset file which is used to provide access to the current files.

Shortcut: **Ctrl-I**

1.4 *Find string in text window* When you enter the retrieval level, the contents of a file will normally be displayed in the initial text window. The present function will search for a string in this window. For the function to work, the text window must not be covered by a set of returns (you can hide these if you like).

Shortcut: **Ctrl-F**

1.5 *Get string in returns* Assuming that you have carried out a search you can now look for any string suspected of being contained in the returns. The selection is updated to reflect the row of a grid which has the string you entered, should this be present. A search can be carried out automatically and the results can be stored to the Windows clipboard.

Shortcut: **F11**

1.6 *Retrieval mode* There is a special fast retrieval mode available in *Corpus Presenter* (see section *Considerations of speed* below for details) and by checking the box you can turn this mode on. You can also do this by clicking on the corresponding image on the parameter specification level.

Shortcut: **F8**

1.7 *Markup codes* Parts of texts can be excluded from retrieval operations. Such text is usually enclosed in markup codes, often used for comments, such as editor or compiler notes. You can either specify a code which indicates a comment line (this must be the first character on the line in question) or you can determine codes, an opening and a closing sequence, which encloses text to be excluded from searches.

Shortcut: **Shift-F8**

1.8 *Cocoa settings* Values for Cocoa parameters are specified in a separate window. This can be done here or from the search parameters window.

Shortcut: **Ctrl-F8**

1.9 *Parameters for search* Here the window in which you enter the parameters for a search is opened.

Shortcut: **Shift-F9**

1.10 *Exit to desktop level* Leaves the retrieval level after user confirmation. Remember to save any retrieval returns to disk, if you want to keep them, before you leave as you are not warned about these when you exit from *Corpus Presenter* entirely.

Shortcut: **Ctrl-Q**

2 Returns

2.1 *Store find and context* Assuming that the split screen mode is active (for this either a line grid or a multi-line grid must have been chosen to return finds) then you can store the find in the bottom text window and its immediate context in a small text window which appears in the top right hand side of the screen. This option is cumulative, i.e. you can use it to gather several finds and their contexts. Make sure you save the contents of the text window to the Windows clipboard (if you need them) before closing the window.

Shortcut: **F4**

2.2 *Toggle split screen* This option will switch between a full screen and a split screen when displaying finds. If the latter mode is active then moving the bar in a grid of finds will lead to the text from which they stem being shown automatically in the bottom window with the find in question highlighted. Use the previous option to copy finds and their contexts to a separate text box.

Shortcut: **F5**

2.3 *Save returns to disk* Once you have a set of returns (from a retrieval run) you can choose to save them to a text file. Note that you may determine here whether to save all rows in a grid or just those which you have selected (highlighted) in the grid. There are four basic types of file which can be used as output on disk.

<i>File type</i>	<i>File extension</i>
1) Plain text file (simplest form)	.OUT
2) Single-line grid (one row per line)	.GRD
3) Reloadable Table Editor format	.TBX
4) Database (finds are stored as records)	.DBF

The choice of file type depends ultimately on what you want to do with the results. If your aim is to have the results transferred to a table in a text file then you should choose type (3). If you wish to process the data in a database management environment, then choose

type (4). Should you wish to have the returns in a grid in which there is one row for each find and where columns can be determined flexibly by the user, then choose type (2). Type (1) is the simplest of all and can be used where you are just experimenting; this storage type for returns is the fastest of all as the returns do not have to be formatted.

Shortcut: **Ctrl-S**

2.4 Output options Here you can specify how a multi-line grid or a single-line grid is to be arranged for retrieval returns. This is also possible from the search parameters window.

Shortcut: **Shift-F9**

2.5 View returns storage There is a text editor window incorporated into *Corpus Presenter* which functions as storage space when you are collecting returns from various types of retrieval. This storage area can be accessed from any level of the program and you can add, delete, copy, move, paste any text to or from it as well as carry out certain basic types of formatting tasks. The text in the storage area can be stored to disk as a Rich Text Format file (with the extension `.RTF`) and hence be loaded into virtually any commercially available word processor on any operating system.

Shortcut: **Ctrl-R**

2.6 Put returns in storage After user confirmation the contents contained in the returns text box are transferred to the returns storage text editor. If you wish to put returns from a grid into the storage area then first select the rows you require and then store them to the *Windows* clipboard. Activate the returns storage text editor by pressing `Ctrl-R` and paste the clipboard contents into the storage area.

2.7 Cumulative clipboard You will have noticed that *Windows* only allows you to deposit one item of data, such as a piece of text, in the general clipboard at any one time. The present command allows you to keep depositing stretches of selected text in an internal clipboard and so collect text cumulatively. When the cumulative clipboard window is open you can retrieve text from the *Windows* clipboard or you can choose to copy part or the whole of the cumulative clipboard text to either the text editor or the storage window assuming that you have called it from either of these modules.

Shortcut: **Ctrl-Y**

2.8 Delete selected rows This option will remove any highlighted rows from a grid after user confirmation.

2.9 Delete current row This option will remove the current row from a grid after user confirmation.

2.10 Copy selected column You may copy any column in a grid by first clicking on the column header (this will select it) and then activating the current command.

2.11 How many rows selected This option show say how many rows are selected of what total. A percentage is also calculated.

Shortcut: **Ctrl-W**

2.12 *Select all rows / text* With either a single-line or a multi-line grid the present option will select all the rows of the grid.

Shortcut: **Ctrl-A**

2.13 *Show last search results* If you have hidden the results from the last search (via the following option) then this option will render them visible again.

Shortcut: **Ctrl-U**

2.14 *Hide results window* The retrieval results occupy the entire screen and hence if you wish to view the text which was displayed before you began the retrieval operation then you must hide the results display. This option will do this without, however, deleting the results from memory.

Shortcut: **Ctrl-H**

2.15 *Goto returns area* The grid in which the list of files from a search or searches and the numbers of finds per file are stored can be shown via the current option. This means that you do not have to repeat a search to view the grid. From here you can save the grid contents to a database and use this for chart generation in Microsoft *Excel*.

Shortcut: **Ctrl-G**

2.16 *Export rows as HTML file* This option will export the rows of the returns grid (all or just those selected) as a HTML file which is then loaded with the internal browser. The returns are displayed one per line with the find highlighted by a special colour if you wish. You can also choose to have the table border in the resulting HTML displayed or not.

Shortcut: **Alt-Backspace**

2.17 *Export to RTF text table* This option has been included to allow users to transfer returns from a multi-line grid directly to a table in a text editor. The text editor within *Corpus Presenter* is activated for this task and the data of the current grid is copied. Before this step you can specify if only the visible columns and/or only the selected rows of the current grid are to be exported to the table. Note that an RTF text table can be loaded directly into virtually any commercially available word processor without loss of data and formatting.

Shortcut: **Ctrl-B**

2.18 *Export to internal text editor* The text which is contained within *Corpus Presenter* is loaded via this option. If you wish to process text from a search in this module then you should copy returns to the Windows clipboard and then paste them into the text window you are presented with.

Shortcut: **Ctrl-N**

2.19 *Export returns to Excel* The returns from a search can be transferred to a Microsoft Excel table via the current command. Here you must specify a file name for the table to be created. There are a few other parameters to be set, notably if you wish all rows from a returns grid to be deposited in the Excel table or only those which are currently selected.

Shortcut: **Ctrl-E**

3 Load/save

3.1 *Get profile from disk* A search profile is a set of values for the search parameters which are customisable. These must have been stored to disk on some previous occasion for the present option to work. Such a file is a plain Ascii file with the extension *.SPR*.

Shortcut: **Shift-Ctrl-F8**

3.2 *Save profile to disk* If you have entered a number of parameters for a search which you feel you are likely to use again, then you can save the entire set to disk under a name which you specify with the present option.

Shortcut: **Shift-Ctrl-F9**

3.3 *Load database into line grid* Assuming that you have a database which resulted from carrying out the following command you can then load its contents with the present option. This is useful if you wish to view returns without carrying out a search again.

3.4 *Save line grid as database* The present option will save the contents of the line grid as a database. The fields of the resulting database have names which are identical to the columns of the returns grid. There are as many records in the database as there are returns in the source grid.

Note. You cannot save the contents of a multi-line grid in the same fashion the cells of such a grid can contain several lines and this is not permitted in a database (the contents of a cell must be a single line of text).

3.5 *Load returns for multi-line grid* If you have saved retrieval returns on a previous occasions to a reloadable file (with the extension *.TBX*) then you can now reload these results into a grid which *Corpus Presenter* opens for this purpose.

Shortcut: **Ctrl-K**

3.6 *Save returns as multi-line grid* Here you can save the contents of a multi-line grid in a format which can be re-loaded with the previous option. The program *Corpus Presenter Table Editor* can also process such files. These always have the extension *.TBX*.

4 Display

4.1 *Set colour for source text* The source text is that which is displayed when you first move to the retrieval level and derives from the text file associated with the node in the tree which you were last located at. With this option you can set the colour used for display of this text.

4.2 *Set colour for returns* A similar option, but this time for the returns grid or list (depending on what is requested).

4.3 *Size for returns grid* You can resize a returns grid dynamically by dragging on the borders of cells and rows on the edges of the grid. Alternatively, you can use the current option and choose from a number of pre-determined sizes.

4.4 *Font for returns grid* Here the font name and size can be determined for use with a grid. This parameter and the others in this menu are stored in the initialisation file for *Corpus Presenter* and hence retained from one work session to the next.

4.5 *Font for text returns* Assuming that you have chosen an RTF text file as the repository for returns from a search then you can specify the font to be used for any part of the returns text via the current option.

Shortcut: **Ctrl-1**

4.6 *Hide finds from view* There are occasions when it might be opportune to *not* display the finds in a returns grid. For instance, if you were using a corpus for teaching purposes you might want to hide the finds to have students guess what words could fit into the contexts in which the finds were found. This option is only available when a single-line grid is used as the repository for retrieval returns.

4.7 *Restore finds column* This option will simply render the finds in the relevant column visible again.

4.8 *Determine collocation* This option serves to render the context in which a string is found more explicit. Essentially what it does is to take either the left flank of a find or the right flank or both and then split this/these up into further fields (up to a total of eight). By these means you could examine, say, the second item to the left of the find or the third to the right. Recall that you can sort the columns of a single line grid by clicking on the column heading. The first click leads to an ascending sort, the next to a descending sort. This option is only available when a single-line grid is used as the repository for retrieval returns.

Shortcut: **F12**

4.9 *Columns to view* It is possible to select only a subset of columns in a returns grid for display. With this option you select columns from a list in a window which now opens. The advantage of this function is that when exporting data from a grid (to a file or a text table) you can specify that only data from visible columns be copied.

Shortcut: **Shift-F5**

4.10 *Toggle syntax colouring* Assuming that the files you are using with *Corpus Presenter* are unfiltered HTML files then you can choose to highlight all HTML tags with the current option. This makes it easier to recognise what is actual text in such files.

Shortcut: **Ctrl-F5**

5 Miscellaneous

5.1 *Main help text* Here you are shown the text files of the help system. There is a special section on retrieval. Click on the node with this heading to access this information.

5.2 *Calculation options* Activates the internal calculator in *Corpus Presenter* which has a variety of options. Any calculations done here can be stored in the Windows clipboard and imported into the internal text editor if required.

5.3 *Open internal editor* Here you can switch to the internal text editor (jotter) of *Corpus Presenter*.

5.4 *Load database module* The internal database editing module can be loaded via this option. On termination you are automatically returned to the advanced search level. The database editing module can be used to process search returns which have been saved to disk as a database, for example.

5.5 *Run List Processor* The utility with which you can manipulate lists (typically the output of some operation within *Corpus Presenter*) is loaded here. Once you terminate the *List Processor* you are returned to the current level of *Corpus Presenter*.

The database module

The database module which is built into *Corpus Presenter* and all of the other major programs of the current suite offers you a comfortable interface to any database you might have on your hard disk. To load a database you simply choose a file with the extension .DBF from a directory listing. If you have not chosen a database, then you are prompted to select one from a listing.

Recall that a database is a structure which consists of a fixed number of columns (or fields) and a variable number of rows (or records). A database is often called a table and can be compared to a box of index cards.

Structure of a database (table)

ROWS	COLUMNS			
	Field 1	Field 2	Field 3	etc.
Record 1	Adams	George	1981	Irish names...
Record 2	Bliss	Alan	1976	Language contact...

Record 3	Hogan	James	1927	The English language...
etc.				

Each record corresponds to a single card and the fields are used to store the information which you would typically enter onto an index card.

Data and display in the database module

<i>Source of data</i>	<i>Means of display</i>
Database on disk	1) Grid mode (columns and rows, see above) 2) Index card mode (a record at a time) 3) Browse view mode (first field list + card)

A few points should be borne in mind about how the database grid behaves. You move around the grid using the arrow keys. To edit a field, click the left mouse button while you are inside the field. Initially the entire contents of a cell are selected so that typing leads to these being overwritten. You can use the normal Windows keys when editing, including Ctrl-C and Ctrl-X, Ctrl-V for *Copy*, *Move* and *Paste* respectively.

You will notice that if you press Delete when one or more records are selected then these are removed from the grid. Technically, they are marked for deletion in the underlying database but not physically removed from the database on disk.

1 File

1.1 *Info on database* Show a windowful of essential information on the current database.

Shortcut: **Ctrl-I**

1.2 *Database structure* Displays the structure of the current database, including the name, type and length of each field.

1.3 *Quit database viewer* After user confirmation, the database module closes down. The database must be re-opened when a new call to the module is made. The database used on the previous occasion is automatically loaded if this is accessible from disk.

Shortcut: **Ctrl-Q**

2 Navigate

2.1 *Move up 20 records* Positions the selection twenty records backwards in the database.

2.2 *Move down 20 records* Positions the selection twenty records forwards in the database.

2.3 *First column* Moves the selection to the first column of the current row.

2.4 *Last column* Moves the selection to the last column of the current row.

3 Display

3.1 *Size of cell* The size of grid cells can be altered in two ways. The first is by dragging on the edge of the cell in the grid border (the mouse pointer changes to a vertical bar which indicates that this option is now active). The second is by choosing a size from a selection of six which are offered in a popup window. These values are retained across work sessions.

Shortcut: **Ctrl-S**

3.2 *Font for cell* The font name and size as well as word attributes (bold, italic, etc.) can be specified for grid display here. These values are retained across work sessions.

Shortcut: **Ctrl-T**

3.3 *Window size* This option simply toggles between a full screen and a partial screen size for the database grid.

Shortcut: **Ctrl-W**

3.4 *Fields to view* You may not always wish to view all fields of a database so this option has been included to allow you to suppress fields by selecting those you wish to view from a list of the fields of the current database. Note that when exporting data, you can choose to have just the visible fields of each records transferred to the output text.

Shortcut: **Ctrl-Y**

3.5 *Background colour* Here you can specify the background colour to be used for the grid in which the database is displayed.

3.6 *Save column widths* If you alter the standard widths of cells in the database grid (you can do this by dragging the dividers at the top of the grid) then you might like to store the values for column widths and have these at your disposal in future. This option is especially appropriate where the fields of a database are very different in size as it is sensible to use wide cells for long fields. The values for cell widths are stored in a file called `CPRESENT.CLW` (where `.CLW` stands for „column widths“).

3.7 *Get column widths* This option will load the file `CPRESENT.CLW` (if it can be located) and will adopt the values for cell widths which were deposited in it via the previous option when this was activated on some previous occasion.

4 Search

4.1 *Find string* You can search for contents in databases via the current option. You enter a string and begin the search. The first find is displayed and selected. A few points must be remembered here. The search begins at the current row so that if you wish to

search the entire database you must first of all go to the beginning. Searches can be case sensitive, for this you check the box under the search string input line. You can specify whether all fields are to be combed through during a search or just one particular field.

Shortcut: **Ctrl-F**

4.2 *Next find* This option just repeats the search, starting from the current grid position.

Shortcut: **Ctrl-N**

5 Browse view

The data from a database can be displayed in a number of ways. When this module opens the data is shown in a grid which consists of as many columns as there are fields in the source database and as many rows as there are records. There are two other display types available: 2) index card display and 3) the simple browse view method. The former display type will show you the contents of a single record along with the names of the fields in a separate window. You can page in the database as will. In addition you can hide the field names and you can copy the contents of a record to the *Windows* clipboard by clicking on the appropriate button. Alternatively, you can mark a section of a record by dragging the mouse across the text you require and then pressing Ctrl-C.

6 Index card

The browse view method is especially helpful as it shows the entries for the first field of every record (say surname in a bibliographical database) in a long list and by clicking on a name you jump straight to the record in question. You can also type the first letter of a name to locate it in the list.

7 Export

Most users will wish to extract information from databases and import this to a text file for further processing. This is done via the current option. You specify whether all records or only those which are selected are to be exported and then whether to use a report form or not. In addition for the copy operation you say whether a separate line is to be used per field or whether all fields of a record are to be placed on a one line. But in fact the most flexible output option is to export data using a report form.

A report form is a filter which is used to organise the input data in a certain fashion for the ensuing output file.

<i>Input</i>		<i>Filter</i>		<i>Output</i>
Database	→	Report form	→	Text file

There is a default report form which will output bibliographical data in a standard fashion. Use the supplied program *Corpus Presenter Report Database* with which you can generate report forms quickly and easily. The simplest way to do this is to copy the supplied report form `default.dbr` to a new file and then adapt this to suit your needs. When the data has been exported you are asked if you wish to view it with the default word processor, *Corpus Presenter Word Processor*.

8 Copy

8.1 *Mark filtered records* Usually a database contains many more records than one wishes to deal with at any given time. It is normal for users to want to extract only a selection of records from the entire database and to specify the condition for this selection. This is done with the present option which will select the records which match a filter condition specified by the user.

A filter condition can consist of up to three parts. For each part you specify a field and the contents it is to have to match the filter. Or the contents which it is not to have. With numeric fields (such as a field `YEAR` in a bibliographical database) you can state whether the filter is to accept values greater than, smaller than, equal to or not equal to your entry. Furthermore, you specify the connector which is to apply between the parts of the filter. There are three logical possibilities here: 'and', 'or', 'and not'. The default connector is 'and'.

A filter can consist of one, two or three parts. Fill in the parts from the top downwards. If the contents field is empty, then the filter is assumed not to contain that part, i.e. for a one part filter you enter something for the first contents field, for a two part filter you enter something for the first and second contents fields and for a three part filter you enter something for all contents fields.

Note that the selection of records is cumulative, i.e. any selection already present is retained. This fact can be useful if you wish to apply two filter conditions in succession to the same database. By these means you could achieve an even more accurate selection via filtering.

Shortcut: **F5**

8.2 *Selected records* To select records in the grid you click on the extreme left-hand side of a row, i.e. outside of the data area of the first field. To mark more than one record, hold the Control key depressed and click with the left button. The current row is selected maintaining the previously selected rows.

If you now choose the current option you are prompted to enter the name of a database to which you can export the selected records (rows). If the database exists then you must confirm overwriting it. If it does not exist, it is created. The selected rows are copied to this target and they are then shown in a window which displays the second, target database.

Shortcut: **F8**

8.3 *View second database* Assuming that you have already activated the second database by selecting records and exporting them to this target you can now view this database. The second database can be edited like the main database. Any changes made in the grid are committed to disk on closing the window.

9 *Text* If there is a text file with the same name as the database and the extension `.RTF` then this is loaded into a text box and displayed in a window below the database. The advantage of this is that you can maintain text files associated with specific databases and view the texts once the database has been loaded.

10 *Help* Shows the database help text. The supplied file `DATABASE.TIP` must be present in the home directory of the current program for this to work correctly.

N.B.: All databases must be in the *dBASE* format. Such files have the extension *.DBF* (= database file) and are in the most common format for databases on personal computers. If your databases are in a different format then you can store them in *dBASE* format from your usual database management program. Afterwards they can be processed with the current software.

Structure of a data set file

A data set file is a small text file which contains all the information needed for displaying the files of a corpus correctly in tree form. For each node of a tree three pieces of information are specified. In addition there are 11 parameters which are set at the beginning of the file and which determine the location of the corpus files and the manner in which they are displayed.

A control file for the *Corpus Presenter* is a plain ASCII file and can be edited with the *Corpus Presenter Text Editor*. Be careful **not** to save this in RTF or HTM format as it would no longer function properly as a control file. Note that any line in a control file which begins with a semicolon is regarded as a comment and ignored.

It is not advisable to alter the text file references unless you know **exactly** what you are doing. Always save the current control file under a new, temporary name, if you alter anything. Then check that it functions properly before deciding to keep it as an original version. There follows a brief description of the 11 parameters at the beginning of a control file.

- 1) Directory where all and only the data files are located. This very is important for a corpus like the *Helsinki Corpus* where the text files have no extension and where there is no formal means of identifying which files in a given directory belong to a corpus and which do not. Where no extension is used, the default assumption is that all files in a specified data directory are corpus files. If there is no path before a file name then the data directory is used; a path for a file will override this directory. A section of a path can be used. In this case the section is assumed to refer to subdirectories below the primary corpus data directory. If you wish to use the current directory as the data directory, i.e. the directory in which the present file is located, then just enter a single dollar sign: \$
- 2) Wallpaper file (full path necessary)
- 3) Name of manual file (full path necessary)
- 4) 'Frequently Asked Questions', FAQ file (full path necessary)
- 5) 'Fact Sheet', FACT file (full path necessary)
- 6) Font for text display (legal names: Arial, Courier, Courier New, Garamond, Letter Gothic, Line Printer, Modern Tahoma, Terminal, Times New Roman).
- 7) Size of font (legal values 6, 7, 8, 9, 10, 11, 12, 14, 18, 20, 24). The font name and size given here only apply when the text files are standard ASCII files. If they are in either RTF (*Rich Text Format*) or HTM/HTML (*Hypertext Markup Language*) format then font information is contained in each file header and this takes precedence over any specification here.

- 8) Zoom factor (legal values 50, 75, 90, 95, 100, 105, 110, 115, 125, 150, 200). This only applies if quick text display is **not** selected.
- 9) Standard extension of files in this corpus. A corpus may use a typical extension for its files, as with *A Corpus of Irish English* — by the present author — the (text) files of which all end in .CIE. Be sure to enter the dot before the extension. Three asterisks indicate that the files have no extension (as with the *Helsinki Corpus*).
- 10) Levels of tree visible at startup.
- 11) What icons for nodes? Book = 0, Folder = 1

First 11 parameters of control file for the Helsinki Corpus

```
G:\HELSINKI\TEXTS
G:\HELSINKI\MANUAL\HELSINKI.JPG
G:\HELSINKI\MANUAL\HEL_CORP.RTF
G:\HELSINKI\MANUAL\FAQS.RTF
G:\HELSINKI\MANUAL\FACT.RTF
Courier New
10
90
***
2
1
```

Sample section of control file for the Helsinki Corpus (beginning, early Old English)

EXPLANATION There are three items of information for each node of a tree. The first is the description to be used as a label for a node (plain text). The second is the file associated with this node. If you enter DUMMY.RTF here then no file is displayed. This is necessary because there will be nodes in a tree which are empty, i.e. there are just links to other nodes further down the tree. Indeed it is normal, though not essential, that only the terminal nodes of a tree contain actual file references. The third item of information usually consists of three asterisks. The reason it is there at all is that with audio files you may wish to display an image file in the background. If you now specify a WAV file as item no. 2 and an image file as item no. 3 then the latter will be displayed while the former is played. By these means you could for example display a map of a region and play an audio file with the speech of that area at the same time.

You will notice that the description of many nodes is indented. This is deliberate and represents the means by which you specify what level in a tree the node is to be displayed on. The principle is as follow: every 4 spaces at the beginning of a node label represent an indent of one level below the first, i.e. no spaces indicate a node on the top-most level (level 1), 4 spaces indicate that the node is on level 2, 8 spaces on level 3, 12 on level 4, 16 on level 5 and 20 on level 6. A maximum of 6 levels is permissible.

```
Old English
DUMMY.RTF
***
    I ( - 850)
DUMMY.RTF
***
        Documents
DUMMY.RTF
***
```

Documents 1 (Harmer, Robertson, Birch)
CODOCU1

Undefined text type (verse)
DUMMY.RTF

Caedmon's Hymn; Bede's Death Song; The Ruthwell Cross; The Leiden
Riddle
CONORTHU

II (850-950)
DUMMY.RTF

Law
DUMMY.RTF

Alfred's Introduction to Laws, Laws (Alfred), Laws (Ine)
COLAW2

Documents
DUMMY.RTF

Documents 2 (Harmer, Robertson, Sweet-Whitelock)
CODOCU2

Handbooks, medicine
DUMMY.RTF

Laeceboc
COLAECE

Philosophy
DUMMY.RTF

Alfred's Boethius
COBOETH

Religious treatises
DUMMY.RTF

Alfred's Cura Pastoralis
COCURA

Prefaces
DUMMY.RTF

Alfred's Preface to Cura Pastoralis
COPREFCP

History
DUMMY.RTF

Chronicle MS A Early
COCHROA2

Bede's Ecclesiastical History
COBEDE

Oonthere and Wulfstan (MS L)
COOHTWU2

Alfred's Orosius
COOROSIU

Tagging advice text

This text has been written in order to help users of the program *Corpus Presenter Text Tool* when tagging texts. The first thing to bear in mind is that there are two modes for tagging with this program:

- 1) *Automatic tagging* (Shift-Ctrl-F3)
- 2) *Interactive tagging* (F3)

Both tagging options are to be found in the *Tools* menu of the program. If you like, you can use the shortcuts indicated in brackets above. Before starting tagging it is strongly advisable to consider what texts you wish to tag in what manner. The procedures available here are mechanical aids. They do not make any decisions on the contents of text. Successful tagging is only possible if the user designs a sensible tagging system and the words in the texts to be tagged are unambiguous. Think first of all what your goal in tagging is. It is a time-consuming procedure, but greatly accelerated by functions such as the present two. Nonetheless, tagging is as good as you make it. Whether it yields the results you want depends on how you go about it.

1) Automatic tagging

This is the easiest form of tagging and will work best when the forms to be tagged in a text are unambiguous. For instance, if you assign a tag `_INTERROG_ADV` to a form like *which* then the results are liable to be incorrect if your text(s) contains a sentence like *The car which was stolen* where *which* is a relative pronoun. So be careful with automatic tagging if there are polyfunctional forms in your text(s).

Before automatic tagging you must write (or adapt) a list file in which the tags and the forms (words or parts of words) which are to be assigned these tags are specified. Please consult the supplied file `AUTO_TAG.LST` to see how this is done. When the Automatic tagging window opens, you can select a list of tags (you can have several on disk if you wish). You must select the rows in table with tags and forms before tagging (or deselect „Only tag subset of forms“). You then select tags in the grid which appears in the usual way, by pressing either the Shift-key or the Ctrl-key along any of the arrow keys, just as in any *Windows* program. You may also say if tags are only to be attached to whole words and you can choose to confirm each tag. The latter is useful if you wish to skip forms to which a tag does not apply. You can also specify whether tags are to be highlighted (by red marking). This is only retained if you stored the tagged file(s) to disk as an RTF file(s).

2) Interactive tagging

Tagging a text consists of attaching a grammatical label as a suffix to a word form. This is an important aspect of preparing text corpora for later linguistic retrieval tasks, either by the compiler(s) or others who have access to these corpora. However, tagging texts is time-consuming and its accuracy depends on the nature of the texts and the tagging scheme used.

With the option *Interactive tagging* the user decides what category of label is to be suffixed to what word forms. Once this operation has been carried out grammatical

information can be retrieved from the texts of a corpus by referencing the tags suffixed onto words which have been tagged. In general you cannot reference semantic information in a corpus, i.e. a tagged corpus is primarily intended for retrieving morphological and possibly syntactic information.

To tag words/forms in the current text, you first select (get from disk) a set of words/forms to fill the list *Words to tag*. You then get a list for *Tags to use*. Now select some words/forms (in the *Words to tag* list) which are to be assigned to one of the tags. The words/forms are entered in the sub-list on the left by clicking on the button *Import checked forms*. Choose a tag to be attached to each of these forms. This appears in the top-left corner. Start the tagging process by clicking on *Start*.

The maximum number of tags and of input forms is 512 items in each case. The lists can be created with *Corpus Presenter Text Tool* itself and stored to disk for later use on this level of the program. Take a look at the small file TEST_TAG.LST which is in the directory C:\Program Files\Corpus Presenter on your hard disk. This can be used for tags. Try the file TEST_INP.LST for a set of words/forms to be tagged. Use the file CHAUCER.ASC in the directory C:\Program Files\Corpus Presenter\Cp_Data\Test_Cp as a text with words/forms to be tagged. You can now experiment to see how the tagging actually works.

Tagging parameters

- 1) *Words or strings* Specifies is only words – or any string – can be tagged.
- 2) *Case-sensitive search* Determines whether small and capital letters are distinguished.
- 3) *Automatic or manual* Here you can decide whether *Corpus Presenter Text Tool* halts at each find and asks the user to confirm whether a form is to be tagged. Note that with manual tagging you can also edit the finds in the current text as you proceed.

Raymond Hickey
January 2010