Conc

A Concordance Generator

Version 1.70 beta

John Thomson

February 1992

Contents

Chapter 1

INTRODUCTION

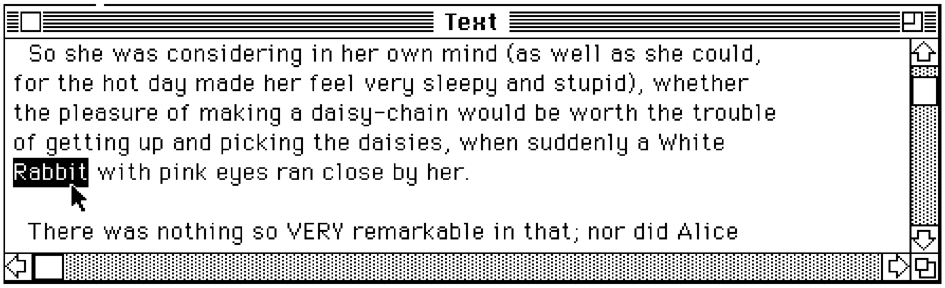
1.1 About Conc

Conc is a program designed to facilitate the intensive study of a flat text or an interlinear text by producing a list of all the words occurring in it, with a short section of the context preceding and following each occurrence of a word. In many fields of study such a list is called a concordance. It is also similar to a key word in context (kwic) index,[1] except that the index does not have to be restricted to particular words.

Conc can also produce a more conventional index, consisting of a list of the (distinct) words in a document, each with a list of the places where it occurs. It can also do some simple statistical studies of a text, such as counting the number of occurrences of words that match a "pattern" (a variation on grep[2] regular expressions).

Conc displays the original text in one window (figure 1) and the concordance of the text in another window (figure 2). If requested, the index is displayed in a third window (figure 3). By clicking a particular word in the concordance you can locate that word in the main text, thus seeing a larger context than is possible in the concordance, which is limited to one line per word. It is also possible to click in the index and thus locate the corresponding group of lines in the concordance (and the first occurrence in the text), or to click a word in the text and locate the corresponding position in the other windows.

Figure 1. The text window.

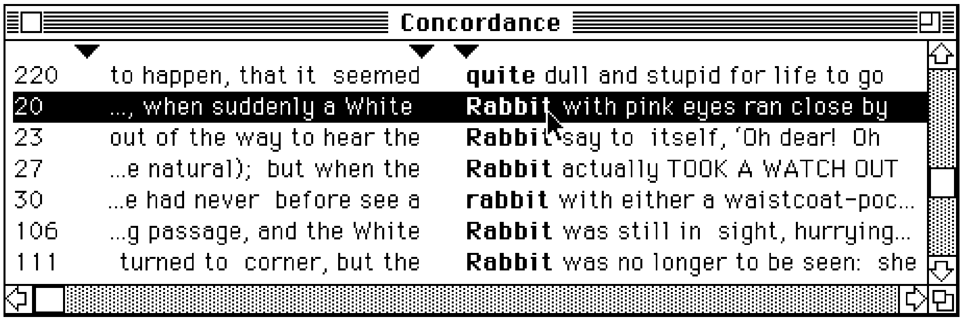


```
 So she was considering in her own mind (as well as she could,
for the hot day made her feel very sleepy and stupid), whether
the pleasure of making a daisy-chain would be worth the trouble
of getting up and picking the daisies, when suddenly a White
Rabbit with pink eyes ran close by her.

 There was nothing so VERY remarkable in that; nor did Alice
```

[1] This is the reason for the key icon

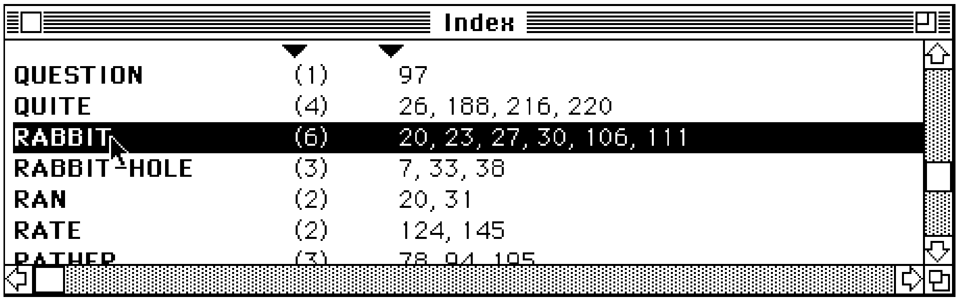
[2] Get Regular Expression and Print—a standard search tool in the Unix environment, and included in many other programs. The exact nature of a "pattern" is explained in the section on pattern matching (section 4.4).

Figure 2. The concordance window



```
220    to happen, that it  seemed    quite dull and stupid for life to go
20     ..., when suddenly a White    Rabbit with pink eyes ran close by
23     out of the way to hear the    Rabbit say to  itself, 'Oh dear!  Oh
27     ...e natural);  but when the  Rabbit actually TOOK A WATCH OUT
30     ...e had never  before see a  rabbit with either a waistcoat-poc...
106    ...g passage, and the White   Rabbit was still in  sight, hurrying...
111     turned to  corner, but the   Rabbit was no longer to be seen:  she
```

Figure 3a. The index window.



```
QUESTION       (1)     97
QUITE          (4)     26, 188, 216, 220
RABBIT         (6)     20, 23, 27, 30, 106, 111
RABBIT-HOLE    (3)     7, 33, 38
RAN            (2)     20, 31
RATE           (2)     124, 145
RATHER         (3)     78, 94, 105
```

It is not always desirable to have all the words of a text appear in the concordance and index, particularly if it is to be exported to a text file or printed (a full concordance can easily be ten times the size of the original file). Conc allows you to select the words you do want to include in the concordance, either by listing them individually or using a pattern[3] to describe them. In addition, you can choose to omit short (or long) words and words that occur very frequently (or infrequently). It is also possible to give a short list of particular words you do not want in the concordance (known as a stop list).

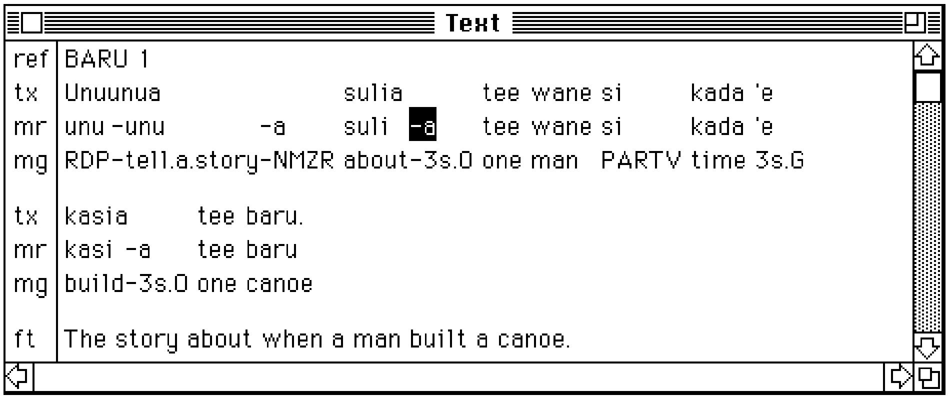
In a printed concordance, it is important not just to have the text surrounding the word of interest, but also to have some form of reference which will make it possible to locate the word in the actual text. References are also needed in the index. Conc supports several types of referencing schemes. You can use the absolute line number in the file, or put your own markers into the text. Conc supports various ways of distinguishing these markers from the main text. It can support either a simple numbering scheme or one with chapters and verses (as used for example in the Bible), or even one with multiple levels such as work, volume, page, paragraph,

[3] Again, in the style of grep, as described in section 4.4.

and line.

In addition to doing concordances of flat text files such as shown in figure 1 above, Conc can also do concordances of interlinear texts. An interlinear text consists of a base line text plus one or more lines of annotations that are vertically aligned. Interlinear texts can be constructed using the *IT* (‘eye-tee’) program described in the book *How to use IT: interlinear text processing on the Macintosh.*[4] A sample interlinear text is shown in figure 3b. (This text exemplifies only one model of analysis; many other models can be defined using *IT.)*

Figure 3b. An interlinear text

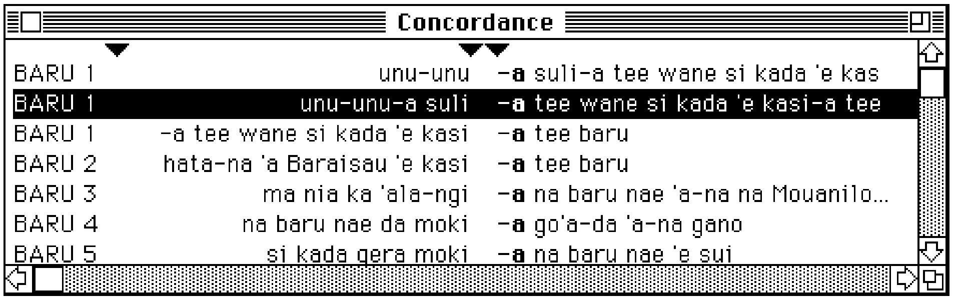


```
ref BARU 1
tx  Unuunua               sulia      tee wane si     kada 'e
mr  unu -unu       -a     suli  -a   tee wane si     kada 'e
mg  RDP-tell.a.story-NMZR about-3s.O one man  PARTV time 3s.G

tx  kasia      tee baru.
mr  kasi -a    tee baru
mg  build-3s.O one canoe

ft  The story about when a man built a canoe.
```

This interlinear text consists of a base line text (marked by *tx* in the left margin) plus two aligning fields: a line with morpheme breaks inserted (marked by *mr* for morphemic representation) and a line of morphemic glosses (marked by *mg).* There is also one nonaligning (or freeform) field for the free translation (marked by *ft).* The entire first sentence of the text is given the reference code BARU 1 (marked by *ref).* Notice that there are two levels of vertical alignment in this text: word alignment and morpheme alignment. Figure 3c shows a concordance of the interlinear text in figure 3b. It is a morpheme concordance, which means that each morpheme in the *mr* line and gloss in the *mg* line are concorded.

[4] Gary F. Simons and John Thomson, Linguist’s Software, Edmonds, Washington, 1988.

Figure 3c. A morpheme concordance



```
BARU 1                   unu-unu   -a suli-a tee wane si kada 'e kas
BARU 1              unu-unu-a suli   -a tee wane si kada 'e kasi-a tee
BARU 1    -a tee wane si kada 'e kasi   -a tee baru
BARU 2    hata-na 'a Baraisau 'e kasi   -a tee baru
BARU 3             ma nia ka 'ala-ngi   -a na baru nae 'a-na na Mouanilo...
BARU 4           na baru nae da moki   -a go'a-da 'a-na gano
BARU 5            si kada gera moki   -a na baru nae 'e sui
```

Conc understands the use of fields in an interlinear document, and can limit the words in a concordance to those from selected fields. It is also possible to have words from one field appear in the concordance, while the corresponding word in another field is used to control sorting or word inclusion.

In addition to word and morpheme concordances, Conc can also produce a concordance of all the letters (characters, phonemes) in a document. A letter concordance can be limited to just those letters that occur in a particular environment. This is useful for doing phonological analysis. Letter concordances can be done either on flat text or on interlinear text. In the case of interlinear text, the letter concordance can be restricted to selected fields.

Conc has several output facilities.

- It can save an entire concordance (including the text on which it is based) in a file for later access.
- It can export the text, concordance, or index (or selected parts of them) in a form suitable for loading into a word processor.
- It can copy selected parts of the text, concordance or index to the clipboard.
- It can print the concordance or index (or selected parts of them).

Figure 3d summarizes Conc's concordance capabilities according to type of text (flat or interlinear) and type of concordance (word, morpheme, or letter). The only combination not allowed is a morpheme concordance of a flat text.

Figure 3d. Text and concordance combinations

	Flat text	Interlinear text
Word concordance	yes	yes
Morpheme concordance	no	yes
Letter concordance	yes	yes

1.2 Using this documentation

This documentation is designed to describe completely the features of Conc. This is necessary to allow advanced users of the program to take advantage of all its features. However, many users will not need to use most of the features. It is suggested that new users first try out the program on one of the supplied concordance files, becoming familiar with the features described in chapter 2. Anyone who wants to construct a new concordance out of a flat text file should read chapter 3 on the input file requirements. After that you only need to read other sections if you are unhappy with some aspect of Conc's default behavior. It is a good idea to have Conc running on a small file (changing options on a large file can take a few minutes) and try things out as you read about them.

If you are working with an existing concordance file and don't want to change it, you can ignore the documentation, certainly after chapter 2 and possibly including it, since the use of Conc to study an existing concordance is quite intuitive. (You might also want to study the material on printing in the section on the File menu, if you do not find the dialog boxes intuitive.) The bulk of this documentation is designed to help you to create a concordance that shows the information you are interested in presented in the way you want.

If you are working with interlinear documents exclusively, you can ignore chapter 3 on specifying references in a flat text; Conc can determine all it needs to know about the structure of an interlinear document from its accompanying model.

If you received this document on disk and are not printing on a LaserWriter, you should probably redefine the Normal text style to some font and size which your printer can print well.

1.3 Limitations

Conc is *not* primarily a retrieval engine, though it has been used as such. That is, the facilities that allow you to choose what will be put in the concordance are not optimized for searching for a few things in a long document, but rather for defining a broad area of interest. Changing the specification of what you want included in the concordance requires it to be reconstructed and resorted, which can be quite time-consuming for long documents.[5] By contrast, you can locate any word in the concordance instantly by typing its first few letters. This is usually a better way to use the program, unless you are working towards a printed concordance.

Conc is limited to texts small enough to fit in memory along with Conc itself and the data structures needed for the concordance and to display the text properly. See section 1.4 on memory requirements.

Conc does not support any kind of modification to a document. It does not allow more than one document to be open at a time (though it does allow you to append documents).

[5] This can be improved somewhat if you only want to remove words, not add ones that were left out. See the section on inclusion of words in the concordance for details.

See appendix A.1 for some more detailed limitations and problems.

1.4 Installation

Conc requires no special installation. Simply copy the program (and the sample files, if you wish) onto the disk from which you intend to use the program.

If you run Conc under Multifinder, you may wish to adjust the amount of memory it is allowed to use. A rough guide is to allow 150K plus *three* times the size of the largest interlinear document you plan to work with, or *twice* the size of the largest flat text file, whichever is larger. If you plan to build concordances in which every letter is an entry, allow three times the size of the input file.

If you don't have this much memory, give Conc as much as you can. To stretch things as far as possible, you can turn Multifinder off altogether (if you are running under system 6.0 or earlier), though doing this sacrifices the ability to quickly switch to a word processor to paste things copied from Conc. Temporarily removing inits and cdevs from your System folder, and reducing the size of your RAM cache (using the control panel) will also give a little more memory. The only benefit from using a RAM cache in Conc would be to make it a little faster to "revert" to a saved version of a concordance.

A last-ditch strategy (short of buying more memory) is to make a concordance of only part of the alphabet. Depending on how small a part you choose, you could save up to about the size of your input file (more precisely, 4 bytes for every entry excluded from the concordance). See the chapter on inclusion of words in the concordance for instructions on how to do this.

1.5 Status of the program

The current version of Conc is a beta test version, not a finished product. There is no guarantee that the release version of Conc will be compatible with this beta version; therefore do not make yourself dependent on any features of this version of Conc. Be sure to use only copies of your text files when using Conc. Also, be sure to keep the original text files since there is no guarantee that later versions of Conc will be able to read concordances made with this version.

We need your feedback on this beta version of Conc. Please report any bugs or problems you find in the program or documentation (see section 1.7 below) and describe any features that you would like added or changed. If you do not tell us what you want, you will have no grounds for complaining if you do not like the release version of Conc! Send your comments about Conc to:

Academic Computing Department (Conc project)
7500 W. Camp Wisdom Road

Dallas, TX 75236
U.S.A.

phone; 214/709-3395
fax: 214/709-3387
e-mail: evan@sil.org (Evan Antworth)

1.6 Permission to use and copy

Conc and its documentation are copyrighted by John Thomson and the Summer Institute of Linguistics. Permission is hereby given for anyone to use it on the understanding that the program is provided on an “as is” basis and anyone using it does so entirely at their own risk. Permission is further given to make copies of the program and to pass them on to others, provided that (1) the program and this documentation file are copied together and neither is modified, and (2) permission is obtained from the copyright holders before copying for any commercial benefit. No payment is required for using this beta version of Conc.

1.7 How to write a bug report

When reporting bugs, please include a detailed description of exactly how to reproduce the problem, and please send a disk containing any files needed. If you have time to experiment and determine some of the circumstances in which the bug does or does not show up, that will help.

It is very difficult to list all the things one would like to know about a bug, since it is the nature of bugs that they appear in unexpected situations. If a program has supposedly been thoroughly tested, and a bug shows up, there must be something that the user is doing differently from the tester and developer, and any information that may hint as to the difference is helpful.

A starting point is the name of the program (Conc) and the version number (from its About box). It is useful to give the created and modified dates of the program (from the Finder's Get Info box) as a further check that the right version is under consideration.

Next, describe the environment in which the program is running, especially anything that might be unusual. Give the System software version number (Get Info of the System file), state whether Multifinder was running, list any other programs running under Multifinder, any DA's that were active, and any inits and cdevs in your system folder. Specify the model of Macintosh, how much RAM, any special cards installed, how many drives, their capacities, and their manufacturers. Describe anything unusual about your display system and any peripherals that are attached. If your disk is full or any other unusual circumstance applies, mention it.

If you have access to a virus detection program (many are available, both commercially and as freeware) please use it to check that the problem is not caused by a virus that has infected your system.

Then, describe the problem in as much detail as possible. State what you did, what you expected to happen, and what actually happened. Indicate whether you have been able to reproduce the problem or whether it only happened once. (There is not much we can do about one-of-a-kind problems, so such reports are likely to get filed for attention when something else clarifies the problem.) If the problem only happens in one situation, be very precise (e.g. “I had opened a file which the program says had 956 words. I copied one word from the text window to the clipboard and pasted it into ACTA. It appeared twice.”) Actually, you can’t be too precise; programming being what it is, it could well turn out that it only happens when you copy the first word on a line, or only when you copy an unusually long word, etc. In fact, if the problem doesn’t seem to occur with a variety of files, it is a good idea to send a copy of the files with the bug report; then you can say (for example) “Open the options file file *xyz* and then the data file *abc*. Click the first word in the second line of the text window. Press command C. Open the ACTA desk accessory. Choose Paste from the Edit menu. The word should appear twice."

If you want your programmer to really love you (and especially if you want fast action and don’t plan to send a sample file) try as many variations as you can think of and report which ones give errors—for example, “Copying anything to the clipboard seems to leave two copies of it there. It doesn’t matter whether you copy one word or many, and the source material can come from any of the main windows in Conc; but if you copy text from a dialog box you only get one copy. It doesn’t matter whether you paste into a desk accessory or another program, but if you are running MultiFinder the problem does not occur.”[6]

[6] This problem actually occurred in an earlier version of Conc. It did not show up during my testing because I always use Multifinder.

Chapter

2

USING CONC ON AN EXISTING CONCORDANCE

2.1 The main windows

To get a quick feel for what Conc can do, double-click the file named *Alice1.conc* provided with the program. This will run Conc and load a concordance of the first chapter of Lewis Carroll's *Alice's Adventures in Wonderland.* You should see two windows, as shown in figures 1 and 2 above. These windows can be moved, resized, zoomed, scrolled, and closed in the usual Macintosh way. In addition, the Windows menu provides commands for reopening closed windows and and a command to tile the windows.

The concordance (see figure 2) is arranged in three columns. The left-hand edge of the right-most column is an alphabetical list of the words in the text. Following each word there is a little bit of what follows it in the text. In the middle column is a bit of what precedes the word in the text. In the left-most column there is a reference indicating where in the text the word appears. You can adjust the width of these columns by dragging the tab controls at the top of the window.

The reference is mainly for use if the concordance is printed out. To see a word in its full context, just click the line in the concordance where the word appears. Conc automatically scrolls the text window as necessary to show the particular occurrence of the word you have selected, and highlights the word itself. Similarly, you can click something in the text window to scroll the concordance window to see the other occurrences of that word. (If you click on a word that you told Conc to omit from the concordance, the concordance window does not scroll, and nothing is selected in it.)

If you type a letter on the keyboard, Conc locates the first word in the concordance that starts with the letter you type (or the next greater word, if nothing starts with the letter you type).[7] The text and index windows scroll to display the selected word. If you type several letters rapidly, Conc looks for the first word that starts with the string you typed. A short pause (equal to your current double-click interval) will allow you to start again, typing a new string. Notice that Conc will always search the concordance window, regardless of which window (text, concordance, or index) is active.

To create an index of the current concordance, use the Build menu and choose the Index command. You will see an index window that shows each distinct word just once, along with the number of times it occurs and a list

[7] This does not work if you specify that the concordance should not be sorted

of references.[8] Click in the index window to select a word there. All occurrences of that word in the concordance window will be selected, and so will the first

[8] This list may be incomplete; see section 7.1 on the index window for details.

occurrence in the text window. See section 7.1 on the Index Options dialog box for information on the options that control the material in this window.

You can use the Font menu to alter the font and size of text in any of the windows. The concordance and index windows use one font for everything; so does the text window if you are working with a flat text file. However, with an interlinear file, you can set the font of each field.

2.2 Adjusting the layout of the windows

By default Conc wraps long lines and bundles to fit the original size of your window. If you later resize the window or make some other change (such as a font change) which results in the right edge of the text not aligning well with the window edge, the command Set Wrap Length” on the Layout menu will reform the text to fit in the window.[9] If some kinds of data should not be wrapped, you can tell Conc this using the Wrap Long Lines dialog box discussed at the end of this section. Of course long lines can be viewed by scrolling horizontally.

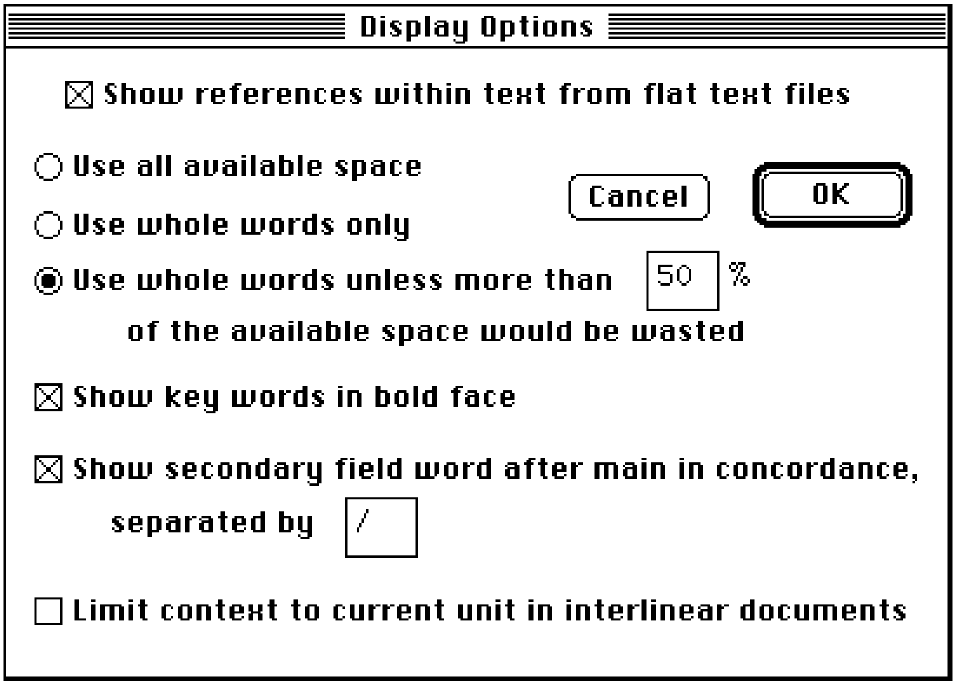
You can adjust the amount of preceding or following context that is displayed by moving the tab arrows at the top of the concordance window. To move the tab arrows, click and drag them. There is a small minimum gap between reference and preceding context. The index window has two arrows which control the space allowed for words, the number of occurrences, and the list of references.

The total width available for context is determined by the shape of the paper you are using (see the Page Setup dialog box) and the margins you have selected (see the Page Layout dialog box). To maximize the amount of context you can see by scrolling, choose a sideways page layout and zero margins.

The exact way context information is displayed is controlled by the dialog box brought up by the Display command on the Layout menu. This is shown in figure 4.

[9] This is not done automatically for every change that might possibly affect it, because it takes a little while to rearrange the line breaks for a long file.

Figure 4. The Display Options dialog box.



☒ Show references within text from flat text files

○ Use all available space

○ Use whole words only

● Use whole words unless more than 50 %
of the available space would be wasted

☒ Show key words in bold face

☒ Show secondary field word after main in concordance,
separated by /

□ Limit context to current unit in interlinear documents

The first check box, “Show references within text from flat text files,” allows you to control whether references are displayed in the text window.[10] This is usually desirable, since they are part of the text, but if you have artificially added references to a text that is clearer without them, you can turn this option off. Turning if off also prevents references showing in the context text in the concordance window, which allows you to see a little more of the real context.

Generally you want to see whole words in the context part of the concordance unless a word is so long you would waste most of the line. This is the default option; if more than half the available space would be wasted by using whole words, the program uses all the space and puts three dots to show a word is incomplete.

You can change the threshold of wastage to anything you like. Particularly useful is a very low or even zero percentage, which will use all the space; you get a lot of partial words with three dots.

The other two options in this group are simply never to show less than a whole word (this gives the least information but is most readable) and to

[10] This option has no effect on interlinear documents, where the reference is considered always relevant.

use all the available space, which gives you the greatest possible amount of context but with no concessions to readability. When using all available space, Conc makes sure you get whole letters, but does not use dots when it breaks a word. If you want to use as much space as possible but have Conc indicate where it has broken words, use the percentage option with zero as the numerical value.

Note that the right hand limit is not the edge of the window, but the edge of the sort of paper you are using, after adjusting for the margins specified in the Page Layout dialog box. Use the horizontal scroll bar if necessary to see more context on the right. If there is no printer driver available the width of the window is used instead. Also note that even with "Use all available space" you will get some white space on a few lines. The first few words of your source file have little preceding context!

The second check box in the window, "Show key words in bold face," allows you to specify that the key word in each line of the concordance (i.e., the first word in the right column) should be in bold to distinguish it from the context. This is the default; turn it off if you want keywords to look the same as the rest of the text.

The third check box, "Show secondary field word after main in concordance," is applicable only to interlinear documents when you are using the "primary and secondary fields" option described in chapter 6. It tells Conc to show the related word from the secondary field in the concordance and also tells how to mark it off from the main word. By default it is on.

The final check box, "Limit context to current unit in interlinear documents," is useful when material in adjacent units is unrelated, and context from preceding or following units is therefore not useful. Turn it on to prevent Conc showing such context in the concordance window.

The appearance of the text window can be controlled by choosing options found in the Wrap Long Lines dialog box on the Layout menu (see figure 5). Most users can just leave all of these options turned on, which will cause Conc to automatically arrange all long lines to fit the window.[11] Also, Conc assumes that adjacent bundles and lines in nonaligning fields in an interlinear document are simply continuations of the first bundle or line, and so treats them as a single, continuous stream of columns or words.

There are two circumstances in which you might want to change the way Conc wraps long lines. If you have formatted your text in some special

[11] Because it takes a while to recompute all the line breaks, Conc does not automatically adjust the line lengths when you resize the window. If you want the line lengths adjusted, issue the "Set wrap length" command in the Layout menu.

way which is confused by continuation lines, you may prefer to scroll horizontally to see long lines. And if some of your text is in a right-to-left script, Conc’s attempts to rearrange it assuming the normal left-to-right style could be very harmful.

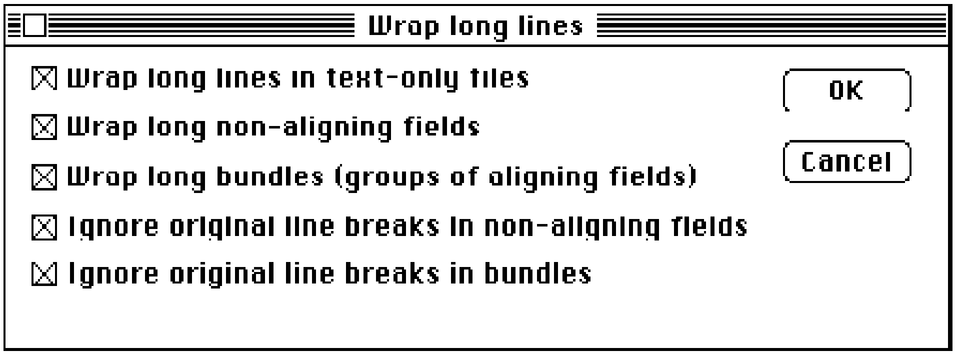
The two most common boxes to turn off would therefore be the "Wrap long bundles" and "Ignore original line breaks in bundles," both of which should be turned off if the base text you are annotating is right-to-left. Similarly, turn off "Wrap long lines in flat text files" if you are building a concordance out of a right-to-left text file or one with complex line layout.

Noticing line breaks and avoiding wrapping in freeform fields is not very useful with the current version of *IT*, which always ignores line breaks in such fields, but may be useful in a later version or with an interlinear document produced by some other means.

In the unusual case that you are using breaks between bundles to signal paragraphs, you might want to allow wrapping but not ignore the breaks.[12]

The Interlinear Layout option allows you to control many aspects of the appearance of your interlinear text. While not a substitute for the sophisticated features of the *Interlinear Text Formatter*,[13] these options are likely to be sufficient for obtaining a pleasing layout for small examples to be copied into another program.

Figure 5. The Wrap long lines dialog box.



☒ Wrap long lines in text-only files

☒ Wrap long non-aligning fields

☒ Wrap long bundles (groups of aligning fields)

☒ Ignore original line breaks in non-aligning fields

☒ Ignore original line breaks in bundles

2.3 Using the clipboard

The selected text in any window can be copied to the clipboard. To make a selection of more than a single word or line, you can either drag (up or down) or click to set the start of the selection and shift-click to set the end. Only the first word clicked will be selected in the other windows. If you drag above or below the window it will scroll automatically. Automatic

[12] This is unlikely enough to make it tempting to merge the "wrap" options with the "ignore breaks" options. What do you think?

[13] See *Formatting Interlinear Text,* by Jonathan Kew and Stephen McConnel, SIL, 1990.

scrolling is supported only for vertical movement; use manual scrolling and shift-click to make a wide horizontal selection.

Cut and paste are not available in any of the main windows because Conc does not support editing either the original text or the concordance. These menu options are provided for use in dialog boxes and desk accessories.

When you are working with an interlinear document, there are two options for copying. The reason for this is that it is difficult to satisfactorily copy interlinear text to another program, since other programs don’t understand the relationship between a word and its annotations.

For some programs, it is important to preserve the exact appearance of the text as Conc displays it. This can be achieved by copying the data as a *picture*, which can be done using the Copy Picture command on the Edit menu. For other programs, it is more important to preserve the textual nature of the data, and the normal Copy command should be used. Here are some of the tradeoffs:

• You can paste a picture into almost any program and it will look right. The alignment of the text, its spacing, its fonts and its styles will all be preserved. You will also get accurate spacing on a postscript printer. On the other hand, when you copy text, alignment will generally be inexact on the screen, and always on a postscript printer. Just how bad the alignment will be depends on how extensively you have varied fonts and styles and whether your word processor supports copying styled information from another application.

• But, in most word processors you cannot edit a picture, it cannot be split over page boundaries, and you cannot search for text inside it.

In each case there are some steps you can take to mitigate the problem. You can edit a picture using a graphics program or a page-layout program that understands graphics; this would allow, for example, a word to be highlighted or underlined, or even arrows to be drawn to items of interest. If it is important to be able to search, you might consider copying the text *both* ways, and making the text version invisible for printing (if your word processor supports this).

You may be able to improve the appearance of interlinear data copied as text by specifying a “narrow space” character, if you have a font installed that has one. Ideally, you want a white space character that is just one pixel wide at a size of 12 points. You can specify such a character (and the font it requires) in the Interlinear Layout dialog box on the Layout menu. This only works if the receiving program understands styled data in the clipboard from another application.[14]

[14] Microsoft Word version 4, for example, can copy text with styles internally but ignores styles on text copied from another application.

A final resort is to set everything in the interlinear document to a monospace font such as Monaco or Courier. This can be copied as text and will preserve alignment even when pasted into applications that don't understand styled data (you may have to select the font in the receiving application, too). It will also preserve alignment on a postscript printer. However, monospaced fonts are not attractive, and may not be feasible at all for some languages.

2.4 Saving, closing, opening, and printing

The complete current concordance, with all options as you have set them, including both the text and the sorted concordance, can be saved to a file using the Save or Save As commands on the File menu. Such a file can be opened using the Open command, or by double-clicking its icon on the desktop. Opening a new document will close any previously open document; Conc does not support having more than one document open at a time (though see below on the Append command).

To create a new concordance, first use the Open command on the File menu to open a text file. Choose appropriate options by using the commands on the Options menu. Then choose a concordance command on the Build menu.

The Append command on the File menu allows you to open additional text files and append them to the end of the current text. If you have both a text window and a concordance window open, Conc will close the concordance before appending another text file.

You can also create a text version of a text, concordance, or index (or the selected part of one) using the export commands on the File menu. Depending on which window is active, the first export command on the File menu will say Export Text As…, Export Concordance As…, or Export Index As…. The Export Concordance As… command will change to Export *filename* As… if the concordance has already been saved. The Export Selection… command will export the selected region of the active window. See the discussion of the File menu commands in section 8.1 for a description of the format of the files produced by the export commands.

Warning Files produced by exporting a complete concordance can be very large, perhaps ten times the size of the original file. This can take a while, but you can halt the process by clicking the Abort button on progress indicator.

Similarly, the commands Print and Print Selection on the File menu print the current concordance or index, depending on which window is active (printing the text window is currently not supported). The Page Setup command is the standard Macintosh dialog box for your printer, while Page Layout and Headers allow you to control the layout of the page, as described in section 8.1 on the File menu.

The Revert command on the File menu returns the concordance and options to the way they were when the concordance was last built. This is particularly useful when you are trying a variety of ways of selecting interesting items for inclusion in the concordance; it is much faster to Revert and then to restrict the concordance again than to start from the beginning, selecting words from the original text and sorting them. Section 4.6 on progressively restricting word inclusion provides more details on this.

Chapter

3

SETTING OPTIONS FOR A FLAT TEXT CONCORDANCE

3.1 Introduction

This chapter describes how to prepare a flat (noninterlinear) text for concording. It discusses how to format a text file so that Conc will be able to report the location in the text of each word of the concordance or index. These text locations are called *references.* This chapter also describes how to declare what characters are used as word separators.

If you are basing your concordance on an interlinear text, there is nothing to interest you in this chapter, since Conc understands the reference structure of interlinear texts produced by the *IT* program.[15]

A flat (noninterlinear) text file suitable for use with Conc can easily be produced from text created with word processors such as MacWrite or Microsoft Word. However, the text must be saved without the special formatting information used by most word processors. Simply choose Save As on the File menu of the particular word processor and click the "Text only" or similar button.

WARNING Do not save the text-only version of your document to the name you have previously used for the document itself. Text-only files have no formatting information, such as paragraph spacing and font and style choices. You would lose any work you have done laying out your document.

Where there is a choice, you will generally want to put carriage returns at the end of paragraphs, rather than at the end of each line (different word processors express this option in different ways; some may not offer it). Conc normally wraps long lines, so you can get paragraphs rearranged to fit neatly in whatever size window you are using.

3.2 Line numbers as default references

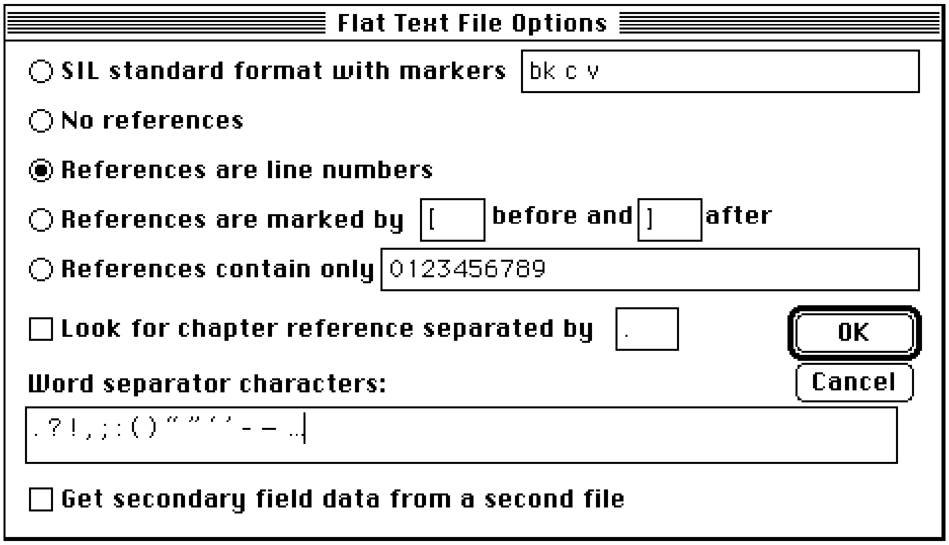
Conc's default referencing scheme is to use the line number where the key word is found in the text. You do not need to insert explicit line numbers into your text; Conc does the line numbering automatically. Note that Conc counts all lines, even blank ones. If you don't want blank lines counted, you can produce a special version of your document just for the concordance, with blank lines deleted. Or, of course, you can put in explicit numbers using one of the schemes described below.

[15] See *How to use IT: interlinear text processing on the Macintosh, by* Gary F. Simons and John Thomson, Linguist's software, Edmonds, Washington, 1988.

3.3 User-defined references

The Text Properties command on the Options menu brings up a dialog box that allows you to change the way Conc handles references. This dialog box is shown in figure 6.

Figure 6. The Text Properties dialog box



Flat Text File Options

○ SIL standard format with markers bk c v

○ No references

● References are line numbers

○ References are marked by [before and] after

○ References contain only 0123456789

□ Look for chapter reference separated by .

Word separator characters:

. ? ! , ; : ( ) “ ” ‘ ’ - – …

□ Get secondary field data from a second file

The default choice shown above is “References are line numbers.” In addition, there are several other referencing schemes available.

- No references shown in either the concordance or index.
- User-defined references (typically numbers) with option of two levels of referencing (such as chapter and verse).
- SIL standard format markers.

To omit all reference information from the concordance and index, click the button for “No reference.”

To define a referencing scheme using your own reference markers, you have two choices. First, you can click on the radio button “References contain only…”. By default this choice will treat all numbers as reference markers. If you also check the box “Look for chapter reference separated by…”, it wil look for two levels of reference markers, such as chapter and verse, with the chapter number separated from the verse by a period (no spaces). You can change separator character from a period to any non-word forming character of your choice. For example, given the text

1.1 This is verse one, chapter one. 2 This is the second verse. 3 And the third. 2.1 Now we come to chapter two.

Conc would put the reference 1.1 in front of each of the words in the first sentence, 1.2 in front of the words of the second, 1.3 for the third, and 2.1 in front of the words of the last sentence. Note that putting a space after the dot will confuse Conc, causing it to miss the chapter numbers. The set of characters used to mark references can be changed by editing the field in the "References contain only…" button. For example, you might have a file of numbers and want use alphabetic references, or want to use a letter such as * as part of a reference. It is important to remember that this referencing scheme will work correctly *only* if the characters that form reference markers do not occur elsewhere in your text.

The second way to define your own referencing scheme is to click on the button "References are marked by…before and …after." If you cannot reserve a set of characters (such as numbers) to use for references, you can reserve a pair of brackets to delimit references. By default Conc uses square brackets ([..]) if you choose this option, but you can choose any two characters you like provided that they are not used in the text. It is recommended that you choose two different characters, however. By using brackets, the characters inside the brackets can be any characters at all, even word-forming characters. The example above could be set out like this:

> [1.1] This is verse one, chapter one. [2] This is the second verse. [3] And the third. [2.1] Now we come to chapter two.

Now it does not matter if numerals occur in the text, since Conc knows that only numerals surrounded by brackets constitute rererence markers. Also, word formation characters can be used in the references, for instance [1.1a], [1.1b].

3.4 SIL standard format markers as references

Yet another possible referencing scheme is to use "SIL standard format." This format uses "markers" (which are character strings preceded by a backslash (\) and followed by a space or end of line) to indicate the structure of a text. Conc allows up to nine different markers indicating different levels of reference. Conc expects each marker to be followed by a word (white space to white space) giving the value of the reference. For example, if you specified standard format markers as "c v" for chapter and verse, the above example could be changed to this:

> \c 1 \v 1 This is verse one, chapter one. \v 2 This is the second verse. \v 3 And the third. \c 2 \v 1 Now we come to chapter two.

Standard format can be used to set up very complex reference structures. In principle you could tell Conc that each reference is made up of book, chapter, page, paragraph, and line by specifying five markers in the references dialog box and inserting the markers with appropriate values into the text. (Conc actually supports up to nine-part references.) Such schemes tend to produce inconveniently long references, however.

3.5 Declaring word separation characters

Before you build a concordance of a flat text, you must distinguish two categories of characters that occur in your text: *word formation characters* and *word separation characters.* This topic is treated in more detail in section 5.2. For a flat text, all word separation characters must be declared in the Text Properties dialog box. Figure 6 above shows the "Word separator characters" text box. This box contains all characters (except white space characters such as space and tab) used as word separators. Simply edit the list of characters to reflect the structure of your text.

Chapter

4

CONTROLLING WORD SELECTION

4.1 Introduction

On the Options menu there are two options which control which words are included in the concordance. By default the concordance includes all words in the text. (Conc considers a word to be all the characters between non-word formation characters[16]). The first command, Include Words, allows you to give a basic description of the words of interest. The second command, Omit Words, allows you to specify that certain words from among those selected by the Include Words specifications should be omitted. Thus for example you could specify that Conc should include all words that begin with *a* except those that have less than four letters.

A further option, Interlinear Fields, allows you to control which fields of an interlinear document should be included in the concordance.

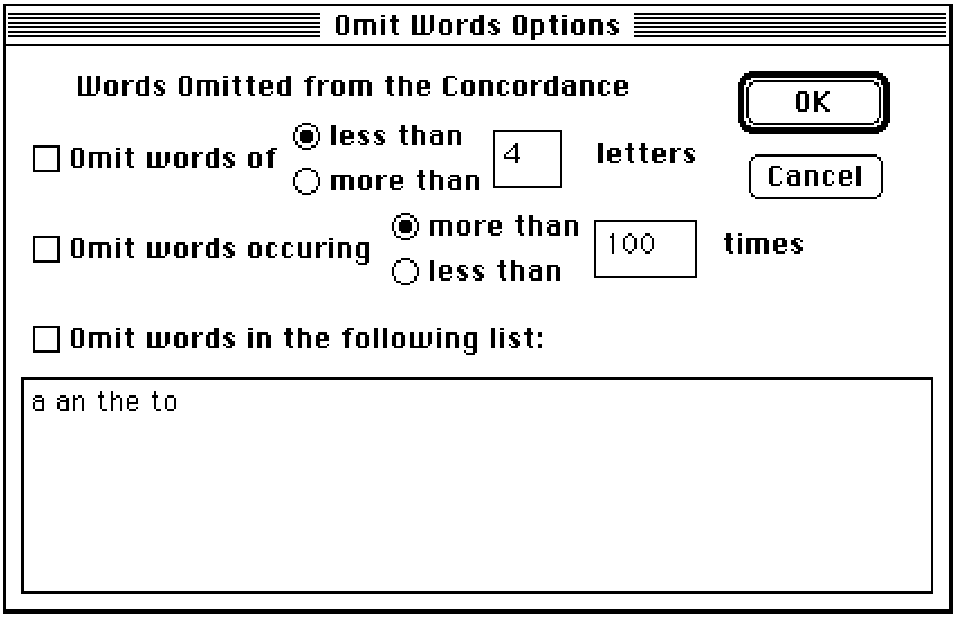
For expository purposes, we will first discuss omitting words, then including words.

4.2 Specifying words to omit

The Omit Words dialog box, shown in figure 7, is self-explanatory. One check box allows you to type a list of particular words to be omitted. Another allows you to specify that short or long words will be omitted, and yet another that words occurring very often or very seldom will be omitted. (A particularly useful application of this is to list all the words that occur only once, and check their spelling.)

[16] See section 5.2.

Figure 7. The Word Omission dialog box.



Omit Words Options

Words Omitted from the Concordance

□ Omit words of ● less than ○ more than 4 letters

□ Omit words occuring ● more than ○ less than 100 times

□ Omit words in the following list:

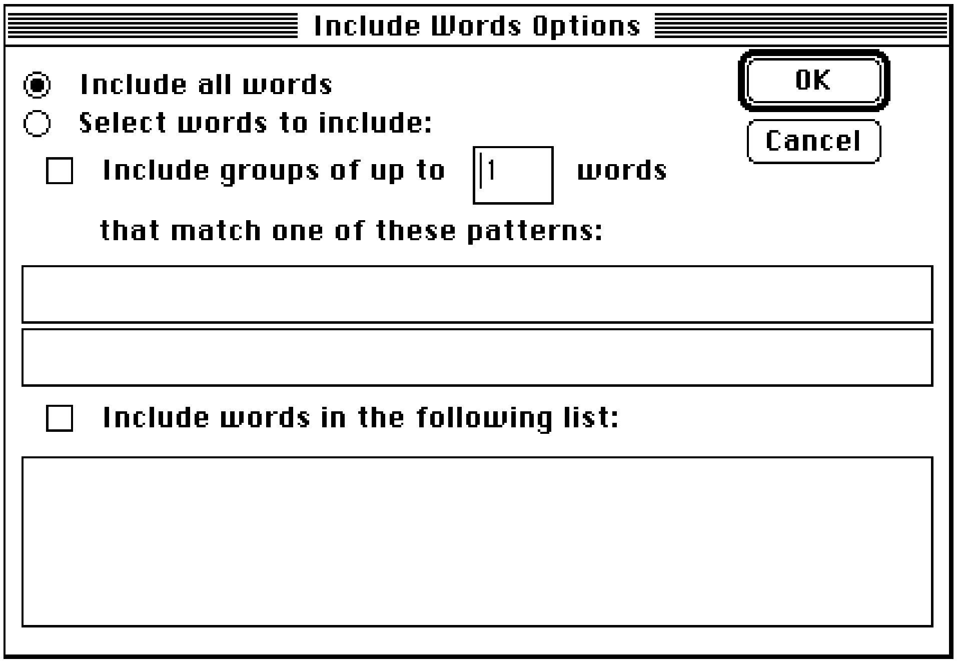
a an the to

Each way of omitting words has a check box associated with it; be sure to turn on the check box for whichever options you want to use. By default they are all off, and no words will be omitted. This does not necessarily mean you get all the words in the document, because the Omit Words conditions are applied in addition to the conditions in the Include Words dialog box. Thus the Include Words dialog box specifies in general what words are of interest, and the Omit Words dialog box allows you to refine this by omitting some of the words selected by the Include Words dialog box.

4.3 Specifying words to include

The Include Words dialog box is a little more complex. It is designed to allow you to specify generally all the words you are interested in. The default is to include all words. Another option is to list the words you want individually. Yet another way to specify the words you want is by means of a “pattern” or “regular expression” as it is more formally called. This is a rather advanced way of describing the words you want to see. The idea is that the pattern tells Conc what the words you want “look like” so it can distinguish them from the rest of the words in the document. It is usual to say that the words of interest “match” the pattern.

Figure 8. The word inclusion dialog box.



● Include all words

○ Select words to include:

□ Include groups of up to 1 words

that match one of these patterns:

□ Include words in the following list:

The program allows for two regular expressions; words that match either will be included.[17] If either is blank it matches nothing. Normally, a single word is matched against the patterns. However, there may be cases where you want to look for certain word sequences. In this case you can specify how many words to include in the comparison. Note that using a large number may make building the concordance considerably slower!

Again, there are check boxes which you should turn on as appropriate to indicate whether you want to include words listed, words matched by the patterns, or both.

Note that the effect of the word inclusion dialog box may be modified if the secondary field option is in use (see chapter 6). Note also that words specified here will be included only if they are not also specified by the Omit Words dialog box.

4.4 Pattern matching

The simplest pattern is just a group of ordinary letters. Such a pattern matches all words which include that sequence of letters. For example the pattern *ll* matches all words containing a double letter *l*, and the pattern *ing* matches all words containing those three letters in that order.

[17] In other words, the two patterns form a Boolean OR expression.

Conc also recognizes certain letters as having special meanings; these make it much easier to describe certain sets of words. It is worth knowing what the special characters are, even if you do not plan to use them, since if you use one by mistake it could have unexpected effects.

The Pattern Matching dialog box (shown in figure 9 in the next section) allows you to specify which characters have which special meanings; the description here refers to the standard settings, which with a couple of extensions correspond to the ones used in the Unix operating system and utilities.

Here is the definition of how a pattern works, using the standard character settings:

1. Any character except a special character matches itself. Special characters are \ *[* . _ # and sometimes ^ * *$*.
2. A . (period) matches any character, _ (underline) matches any character considered to be part of a word (see section 5.2 on word formation characters), and # matches any character that is *not* part of a word. For example, the pattern *a_d* will match an *a* followed by any letter followed by a *d*. The pattern *a.d* will match the same, except that if you are searching groups of more than one word, the *a.d* will allow the *a* to be in one word and the *d* in the next; *a#d* will find places where a word ending in *a* is followed by one starting with *d*, and where they are only one letter apart.
3. A \ (backslash) followed by any character except a digit or the parenthesis characters *(* or *)* matches the character that follows the backslash. This is useful if you want to look for characters that are normally special. For example, *e*\. will find words containing the sequence *e.* (provided that a period is a character that occurs in words, or you are including non-word characters in the comparison (see below)). *Note:* on the Macintosh screen, and on dot matrix printers in draft mode, the italics used for examples in this section make the backslash look like a forward slash. Be sure to use a backslash (top to the left of the bottom) when you are trying the examples or creating your own patterns.
4. A nonempty string of characters surrounded by square brackets matches any character included in the string. For instance, the pattern [aeiou] will match any one of the characters *a, e, i, o,* or *u.* In such a character string, the backslash character \ has no special meaning, and the closing bracket character *]* may only appear as the first letter. A string of characters can be negated using the symbol ^. Thus the pattern [^aeiou] will match any character except *a, e, i, o,* or *u.*

A substring *a-b*, with *a* and *b* in ascending ASCII order, stands for the inclusive range of ASCII characters.[18] For example, the pattern *n[aeiou]* will match any words where the letter *n* is followed by a vowel; but the pattern *[a–m][aeiou]* finds words where any letter between *a* and *m* (in the ASCII order) is followed by a vowel.

The pattern *[aeiou][aeiou]* will match pairs of vowels (not necessarily the same vowel). The pattern *[^aeiou][^aeiou][^aeiou]* will match words with at least three consecutive letters that are not vowels (because of the ^ (caret) meaning *not*). Of course, if you are working in a language where other letters signify vowels, you would need to list those instead.

5. A regular expression of form 1-4 or 7 followed by * matches a sequence of 0 or more matches of the regular expression. If followed by % it matches one or more. For example a_*e matches any word that contains an *a* followed by an *e* irrespective of what comes between, including words where there is nothing between. b[aeiou]*b matches words where a *b* is followed by any number of vowels and another *b*. *b[aeiou]%b* is the same except that there must be at least one vowel; it will not find words with a double *b*.

6. A regular expression, *x*, of form 1-8, bracketed, *\(x\)* matches what *x* matches. Thus for example *\(ll\)* matches words with a double *l*, just like *ll*. This is useful only in conjunction with the next special case.

7. A \ (backslash) followed by a digit *n* matches a copy of the string that the bracketed regular expression beginning with the nth \(matched. (It is very rare to use more than one *\(\)* pair). This is mainly useful when what is inside the brackets could match several things (it includes other special characters) and you later want to check for a repeat of the thing that matched. For example, \(_\)\1 matches (believe it or not!) any word that has a double letter. To read it, first consider that the underline matches any letter. Hence \(_\) (underline enclosed in parentheses preceded by backslashes) also matches any letter; the parentheses just make Conc remember which letter it was. Finally the *\1* requires there to be a repeat of whatever was in the first pair of brackets; in this case, another occurrence of the same letter. Further examples: \(__\)\1 matches any word where a sequence of two characters is repeated. (There are

[18] Unfortunately, at present this means exactly what it says. It would be much better to use the user-defined collating sequence. This may be done in a future release.

two underline characters in the pattern.) \(__\)_*\1 is the same except that there may be other characters between the pair that repeat.

Note that the three underlines don't all have to match the same character.

8. A regular expression of the form 1-8, *x*, followed by a regular expression of form 1-7, *y* matches a match for *x* followed by a match for *y*, with the *x* match being as long as possible while still permitting a *y* match. (This is just a formal way of saying what we have been assuming all along, that a pattern is a sequence of things, each of which must match something in the word. It is included only for completeness, since it rarely matters whether Conc matches the longest possible string or not.)

9. A regular expression of form 1-8 preceded by ^ is restricted to matches at the beginning of a word (or group of words, if that option is selected). For example, the pattern *^a* matches all words that start with the letter *a*.

   A regular expression of form 1-8 followed by *$* is restricted to matches at the end of a word (or group of words, if that option is selected). For example, the pattern *ing$* matches words that end in *ing*.

   The pattern ^a_*p$ matches words that start with *a* and end with *p*; the _* matches anything that may happen to come between.

10. A regular expression of form 1-9 picks out the longest among the leftmost matches in a group of words, while still permitting the rest of the pattern to match. Again this is part of the definition of pattern matching, but it is hard to think of a situation where it would make a difference in this program.

Here are some more examples. Most users will rarely need to use patterns this complex.

^[Aa] matches words starting with a or A.[19]

^[a-f] matches words starting with letters between a and f.

^\([aeiou]\)_*\1 matches words that start with a vowel provided the vowel also occurs elsewhere in the word.

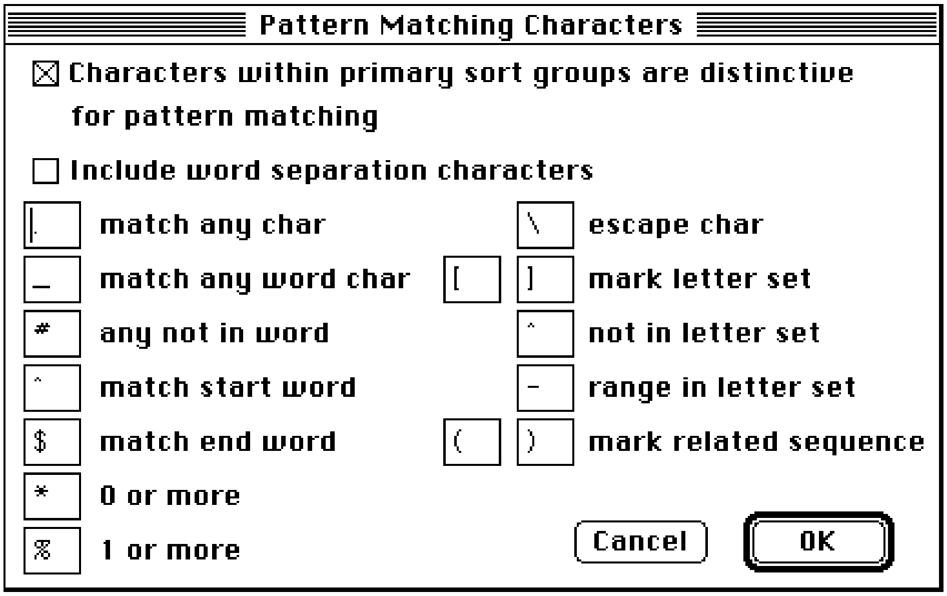
_[.;:,] matches any word followed by the specified punctuation characters. This is only useful with the "Include non-word chars" option described in the next section, or if these letters have been said to occur in words.

[19] However, there is an easier way to do this if you have specified that *a* and *A* are equal for sorting purposes: just use *^a*.

4.5 The Pattern Matching dialog box

The Pattern Matching command on the Options menu brings up the dialog box shown in figure 9, which allows you to control how pattern matching works. It also functions as a quick reminder of the special characters described above. Most of the dialog box simply contains a list of the special characters, each labelled to indicate its special meaning. You can edit the list if you find some other character easier to remember (or more convenient because the standard one is something you frequently want to search for). You can also effectively disable an option you don't use by placing there some character that never occurs in your document.

Figure 9. The Pattern Characters dialog box



Pattern Matching Characters

☒ Characters within primary sort groups are distinctive for pattern matching

☐ Include word separation characters

.	match any char	\	escape char
_	match any word char	[ ]	mark letter set
#	any not in word	^	not in letter set
^	match start word	-	range in letter set
$	match end word	( )	mark related sequence
*	0 or more		
%	1 or more		

The box "Characters within primary sort groups are distinctive for pattern matching" controls how Conc tests characters for equality. It is similar to the check box "Characters within primary sort groups are distinctive" which is located in the Sorting dialog box (see section 5.3 on specifying sorting order). If you turn it on (by checking the box), two letters must be exactly the same to be considered equal. If you turn it off (by unhecking the box), letters are considered equal if they are both members of the same sort group. Thus for example, with the option off, *a* will match either *a* or *A*; *[aeiou]* will match any vowel, upper or lower case; and ^\\(_\\)\\1 will match any word starting with a double letter, whether or not the first character is upper case. It should be understood that the "Characters…are distinctive" button in the Pattern Matching dialog and the corresponding button in the Sorting dialog, though similar, function independently; that is, you can have one button on and the other one off.

If certain characters are specified as of secondary interest only for sorting, they are ignored for pattern matching if "Characters…are distinctive” is off.

The box “Include word separation characters” controls whether characters that are not considered to be in words (see section 5.2) are included in the match. For example, with this option on, if you selected words beginning with *a* (using a pattern ^*a*) you would miss a word that began a quotation, since such a word would start with a quote (as in “a quote”). This is why Conc normally excludes punctuation. But if you want to find all the words that begin quotations (using for example ^*['"]* as your pattern) it would be annoying to have none of them found because Conc leaves out such punctuation characters. This box allows you to specify which should be done.

4.6 Progressively restricting word inclusion

If you construct a concordance and then change something in the Include Words or Omit Words dialog boxes, without changing any other options, Conc will ask “Do you want to generate a new concordance based on all the words in the text or only on the words in the current concordance?” If you choose “Original text” you will finish up with a concordance that includes exactly the words you specified from the original text.

If you choose to construct the new concordance starting with the words of the current concordance, Conc just checks each of these words to see whether it matches the new criteria and throws it away if not. For large documents, this is *much* faster than completely reconstructing the concordance. The only difficulty is that, if there are words in the text that were omitted from the concordance by earlier word selection criteria, but which should now be included, they will not be found.

This fact can actually be useful. It allows you to start with a complete list, and converge on the words of interest by progressively eliminating more. For example, suppose you want all the words that contain *a* and *e* and *i*, in any order. There is no single pattern match that will do this. However, if you first limit the concordance to words containing *a*, then to words containing *e*, and finally to words containing *i*, basing the new concordance on the previous one each time, you can easily get the required result.

A particularly valuable way to use the “Current concordance” option is to begin your study of a text by creating a concordance containing all the words you ever expect to be interested in (possibly all the words in the document). Then save this as a complete concordance, using the Save command on the File menu. Then, for each set of words of interest, open this file, change the set of words to include, and choose “Current concordance.” Use the Revert command on the File menu when you want to start again. In most cases (especially for large files and if you have a hard disk) the combination of Revert followed by building a new concordance based on the current one will be much faster than building a new concordance based on the original text.

4.7 Saving memory by producing partial concordances

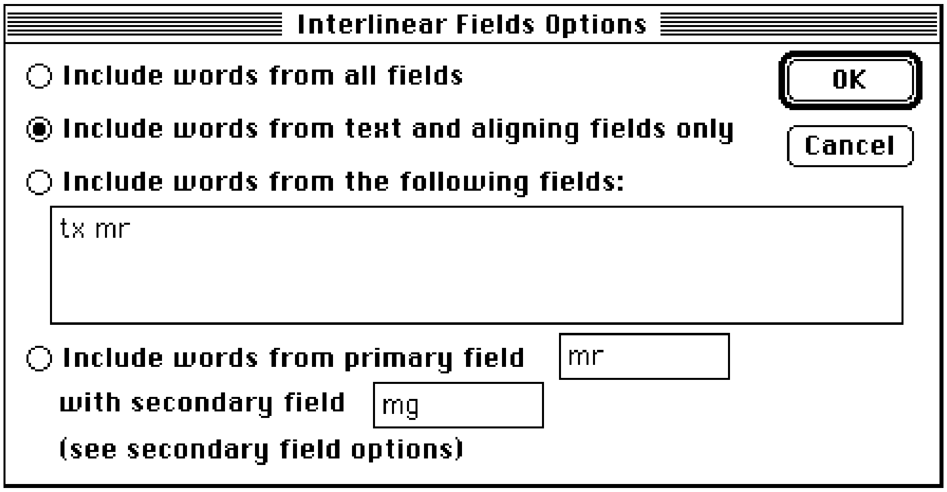
Conc can only work with texts that fit entirely in memory. In addition, the concordance itself needs to fit. If you have a text that is only just too big, you may be able to do useful work by limiting the words that are included in the concordance. For example, the pattern *^a* tells Conc only to include the words that begin with *a*. You could repeat this for each letter to study each part of the alphabet in turn. If you are not quite so short of memory, you may be able to use a range of letters; for example, *^[a-m]* in a typical text will only use about half as much memory for the concordance as including all the words.

Note that telling Conc to omit frequent words does not, unfortunately, save memory. This is because Conc cannot tell that a word is frequent until after it has put all the occurrences into the concordance and sorted it. The frequent words are then removed, but sufficient memory to include them in the first place must be available. For this reason if you are short of memory it may be worth listing explicitly the most common words that you don't want, even if you are generally omitting short words.

4.8 Choosing words from a particular field

In the case of an interlinear document, you have further options for restricting the words included in the concordance. The data in an interlinear document are organized into named fields, and the Interlinear Fields dialog box (figure 10) allows you to control which of these fields are used to build the concordance.

Figure 10. The Interlinear Fields dialog box



Interlinear Fields Options

○ Include words from all fields

● Include words from text and aligning fields only

○ Include words from the following fields:

tx mr

○ Include words from primary field mr

with secondary field mg

(see secondary field options)

By default, words from all fields are included. Alternatively, you may restrict the concordance to just forms from the baseline text and other aligning fields, or specifically list the fields you want. Simply type the field names (without backslashes) into the text box.

The final option here allows you to take advantage of Conc's knowledge of the structure of an interlinear document by making a concordance out of one field (the primary field) but organizing it with data from another (the secondary field). This is discussed in chapter 6.

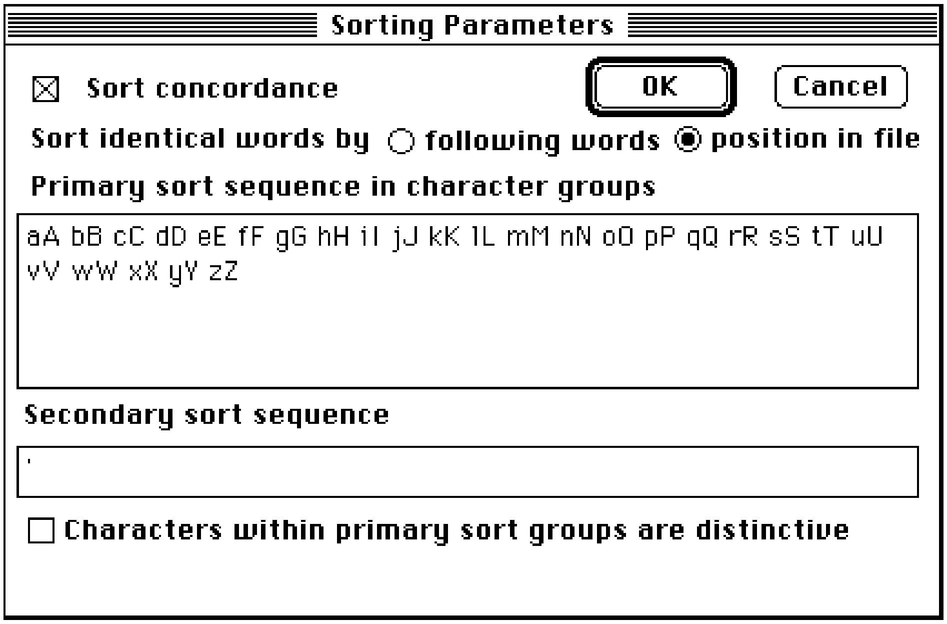
Chapter 5

SORTING

5.1 Sorting options

To control how the words in the concordance are sorted, choose Sorting on the Options menu; you will see the Sorting dialog box as shown in figure 11.

Figure 11. The Sorting dialog box



☒ Sort concordance

Sort identical words by ○ following words ● position in file

Primary sort sequence in character groups

aA bB cC dD eE fF gG hH iI jJ kK lL mM nN oO pP qQ rR sS tT uU vV wW xX yY zZ

Secondary sort sequence

'

□ Characters within primary sort groups are distinctive

The Sorting dialog box offers choices for the way words are sorted. First, you can choose to sort the concordance by checking the button "Sort concordance." If this button is left unchecked, the concordance will not be sorted. This is useful, for example, if all you want to use is the Statistics dialog box to count various things, and you don't want to wait for Conc to sort the concordance (which can take quite a few minutes on a long file). It also has a special application in correcting problems with the use of a secondary file to simulate an annotation field; see the discussion in section 6.4.

Second, if you choose to sort the concordance, Conc first sorts all the words of the text into groups of identical words (by the sorting sequence criteria described below). Within such a group, there are two options for sorting the identical words which are activated by the radio buttons "Sort identical words by following words" and "Sort identical words by position in file". By default, Conc sorts identical words by their position in the file, so the references for a particular word come out in the order in which they appear in the text. Alternatively, you can choose to sort identical words by the following words in the text. In this case intervening word separation characters (punctuation) are ignored. This second option is useful if you are studying sequences of words, especially if your index entries are set to contain more than one word each (see section 7.1).

Note that the above description of sorting may be modified if the secondary field mechanism is in use (see chapter 6).

5.2 Word formation characters and word separation characters

All the characters used in your text (with the exception of those used to indicate references [see chapter 4]) fall into two categories: *word formation characters* or *word separation characters*. In order to control how the concordance is sorted, you must explicitly declare the complete sets of word formation and word separation characters. Any characters used the text that are not declared as either word formation or word separation characters are ignored (again, except reference markers). Such characters will still appear in the text and concordance displays, but will be omitted from the index. For instance, if apostrophe is left out of the character lists, then the English words *it's* and *its* will be listed in the index under the keyword *its* (no apostrophe, since it is ignored).

Word separation characters are typically white space and punctuation characters. White space characters are obligatorily word separators; these include space (ASCII 20), carriage return (ASCII 13) and tab (ASCII 9). For an interlinear text created with the *IT* program, word separation characters are defined in the text model. To declare word separation characters for a flat text, choose the Text Properties command on the Options menu and type them into the text box "Word separator characters." By default this field contains:

. ? ! , ; : ( ) " - – — …

where - is a hyphen, – is an en dash, and — is an em dash.

Word formation characters constitute all the characters that you consider to form the words found in the text. Word formation characters must be declared for both flat texts and interlinear texts. They are listed in the two text boxes found in the Sorting dialog box (see figure 11). These two fields between them should contain every word formation character used in your text. The difference between the two boxes has to do with how the characters affect sorting. The first text box contains characters that determine the primary sorting order of words while the second text box contains characters that determine a secondary sorting order. The next section describes how to specify sorting order using these two text boxes.

5.3 Specifying sorting order

The first text box in the Sorting dialog is called “Primary sort sequence in character groups” (see figure 11). All word formation characters that are relevant to the primary sorting order are arranged in groups in the desired sorting order. Each group contains characters that can be treated as equal with respect to the sorting order. The typical way to use this feature is to ignore the distinction between uppercase and lowercase letters. For example, the default setting of this text box is “aA bB cC dD eE fF gG hH iI jJ kK lL mM nN oO pP qQ rR sS tT uU vV wW xX yY zZ”; this specifies that words are made up of the twenty-six alphabetic characters, sorted in the usual sequence for English. If the button “Characters within primary sort groups are distinctive” is left unchecked, this will have the effect of ignoring the distinction between, say, uppercase *A* and lowercase *a.*

As a demonstration of how to use sort groups and the “Characters… are distinctive” button, load the concordance *Alice1.conc,* ensure that the button is checked and that the button to sort by position in file is checked, rebuild the concordance (if you changed the status of the button), and examine the concordance window. You will see the following lines.

```
147    ...n she went back to the table    for it, she found she could not pos...
155   ...d trying to box her own ears    for having cheated herself in a game
156       was playing against herself,    for this curious child was very fond
171       seemed quite dull and stupid    for life to go on in the common way
110     if I only know how to begin.’    For, you see, so many out-of-the-...
```

This display shows that the word *For* is sorted after the word *for* even though it precedes it in the file. This is so because the difference between *f* and *F* is being treated as distinctive, with *f* preceding *F.*

Now uncheck the “Characters…are distinctive” box in the Sorting dialog, rebuild the concordance, and again examine the concordance window. The entries shown above will now appear as follows.

```
110     if I only know how to begin.’    For, you see, so many out-of-the-...
147    ...n she went back to the table    for it, she found she could not pos...
155   ...d trying to box her own ears    for having cheated herself in a game
156       was playing against herself,    for this curious child was very fond
171       seemed quite dull and stupid    for life to go on in the common way
```

Because the differences within character groups are treated as nondistinctive, the words *For* and *for* are dentical for purposes of sorting and are thus listed in the order in which they appear in the text (since this was the sorting option chosen for handling identical words).

Here are some other ways in which you might want to modify the primary sorting order:

- Add ‘-’ if you want hyphenated words considered as a single word (but be sure to remove it from the list of word separation characters).
- Add some special characters if your text is a grammatical analysis where they have special meanings.
- Add letters used in a non-English alphabet. For example, add Åå to the Aa group to make all four letters sort together, or add them after Aa separated by a space to make them sort separately.

If two words are identical when comparison is made using the sort groupings, Conc then falls back on a “secondary” sort order (unless the words are absolutely identical). If the first difference between two words is the occurrence of a different letter from one group in the main sorting sequence, the one with the letter that occurs first is considered smaller; for example, *And* sorts before *and* in the standard sort sequence. If the first difference is the occurrence of a character from the secondary list, the order of appearance in that list determines the order. If one word has a character from the secondary list and the other does not, the word without the character sorts first. For example, if apostrophe is placed in the secondary list, then *were* will sort before *we’re.*

The characters listed in the primary sort sequence and secondary sort sequence boxes are treated as non-overlapping sets. If you put the same character in both boxes, it will be treated as part of the primary sequence and will be ingored in the secondary sequence. There is one situation where it is useful to do this deliberately, namely if you are using a font where an accent is a separate character. This is the case if you can add an accent to anything, and you type it before or after the character it is added to, rather than typing a different character to get an accented one. For example, you may keyboard a vowel with an circumflex accent as *i^* (two separate characters) in order for it to be displayed as *î.* The usual way of handling this situation is to put all the accents in the secondary string, so that they do not affect the sort order unless two words differ only by an accent. But overstriking accents are difficult to read without an accompanying non-overstrike character. You can get around this simply by placing each accent on some suitable character from the main sort sequence. For instance, your secondary sort sequence box could contain *i^,* which (assuming that *i* is already in the primary sort sequence) has the effect of declaring the circumflex as a secondary sort character while ignoring the vowel *i.*

If some word does not seem to sort as you expected, it may be that it contains some almost imperceptible letters (some fonts have a space only a pixel wide used for precise spacing of letters) or some characters similar but not identical to the ones you specified (some fonts have several variants of the same accent, designed for correct position over vowels of different widths). If you are using such a font, temporarily switching to a conventional font like Geneva may help you to work out what is going on.

5.3 Sorting as a discovery procedure

The use of the sorting sequence in the sorting dialog box is not limited to getting words to print in the "correct" alphabetical order for the language; in fact, it is not yet powerful enough to do that really well, since it does not understand sorting digraphs.[20] You can also vary the collating sequence creatively to help discover patterns in data.

For example, you can easily set up a collating sequence in which all the vowels come first to help locate patterns involving vowels. You can arrange to order the vowels by height, or the consonants by voicing and then by point of articulation (or the reverse).

Another useful trick is to put letters of lesser interest in the secondary sorting sequence. For example, if you put all the consonants there, your words will be sorted by vowel patterns. If you put everything in the secondary sequence except the characters you are using to mark tone, the words will be sorted by tone patterns.

You can come up with as many variations as you like on this idea. It is particularly useful in combination with the letter concordance option (see below) for studying phonological problems.

[20] Even that would not make it perfect; different languages have different conventions for handling special letter sequences and words, such as (in English) treating *Mc* and *Mac* specially at the start of words, and ignoring *The* and *A* at the start of titles. Getting this sort of thing to happen is well beyond the scope of a simple collating sequence such as Conc uses.

Chapter

6

USING A SECONDARY FIELD

6.1 The use of secondary fields

In some situations, there may be no way of getting the concordance to contain the words that are wanted in the order wanted by using the information that Conc can derive from the words themselves. Instead, it may be necessary to use information from one of the annotation fields, even though you might not want those annotations included as key words in the concordance. For example:

- If you have a text annotated with glosses, you could do things like sorting the concordance by gloss, to see which words are glossed the same way.
- If you want to study the grammatical structures of a text, you could create an annotation containing a grammatical description for each word. You could do things like sorting the concordance by part of speech, or selecting nouns that are immediately followed by a verb.
- If you want to study the sources for a text, you could make an annotation containing, for each word, a code indicating which source document it is derived from. Then you can make a concordance with all the words from a particular source, or sort by the source of a word.

Many other kinds of annotation are discussed in *How to use IT*.[21] Most aligning annotations are candidates for helping to select the interesting forms in the field they annotate.

At present, Conc can only cope with the relationship between two fields. One of these is called *primary* and is the field which contains the words that are to appear in context in the concordance. The other is called *secondary* and contains annotations which Conc will use to modify the sorting sequence or word inclusion patterns that would be used if you simply made a concordance of the primary field.

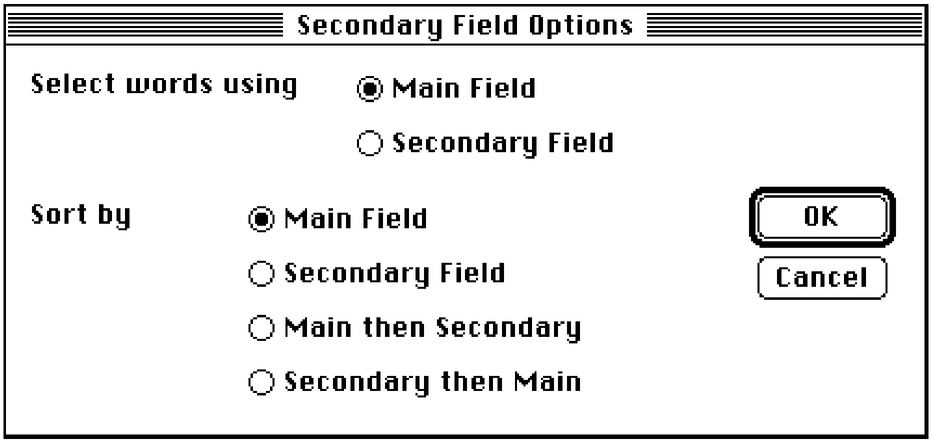
To specify that primary and secondary fields should be used, you click the appropriate button in the Interlinear Fields dialog box (figure 10 above) and enter the names of the two fields into the text boxes in that dialog box. The fields specified must be aligning fields (including the text field) since there is no particular relationship known between words in other fields.

[21] *How to use IT: interlinear text processing on the Macintosh, by* Gary F. Simons and John Thomson, Linguist's software, Edmonds, Washington, 1988.

6.2 Controlling word inclusion

The Secondary Field dialog box (found on the Options menu), shown in figure 12, controls how the data from the secondary field is used.

Figure 12. The Secondary Field dialog box



Secondary Field Options

Select words using
● Main Field
○ Secondary Field

Sort by
● Main Field
○ Secondary Field
○ Main then Secondary
○ Secondary then Main

First, you can specify whether word inclusion should refer to the main or secondary field. This has no effect if you also select “Show all words” in the Include Words dialog box and don’t specify anything to be excluded, but if you choose one of the other options, it will work as usual if you select “Main Field.” If, instead, you choose “Secondary Field” the functions that decide which words to include will, in each case, examine the corresponding word in the secondary field. Thus, for example, if you choose this option along with the option “Omit words in the following list” (from the Omit Words dialog), words will be omitted if the corresponding word in the secondary field is in the list. If you choose it with the option “Include words that match one of the following patterns” (from the Include Words dialog), the word in the secondary field will be checked against the patterns. For example, if you have a grammatical category annotation that begins with N for nouns and V for verbs, you can make a concordance of all the nouns using the pattern ^*N*.

The option “Omit words that occur more than N times” (from the Omit Words dialog) works a little differently. Conc can only detect multiple occurrences of a word when they are sorted together. Therefore, the option to omit words that occur frequently is based on how words are to be sorted, as described below, not on the “Select words using…” choice. So, for example, if you say to include words based on the main field but sort by the secondary field, and then to omit words that occur more than ten times, Conc will omit words where more than ten have the same corresponding word in the secondary field. If you are sorting using both fields (either secondary then main or main then secondary) words are considered to be the same if both main word and secondary word are identical. Similarly, if you are sorting by the secondary field, a new index entry is started for each different word in the secondary field (see section 7.1 on building an index).

6.3 Controlling sorting

Next, you can specify how sorting is done. You can sort so that the words in the main field or in the secondary field are in order, or use a combination. For example, suppose the secondary field contains an indication of the source of a word. You might only have five or six sources, and you may not want the words from a given source sorted by either the source of the following word or the position in the file (the options in the Sorting dialog box). Instead, you might like words from a given source sorted alphabetically. To achieve this, just choose “Sort by secondary then main.” Alternatively, you might like to have the words alphabetical, but to sort by source where the same word occurs several times. For this choose “Sort by main then secondary.”

The interaction between these options and the ones in the sorting dialog box is a little complicated. Most users probably won’t need to worry about it, or will be happy to try and see what happens. But for those who like to know, here is how it is done.

If you are sorting by secondary and then main, the program first compares the words in the secondary field. If they are different, that ends the comparison. If they are the same, it next tries the words in the main field. Again, if they are different, that settles it.

If both pairs of words are the same, what happens next depends on the setting of the sort method in the sorting dialog box. If it is set to sort by position in file, then the position settles things at once. If it is set to use the following word, the program next compares the following word in the secondary field. If the pair of words there are also the same, it then tries the following pair in the main field. This continues, alternating between the fields, until a pair of words that are different is found. All this is reversed if sorting is by main then secondary.

If you are puzzled by the way words are selected or sorted, see that the box “Show secondary word after main in concordance” is checked in the Display dialog box. To reduce confusion, Conc always shows the secondary word in the index window if it affects sorting.

6.4 Using a secondary file with a flat text file

By far the most reliable way to annotate a text is to create proper aligning annotations using *IT*. Such files can be displayed with the annotations neatly aligning. However, some users may have data in another form: two files containing parallel word sequences. For example, one file could could contain the Greek text of a book of the New Testament, while another contained a grammatical tag for each word.

Conc can use such a pair of files as if the second file were an annotation of the first. The last option in the Text Properties dialog box on the Options menu controls this. Turning it on has no immediate effect, not even after you close the dialog box. This is because no secondary file has been loaded. But when you next build a concordance, you will be asked for a secondary file as well, and all the secondary file options will then take effect.

The major problem with this approach is that the words in the secondary file must correspond *exactly* to words in the main file. References in the secondary file should be identical to those in the main file. If Conc does not find the same number of words, it warns you, but still builds the concordance. If there are extra words in the secondary file they are ignored, while if there are not enough the extra words in the main file just have nothing corresponding.

It can be quite difficult in the case of a long file to work out just where something was left out or added to make the numbers of words different. To help detect this, turn on the check box “Show secondary word after main in concordance” in the Display dialog box, and in the Sorting dialog box, choose the option “Don’t sort concordance.” Set the word inclusion options to include everything. Then, look at the concordance window. It is now in the order of the words in the file, and each main word has beside it the matching word from the secondary file. This makes it relatively easy to scroll through the concordance and locate the first word that is matched with the wrong thing. You will still need to use a text editor to correct whichever file contains the wrong information, but at least this technique can help you locate the problem quickly.

Chapter

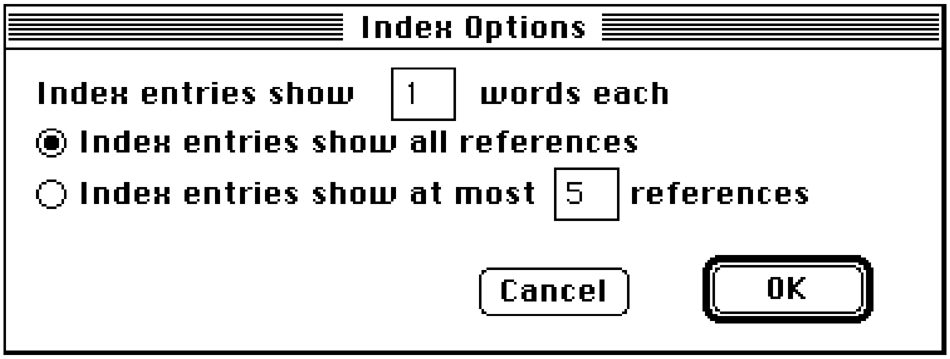
7

OTHER FACILITIES

7.1 Building an index

As well as a concordance, Conc can produce a more compact indication of where words occur in a document, namely an index. To build an index, choose the Index command on the Build menu. The index window shows each different word in the document, shows how often it occurs, and gives a list of the references. An example was shown in figure 3 above. The list of references can be scrolled using the horizontal scroll bar on the window; the word and count don't move when you do this. Several aspects of the index window can be controlled using the Index dialog box, accessed through the Options menu (figure 13).

Figure 13. The Index dialog box



Index Options

Index entries show 1 words each

● Index entries show all references

○ Index entries show at most 5 references

First, you can control how many words occur in an index entry. Normally, a new index entry is started for each different word, and only one word is displayed. But it is sometimes useful, particularly in combination with the use of a secondary field, to count the number of times particular sequences occur. If the number of words in an index entry is set to two or more, that number of words will be displayed, and whenever either changes a new index entry will be started. Note that it is highly recommended that you sort by following words (rather than by position in the file—see the description of sorting above) if you set up an index with more than one word displayed; otherwise, identical groups of words may not appear together in the concordance, and the index will not notice that they are the same, with the result that you will get more than one index entry for the same group of words.

Next, you can control how many references are included. Some words have a lot of references, and it may be useful (for example, to save paper) only to see the first few references in the longer lists. This option controls whether all references are displayed, and if not, what is the maximum number.

Note that the index may be a little confusing if you are sorting by secondary field. A new index entry starts whenever one word in the concordance is not the same as the one before it. If the index is sorted by the secondary file, then the word identifying the index entry is really the secondary file word—a new index entry will be started for each different secondary file word. In this situation Conc shows the index entry as “…/*secondary word*” to show that what is being counted is the number of adjacent occurrences of a particular secondary word. If you sort by secondary then main or main then secondary, the index will count the number of times that a particular (main, secondary) pair occurs.

In the Sorting dialog box there is a button titled “Characters within primary sort groups are distinctive.” The setting of this button has a dramatic effect on an index. If the button is checked, then the index will contain entries for each word exactly as it appears in the concordance. If the button is not checked, then the index will contain entries that collapse words that differ only by using differenct characters from within a sorting character group. The index entry will be spelled using the first character of each group. To demonstrate this, load the concordance *Alice1.conc,* ensure that the button is checked, and build an index. The first few entries in the index window will look like this:

```
about           (8)     33, 36, 47, 56, 101, 106, 122, 139
across          (2)     25, 27
actually        (1)     23
ADVENTURES      (1)     1
advice          (1)     153
advise          (1)     152
afraid          (1)     73
after           (4)     27, 29, 45, 142
After           (1)     144
afterwards      (1)     21
again           (4)     30, 59, 70, 94
```

Notice that the entries for *after* and *After* demonstrate that the members of the character groups are considered distinctive. In this example, uppercase and lowercase are sorted differently resulting in two index entries.

Now uncheck the box in the Sorting dialog to ignore distinctions withing character groups and rebuild the index. The first few entries in the index window will now look like this:

```
about           (8)     33, 36, 47, 56, 101, 106, 122, 139
across          (2)     25, 27
actually        (1)     23
adventures      (1)     1
advice          (1)     153
advise          (1)     152
afraid          (1)     73
after           (5)     27, 29, 45, 142, 144
afterwards      (1)     21
again           (4)     30, 59, 70, 94
```

Two things are apparent from this display. First, the index entries are spelled using the first character of each sorting character group. And second, by ignoring distinctions based on character groups, multiple index entries such as *After* and *after* shown above are collapsed into a single entry, namely *after.*

3.6 Generating an index with page references

A special application of Conc is to produce a conventional word/page-number index for a piece of text. For this to work, it is necessary for the references to be page numbers. At present Conc does not know how to determine page numbers from word processor document files, so you have to give it some help. To prepare a file for a page-number index, do the following:

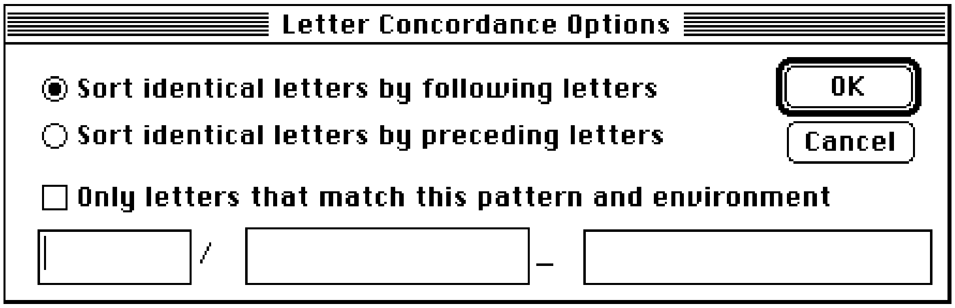
1. Choose two characters that do not appear in your document to be used as brackets. I will assume [..] are chosen, but if you choose something else, remember to set them in the Text Properties dialog box.
2. Open your document in the word processor, and repaginate it if your word processor has such a command. Of course, for the index to be very useful, you must be working with the final version of your document, exactly as it will be printed.
3. Starting from the last page in your document, and working towards the front, put a page number on each page, right at the top of the page, enclosed in whatever brackets you are using. For example, on page ten you would put [10] at the start of the first line. (The reason for starting from the end is so the later numbers will be in the correct places even if your word processor repaginates automatically as you insert the earlier numbers. If your word processor doesn't do this you can start at the beginning.)
4. Save your document, to a new file name, as a text-only file.
5. Run Conc, and choose the Text Properties… dialog box on the Options menu. Choose the radio button "References are marked by…" and set the sort of brackets you have used. You may want to set other options at this time (see below) to omit certain words from your index.
6. From the File menu choose Open and open the file you just created.
7. From the Build menu choose Index.
8. You should now see the index. You can use the commands on the File menu to save or print all or part of it.

The index produced in this way is a rather primitive one. In one way, it includes too much information: not every word in a document is worth indexing, and not every occurrence of even an interesting word is worth indexing. You can improve this somewhat by using the options that control word inclusion. In another way, Conc's index includes too little information. A good index should include references to places where key concepts are discussed even though the context may not include the key word for the concept. Also, a good index should have sub-headings for words that occur frequently and use ranges when a word appears on many successive pages. Nevertheless, the index Conc produces may be a useful starting point, particularly if you don't have a lot of time to devote to polishing an index.

7.3 Building a letter concordance

Normally, Conc makes only one entry in the concordance for each word it finds in the document. However, sometimes you may want to see letters (i.e. characters) in their contexts. The Letter Concordance command on the Build menu will produce a concordance of each letter found in the text. To set various options related to building a letter concordance, choose the Each Letter command on the Options menu (see figure 14).

Figure 14. The Letter Concordance Options dialog box



● Sort identical letters by following letters
○ Sort identical letters by preceding letters
□ Only letters that match this pattern and environment

When you choose to build a letter concordance, Conc ignores the selected sorting method indicated in the Sorting dialog box (and the options in the Secondary Field dialog box, if that is in use) and simply sorts identical letters either by the following letters or the preceding letters in the word. Letters that you have specified not to be in words (or to be of only secondary interest) are ignored. White space at the end (or beginning, if sorting by previous letters) of a word is considered to come before anything else.

A secondary field, if you have specified one, is ignored when a letter concordance is in use, except for its use in determining which words to include. If you ask for secondary field words to be displayed, you will find that each letter of a primary file word is associated with the entire corresponding secondary field word. No attempt is made to make a letter-for-letter match.

The letters included in the letter concordance are normally all the letters you have specified as word-forming (see section 5.2 above) in all the words that you have told Conc to include using the include and exclude dialog boxes. You can further tell Conc to include only certain letters by specifying patterns for the letter itself, and for its following and preceding contexts. This is done by checking the box “Only letters that match this pattern and environment.” (For a discussion of patterns and the “magic” characters you can use within them, see section 4.4. The following discussion merely describes how to use patterns to build a letter concordance.)

The notation used to specify a letter pattern is similar to the notation used for phonological rules, namely *a / x___y,* where *a* stands for the target letter (or set of letters), *x* stands for the preceding context, and *y* stands for the following context. These three parts correspond to the three text boxes shown in figure 14 above. Each box can be filled with a pattern.

The first text box allows you to specify the “head letter” which is the actual concordance entry. The simplest use of this pattern-matching capability is to specify a letter or set of letters you are interested in. For example, *a* would limit the concordance to just the letter *a; [aeiou]* would cause just the vowels to be included; and *[^aeiou]* would include just the non-vowels.

The second box allows you to specify preceding context, that is, what must appear immediately before the head letter. For example, to see a concordance of all the letters that can appear after a *t*, simply put the letter *t* there. Note that this will match only if the *t* is immediately before the head letter; if you want all the letters which come anywhere after a *t*, use the pattern *t_** (t followed by any number of word-forming characters).

Similarly, you can use the third box to specify following context. To find all the letters that occur immediately before a vowel, put the pattern *[aeiou]* in the third box.

You can combine the boxes; only letters where all three patterns match will be included. To find places where a *t* occurs between vowels, put a *t* in the first box, *[aeiou]* in the second box, and *[aeiou]* in the third box.

The pattern matching facility for remembering what was matched in one place and using it elsewhere carries over from the head letter to the following context.[22] Thus, for example, to make a concordance where each head letter is the first of a pair of identical letters, put *\(_\)* in the first box and *\1* in the third. (That is, the head letter can be anything, but remember what it was; the next letter must be the thing that was remembered.)

Note that the letter before the first letter in a word is a space, not the last letter of the preceding word. Thus the *t* pattern for preceding context

[22] Unfortunately, it does not currently carry over from the preceding context to the head letter or following context. If this capability would be valuable to you, please let me know.

will never find the first letter in a word, even if the preceding word ends in a *t*. If you want to include such letters, use the pattern *t* * (t followed by zero or more spaces).

You can use ^ in the preceding context and $ in the following context to mean the beginning and end of what is matched. Thus for example you could include letters that follow an initial vowel with the preceding context pattern *^[aeiou]* (start of word, then a vowel). Note that if a pattern includes a space, or something which could match a space, Conc includes the previous (or following, or both, as appropriate) word in the string to compare; ^ then matches the beginning of that extra word, while *$* matches the end of the extra word. Conc always includes in the appropriate place any adjacent morphemes, so ^ for example matches the beginning of the first adjacent morpheme.

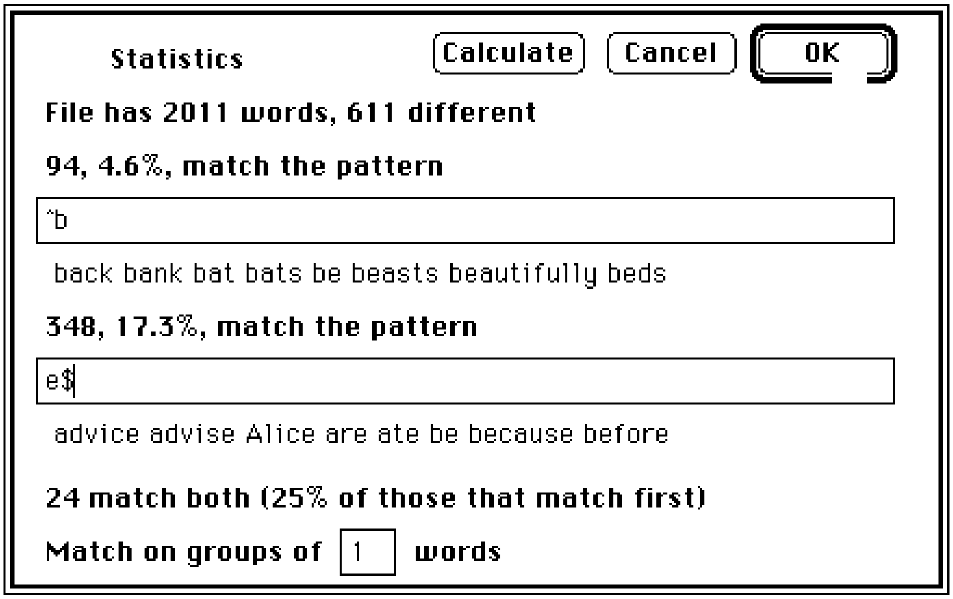
A *$* at the end of the preceding context pattern, or a ^ at the start of the head letter or following context patterns, has no special meaning; it just matches itself.

Note that if you want to match morpheme-break characters (e.g to make a concordance of the letters which occur after morpheme breaks) you need to turn on the “Include non-word chars” option in the Pattern chars dialog box. If you want morpheme-break characters to be head letters, you must place them somewhere in the sort sequence so that they are considered word-forming; Conc only considers word-forming characters as candidates for inclusion in the letter concordance.

7.4 Statistics

Conc can count words that match a pattern and perform some simple calculations on the results. The Statistics command on the Windows menu brings up the dialog box used for this purpose, which is shown in figure 15.

Figure 15. The Statistics dialog box



File has 2011 words, 611 different

94, 4.6%, match the pattern

^b

back bank bat bats be beasts beautifully beds

348, 17.3%, match the pattern

e$

advice advise Alice are ate be because before

24 match both (25% of those that match first)

Match on groups of 1 words

This dialog box has two patterns, which are regular expressions as described in section 4.5. For example, here the first pattern describes words beginning with *b*, while the second describes words ending in *e.*

Figure 15 actually shows the appearance of the dialog box *after* the Calculate button is clicked. This causes Conc to calculate how many words in the document match each pattern. To help ensure that the patterns are describing what you want them to, several examples of matching words are displayed just under each pattern item.

In this case, the dialog box informs us that there are 2011 words in the concordance.[23] Many of these are duplicates; there are only 611 distinct words (this is the number of lines in the index window). Of the 2011 words, 94 start with *b*, and 348 end in *e*. Finally, 24 both begin with *b* and end in *e*. This is 25% of the words that begin with *b*.

With clever use of patterns, it should be possible to use this facility to test a variety of coocurrence hypotheses.

[23] Note that this is not necessarily the number in the document as a whole, if you have specified that some words should be omitted from the concordance.

Chapter

8

MENU SUMMARIES

8.1 The File menu

Append… allows you to open another text file and append it to the end of the one currently loaded. In the concordance window is open, it will close it before appending the new file.

Open… opens a text file, a previously saved Concordance, or an options file. (Concordance files include a record of all the options used to create them, and will restore those options when opened.)

Save creates a file containing a record of the current text, concordance, and options. It is much faster to reload a sorted concordance from a file than to regenerate it, so it is worth saving a concordance that you expect to use often. Also, saving a concordance allows the Revert command to be used. The saved concordance cannot be used by any other program. If you have not saved the concordance before, Conc asks you for a name; otherwise, it simply replaces the previously saved version. Saved files are usually about twice as big as the original text file—more precisely, they are the size of the original file, plus four bytes for each entry in the concordance, plus a small overhead the size of an options file. Saved files for interlinear documents are larger (about five times the size of the original), since they also save the data structures representing the annotation structure.

Save As… is the same as Save except that it always asks for a name and creates a new file.

Close closes the text, concordance, or index window, whichever is foremost. A window can generally be reopened by using a command on the Windows menu. However, if you close all the main windows associated with a document (text, concordance, and index), Conc forgets about that document; you will have to reopen it to get the windows back. If you have changed the concordance (for instance, by sorting it differently), Conc will prompt you to save the concordance if you attempt to close the concordance window.

Revert… restores the options and concordance that were in effect the last time the concordance was saved. (It does nothing if you are working with a newly generated concordance that has not been saved.)

Save All Options… saves to a file the current settings in the Font, Options, and Display menus plus the page layout and headings specified by commands on the File menu. An options file can be loaded using the Open command on the File menu, thus restoring all the saved options.

It is useful to work out a set of options that work well with a particular kind of file, save those options, and restore them before opening one of those files. The files created in this way “belong” to Conc just like complete concordances; they have a key icon with a miniature menu bar, and Conc can be started by opening them. If you open one while a concordance is open, you will need to rebuild the concordance in order for the new options to take effect.

It is also possible to save a subset of the options. This could be useful if, for example, you have a standard set of headings that is useful for several kinds of texts (which otherwise require different options) or several sets of patterns and collating sequences which it is useful to switch between without changing anything else. To do this just open the dialog boxes for the options you want to save, make sure one of them is the front window, and choose Save or Save As… from the File menu. Conc then makes an options file containing just the information from the dialog boxes that are open. So, for example, to make an options file that only affects the collating sequence, choose Save while the Sorting dialog box is the only options window open. To make one that will restore a particular word inclusion pattern plus an unusual set of pattern-matching characters which go with it, open those two dialog boxes and choose Save. Options files created in this way work just like those created by Save All Options except that opening them changes only a subset of the options.

Export Text/Concordance/Index As… creates a plain text file of the contents of either the text, concordance, or index window, depending on which is active. An exported text file can be either a flat text or in interlinear ITX file. In an exported concordance file, each line consists of reference, tab, context before the word, tab, then the word and the following context. In most word processors, if you put a right tab closely followed by a left tab for all these lines, you get a display rather like the one produced by Conc. This is useful for including a concordance in another document, or printing it in more sophisticated ways than Conc is capable of doing. If you have already saved the document in Conc format this command appears as Export *filename* As.

WARNING Use this option with care; it can produce very large files, more than ten times the size of the original text. There is no way Conc can tell in advance whether the file will fit, without making things much slower, so make sure you have plenty of room. You can halt the export process by clicking the Abort button on the progress indicator. Consider whether you could better use the Export Selection… menu option to save just part of the concordance.

Export Selection…. You can select a group of words (lines) in either the text, concordance, or index window by clicking and dragging or by shift-clicking. Then you can use this command to save just the lines so selected. If you want to make files of a whole large concordance, saving a number of moderate sized selections could be a good idea.

Print…. This makes a printout of the entire contents of either the concordance or index window (the text window can be exported, but not printed). Printing has been tested successfully on an ImageWriter I and II and on a Laserwriter.

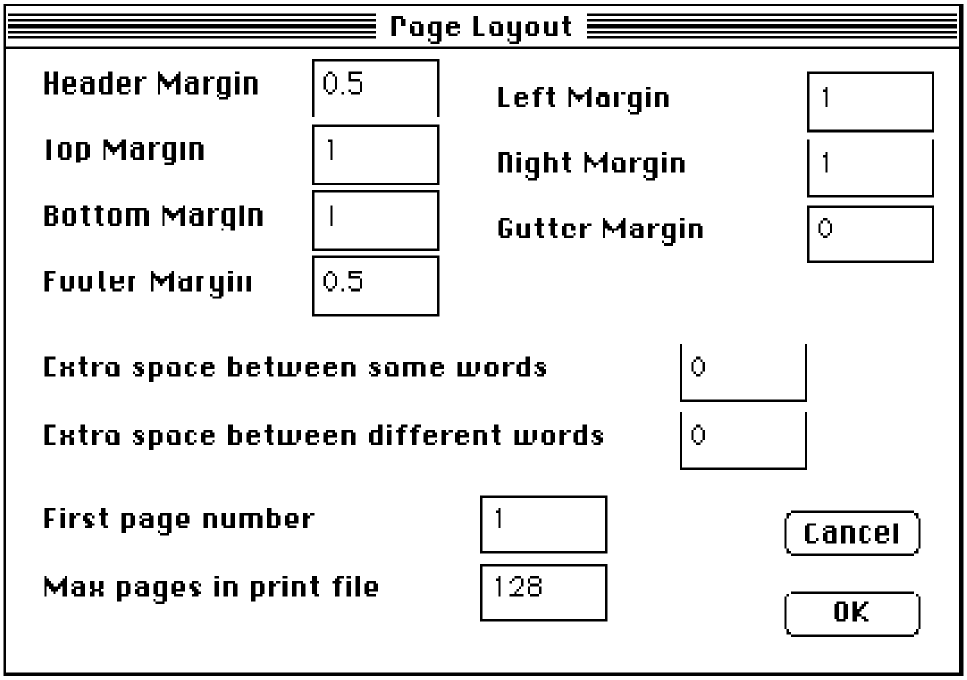
Print Selection…. This is similar to Print…, but prints just the selected lines.

Page Setup…. This brings up the standard dialog box used for setting the paper type, orientation, etc.

Page Layout…. This brings up a dialog box (figure 16) which allows you to specify margins and related things. Measurements are from the edge of the page; Conc will warn you if the margins are smaller than your printer can handle. The numbers are expressed in inches, but rounded to the nearest pixel. In addition to the usual top, bottom, left and right margins, there are the following settings.

- Margins for the headings and footers.
- A gutter margin, which is added to the left of odd pages and the right of even ones.
- An amount of space added between lines that have different words. This can be used to emphasize a change from one word to another.
- An amount added between lines with the same word. This could be used to double space, for example, or just to produce a slightly less dense printout.

Figure 16. The Page Layout dialog box



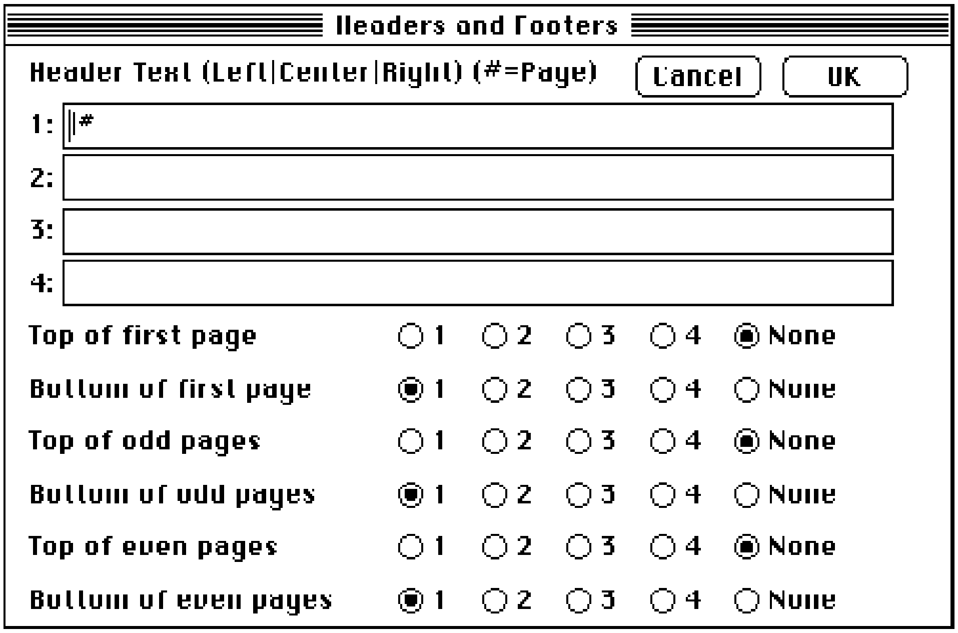
Page Layout

Header Margin	0.5	Left Margin	1
Top Margin	1	Right Margin	1
Bottom Margin	1	Gutter Margin	0
Footer Margin	0.5		

Extra space between same words	0
Extra space between different words	0
First page number	1
Max pages in print file	128

Headers…. This brings up a dialog box, shown in figure 17, which allows you to specify up to four headers or footers, and to indicate which of them should appear on the top and bottom of the first page, and subsequently of odd and even pages. A header or footer can have up to three parts, separated by '|' (vertical bar): one that is printed flush with the left margin, one flush with the right, and one centered. For example: *left*|*center*|*right*. A heading may contain the # character, which if present is replaced by the page number. Thus, the default heading, which by default appears on the bottom of all pages, is a centered page number, indicated by |#.

Figure 17. The Headers dialog box[24]



Header Text (Left|Center|Right) (#=Page)

1: |#

2:

3:

4:

Top of first page ○ 1 ○ 2 ○ 3 ○ 4 ● None

Bottom of first page ● 1 ○ 2 ○ 3 ○ 4 ○ None

Top of odd pages ○ 1 ○ 2 ○ 3 ○ 4 ● None

Bottom of odd pages ● 1 ○ 2 ○ 3 ○ 4 ○ None

Top of even pages ○ 1 ○ 2 ○ 3 ○ 4 ● None

Bottom of even pages ● 1 ○ 2 ○ 3 ○ 4 ○ None

Quit. Exits the Conc program after prompting to save the concordance if necessary.

8.2 The Edit menu

Cut, **Copy**, and **Paste** work in the usual ways on text you are editing in the various dialog boxes. In addition, the selected text in any window may be copied to the clipboard.

Copy picture is available only when the active selection is in the text window and it is showing an interlinear document. It copies the selected lines to the clipboard as a picture, which can be pasted into most other programs without losing the alignment and other layout.

[24] See section 4.5 of the *IT* manual.

8.3 The Font Menu.

This is stock standard. It affects the active window. Thus you can have different fonts or sizes for the concordance and the text.

In interlinear documents, you can change the font and size of different fields independently. All fields that contain selected text are affected, and so are all other fields of the same type (i.e. the same marker). All extraneous fields have the same font and size.

The font used for editable text in the dialog boxes is always the font currently selected for the concordance window. This is useful if for example you want to type a list of Greek words to look for or omit. Once you have opened a dialog box you can use the Font menu to change fonts within it. If you change the font on the Headers dialog box, that font will be used to print the headers; the other dialog boxes don’t remember changed fonts.

The font selected in the Interlinear Layout dialog box is used to locate the appropriate narrow space character specified there for padding text in copy operations.

8.4 The Options menu

Sorting… controls the sorting order of the concordance. Sorting is discussed in chapter 5.

Include Words… controls which words will be included in the concordance. Word inclusion is discussed in chapter 4.

Omit Words… controls which words will be omitted from the concordance. Word omission is discussed in section 4.2.

Text Properties… controls how references are handled. References are discussed in chapter 3. The Text Properties dialog box also controls word separation characters, discussed in section 5.2.

Interlinear Fields… controls which fields of an interlinear document are included in the concordance. See section 4.8 on selecting fields.

Secondary Field… controls word selection and sorting paramters related to the use of a main and secondary field from an interlinear document. The use of main and secondary fields is discussed in chapter 6.

Each Letter… offers sorting and selection options for letter concordances. Letter concordances are discussed in section 7.3.

Index… offers options related to building indexes. Indexes are discussed in section 7.1.

Pattern Matching… displays information related to Conc’s pattern matching facility. Pattern matching is discussed in sections 4.4 and 4.5.

Restore Last Build… discards all recent changes made to the various options settings and restores the options to the setting at the last time the concordance was built.

8.5 The Layout menu

Set Wrap Length… adjusts the length of the text lines to fit the size of the window.

Wrap Long Lines… sets various options related to wrapping lines in the text window.

Display… sets various options related to the display of the concordance, including references and amount of context.

Interlinear Layout… sets various options related to the display of an interlinear document in the text window such as the amount of verticle spacing between lines and bundles.

8.5 The Build menu

Word Concordance builds a concordance of all selected words in the current text and displays the results in the concordance window. A word concordance can be built for either a flat text or an interlinear text. The Word Concordance command is dimmed if no further changes have been made in the various options that control the content of the concordance. As soon as you made any changes to the concordance options (such as sorting or word inclusion), the Word Concordance command becomes available again. Notice that after changing options settings, the concordance is *not* automatically rebuilt; you must explicitly issue the Word Concordance command to rebuild it using the new options.

Morpheme Concordance builds a concordance of all the selected morphemes in the current interlinear text. It is not possible to build a morpheme concordance of a flat (noninterlinear) text. The command is dimmed under the same conditions as described above for the Word concordance command.

Letter Concordance builds a concordance of all the selected letters in the current text. A letter concordance can be built for either a flat text or an interlinear text. The command is dimmed under the same conditions as described above for the Word concordance command.

Index builds an index of the current concordance. Before you can build an index you must first build either a word or a morpheme concordance. It is not possible to build an index of a letter concordance.

Statistics… counts words that match a pattern and performs some simple calculations on the results. See section 7.4 on using the Statistics command. Notice that Statistics is a modal dialog box, which means that you do not have access to any other windows or dialog boxes until you close it.

8.5 The Windows menu

The Windows menu allows you to bring to the front any window which you have closed or which has become hidden (except the dialog box windows—to bring one of them up choose the appropriate command on the Options menu). Closing all windows associated with a concordance effectively closes the concordance and the Windows menu will become unavailable. You will need to use the Open command to reload the concordance.

Text brings the text window to the front.

Concordance brings the concordance window to the front.

Index brings the index window to the front.

Tile Windows resizes all currently open windows to fit on the screen without overlapping. Only horizontal tiling is available.

Appendix A

LIMITATIONS, PLANS, AND RECENT CHANGES

A.1 Major known problems

The following are recognized limitations of the present version of Conc. In some cases the needed fix is obvious; for others, possible fixes are described in the following section. I can make no promises to fix these, but they are at least under consideration!

- The text to be indexed, and the index itself, must fit in the available memory.
- Printing the text window is not supported.
- Printing is limited to 128 pages per printout.
- There is no way to interrupt lengthy operations such as sorting a large file. (However, exporting can now be interrupted.)
- Under certain low-memory conditions, Conc may exit rather ungracefully.
- Lists of words to include or exclude are limited to the 240 characters that a simple dialog box item will hold.
- An interlinear display cannot be produced when a secondary file is used to simulate an annotated text.

A.2 Planned improvements

A number of extensions to Conc are planned. If there are some you particularly need, or something else you think is a logical and useful extension, let me know. I am making no promises about implementing *any* of these, since Conc is a spare time project; this is just to let you know some of my ideas, in the hope of inspiring even better ones from you!

- Improve printing so it can do really big concordances by making several print files.
- Overcome the present requirement that the file fit in memory. I have not yet found a way to do this with reasonable performance.
- If the program runs out of disk space while saving a concordance, truncate to end of last line, and invite the user to insert a new disk and continue with a new file.
- Implement an on line help facility.
- Provide a more user-friendly form of pattern matching. One idea is a pop-up menu of the magic characters. Any other ideas?
- Allow longer lists of words to include or omit. Most likely allow

opening a file of them, in addition to adding a scroll bar to the relevant dialog box items.

- Allow editing of main/secondary files; especially useful would be a function to insert/delete a word from the secondary file to correct alignment.
- Allow a list of named patterns, and allow them to be combined using “not,” “and” and “or.”
- Do sorting in the background, while running other applications under MultiFinder. (Not very useful unless you have lots of memory, in view of Conc’s appetite, but then some people *do* have lots of memory.)
- Do selection of words to include in the background, displaying and scrolling whatever have been found at once. This would make Conc a much more useful retrieval tool.
- Allow Conc to work with a subset of a large document, to test options before constructing a full concordance.
- Allow digraphs and multigraphs.

A.3 Recent Changes

Version 1.22 of Conc, and some subsequent versions, were fairly widely distributed. For the benefit of users who are familiar with one of them, here are the main changes since:

1.22 to 1.23:

• Slowness in the presence of lots of fonts fixed.

• Compiled under MPW, so hopefully less likely to crash.

• 32K word and line limits removed (however, text must still fit in RAM).

• Manual selection in the text window, scrolling the other windows.

• Copy selection in front window to clipboard.

• Handle standard format texts (up to 9 levels of reference markers).

• Locate words by typing (as in standard file dialog box).

• Format of Options file changed slightly to make radio button settings more readable and portable. This means old options files may not produce quite the results you expect.

1.23 to 1.24:

• Removed a bug introduced in 1.22 which caused scroll bars not to work properly in windows with more than 1000 lines.

• Moved “Show all words” into the “Show only” dialog box, and renamed “Show only” “include.” Renamed “Show all but” “except.” Changed the word selection algorithm so that both sets of conditions are applied—that is, the “except” dialog box is no longer an alternative to the “include” one, showing everything in the file except what is described in the “except” dialog box; instead, the concordance includes everything specified by the “include” dialog box and *not* specified by the “except” dialog box. The original behavior of the “except” dialog is available by choosing “include all words” in the “include” dialog box, but choosing other options in that dialog box now provides new possibilities.

• Extended the “exclude” dialog box to allow long and infrequent words to be omitted, as well as short and frequent ones.

• As a consequence of these changes, old options files may give slightly unexpected results for the include and exclude dialog boxes.

• Added code to display the watch cursor during long operations, and added progress indicators for the two longest operations, creating and sorting the concordance. Note that the progress of the “sort” indicator is sometimes a little irregular.

1.24 to 1.25:

• Removed a bug which caused clipboard contents to be pasted twice when the Paste command was issued in Conc using Command-V and a desk accessory was the front window.

• Added the facility described in section 4.5 for progressively restricting the contents of a concordance.

• Added the facility described in section 7.3 for building a concordance of the letters in a document.

1.25 to 1.26:

• Renamed “Save Concordance” “Export concordance,” to make way for the new Save command.

• Added Save and “Save as” commands, which create a new type of file containing a complete set of options, the original text, and the concordance produced by that input. Reloading such a file is generally much faster than regenerating and resorting a concordance.

• Renamed Open Source to Open, and made it recognize the new saved concordance files as well as text files.

• Added the Revert command, which if the current concordance has been saved restores all options and the concordance to the way it was when it was saved. This is generally much faster than Undo.

• As a result of this, Conc now creates two kinds of files: text files, which contain only a record of the options, and Conc files, which contain complete concordances. (Conc also creates exported text files, but it is not useful to reopen those using Conc.) The key-in-document icon previously used for options seemed most appropriate for complete concordance documents, so a new icon (with a key next to a pulled-down menu) was invented for the options. You probably won't see these icons if you have an earlier version of Conc unless (after putting Conc on your disk) you rebuild your desktop file by holding down the command and option keys while inserting the disk to be fixed (or while rebooting, for a hard disk). WARNING: rebuilding the desktop in this way destroys any file comments you may have; don't try it unless you don't use the Finder file comments. There are programs available from various user groups to allow rebuilding the desktop without losing the comments.

The dual use of text files is becoming confusing, and I expect to change it in a future version, making options files a private type. Another possibility is to make Open only open concordance and options files, and introduce New as the way of constructing a new concordance from a different kind of document.

1.26 to 1.5 alpha

(Note: this version was not fully debugged and distribution was very limited.)

• Extended the program to support interlinear documents as produced by *IT*. Added some new options to control things unique to interlinear documents. The display formatting and clipboard options available in this program go considerably beyond those provided by *IT* itself.

• The "secondary file" options are now most naturally used in relation to a secondary *field* of annotations, so the dialog boxes and documentation now refer to secondary fields instead of files. However, the old capability is still provided; it is now described as simulating an annotation field with a second text file.

• Some options that only affect the display, and do not require the concordance to be reconstructed, were moved to the Context dialog box, which was consequently renamed Display. The options moved will not reload correctly from old options files.

1.5 aplha to 1.51 alpha

(This version was also not fully debugged and not widely distributed.)

• Changed sorting to use secondary order for words that are identical as far as the main sequence is concerned.

• Changed the program where necessary so that a "word" is consistently understood as the text between white space. (Previously, some parts of the program worked this way, while others considered a word to end at the first non-word character.)

- Fixed some of the worst bugs in 1.5 alpha.

1.51 alpha to 1.52 alpha

(This version was also not fully debugged and not widely distributed.)

• Implemented letter concordance for interlinear documents.

• Fixed more bugs.

1.52 alpha to 1.54 alpha

(This version was also not fully debugged and not widely distributed.)

• Worked to reduce the memory requirements for interlinear document concordances. Fixed several bugs.

• Made Options files a distinct, private file type instead of a text file. This removes the confusion caused by opening an options file and having it made into a concordance. Since Options files are now clearly identifiable to Conc, the Open Options command was removed from the File menu.

(Note: if you know how to use ResEdit or some similar tool to change the type of an old options file to CONO you can continue to use it. The internal format has not changed except as noted above.)

• Added the New command for creating concordances from text or interlinear documents. (Previously this was done using the Open command.) Accordingly, the Open command will now only open Concordance or Options documents.

1.54 alpha to 1.55 beta

• Added an ability to use pattern matching to select letters of interest for a letter concordance.

• Caused font selection for the concordance, text, and index windows to be saved in options and concordance files. (Font selections for the interlinear window are already saved in the model.)

• Changed some defaults: wrapping long lines, showing the keyword in bold face, and showing the secondary word in the concordance when a secondary field is in use, are now all done by default.

1.55 beta to 1.56 beta

• Changed the Wrap Long Lines menu option to bring up a dialog with various options for what to wrap and how to concatenate.

• Allowed a subset of options to be saved using the Save command.

• Made letter concordance pattern-matching work for text files.

- Made only the head letter be displayed in bold for letter concordances.

1.56 beta to 1.58 beta

- Fixed several bugs, including an inability to create a concordance from a text-only file since version 1.55, inability to open its own files when double-clicked in the Finder (since version 1.50), and some problems with printing indexes. The index window now works as advertised when using a secondary field in an interlinear document.

- Added the use of “word/...” or “.../secondary word” to clarify what is going on when an index is built while a secondary field is in use and sorting is done using just the main field word or just the secondary field word.

- Added the check box to allow context to be limited to the current unit.

- Margins specified in the page layout dialog box are now relative to the edge of the paper, as in *IT* and most other Macintosh programs. The layout of this dialog box has been changed to conform to that used in *IT*.

1.58 beta to 1.63 beta

- Added a progress indicator and cancel button for exporting files.
- Inapplicable menu items are dimmed.
- Removed New command and merged its function with Open.
- Added a Build menu with commands to build word concordance, morpheme concordance, letter concordance, and index.
- Added Append command to append text files.
- Export text window and selections of text.
- Added word separation character box.
- Added “Ignore distinctions within character groups” option to sorting.
- Added window tiling and zoom boxes.
- Arrow keys will now scroll the concordance and index windows.

1.63 beta to 1.68 beta

- Various bug fixes.