UNIVERSITY OF EDINBURGH
Edinburgh Regional Computing Centre

User Note 100

(March 1987)

Title:

INDEX: A Program to Simplify Preparing an Index for a Document

Author: John M. Murison

Contact: Your Support Team

Software Support Category: See Note 15

Synopsis

INDEX is a program, available on all EMAS systems, designed to simplify the task of preparing an index for a document.

Keywords

Document formatting, INDEX, LAYOUT, text formatting

© 1987 Edinburgh Regional Computing Centre

Introduction

INDEX is a program, available on all EMAS systems, designed to simplify the task of preparing an index for a document. The user must first prepare a file consisting of a series of index entries with associated page numbers, and INDEX then sorts the entries into alphabetical order. It can also generate a new input file, in a form which simplifies correction of the entries.

To obtain access to INDEX:

```
Command: OPTION SEARCHDIR=CONLIB.GENERAL      EMAS, BUSH

Command: SEARCHDIR ERCLIB:GENERAL             EMAS-A, EMAS-B
```

The program is then invoked by a command of the following form:

```
Command: INDEX input/ output, new input 1, new input 2
```

The details of the various files are given below.

Input file

Before INDEX can be used, an input file must be created; normally this is by use of a text editor. The basic form of each line of the input file must be one of the following:

```
string1\page no
string1,string2\page no
string1,string2,string3\page no
string1,string2,string3,string4\page no
```

Spaces following ',' or '\' are optional. The final line of the file must be 'zzzz' or 'ZZZZ'.

Notes

1) As indicated, up to four strings, separated by commas, can be specified on a single line. Each succeeding string is taken to be a sub-reference of the preceding one. Thus, 'files, use of' and 'files, creation of' each have the main string 'files' with different substrings. These would appear in the final index as

```
files
   creation of
   use of
```

2) A single page number reference should be given after the string or strings on each line. As indicated above, it is preceded by a back slash (\). Page numbers are usually made up only of digits, e.g. 23, but sometimes it is desirable to indicate a chapter or section, e.g. A-23 or 1.23. The permitted page number formats and their collation sequence are described below.

3) An input line can start with an asterisk (*). This means that the page number reference is treated as the main one for the index entry, and that page number is consequently given first in the list of page numbers in the index. Thus

```
files\ 23
files\ 72
```

would appear as

```
files       23,72
```

in the index, but

```
files\ 23
*files\ 72
```

would appear as

```
files       72,23
```

4) To save typing, parts of a line can be omitted in the input file. The last text given for the part missing is then assumed. For example, if one were preparing the input file by going through the document page by page, it would only be necessary to specify a page number once, subsequent input lines implying that page number if they have no page number specified explicitly. For example:

```
files, use of\ 23        and    files, use of\ 23
files,creation of                files, creation of\ 23
access to programs               access to programs\ 23
directories                      directories\ 23
```

are equivalent. Note that when a page number is omitted, the preceding '\' is omitted also.

Strings can also be omitted. The last specified string at that level is implied. Thus:

```
files\ 18                and    files\ 18
\ 75                            files\ 75
```

are equivalent. Apart from blank lines (which are ignored), every line of the input file has an associated first string and page number, whether explicitly given or implied.

Substrings can also be implied, but the comma following an implied substring must be given. For example:

```
files\ 18                and    files\ 18
,use of                         files,use of\ 18
,,online                        files,use of,online\ 18
,,offline\ 40                   files,use of,offline\ 40
,creation of\ 16                files,creation of\ 16
```

are equivalent. The relevant lines of the index would appear as:

```
files                18
   creation of       16
   use of            16
      offline        40
      online         18
```

(unless of course other lines of the input file had caused intervening entries in the index).

Note that once a string has been specified explicitly on a line, subsequent strings on that line *cannot* be implied.

5) Within a string, multiple spaces are reduced to single spaces, and spaces at the start and finish are ignored.

6) The symbol '$' in a string is treated specially. Rather than representing itself, it causes the *next* symbol to lose any special significance it might otherwise have. Possible uses include:

   $* at the start of the first string, to cause '*' to appear in the string rather than indicating a main reference (see Note (3) above).
   $\ in a string, to enable '\' to be part of the string rather than introducing the page number.
   $, in a string, to enable ',' to be part of the string rather than separating strings.
   $$ in a string, to enable '$' to be part of the string rather than marking the next symbol.

Page numbers

Page numbers can be presented to the program as any combination of letters, digits and other symbols (e.g. . / -), as well as the normal all-digit form. Thus `23a`, `A2-4`, `17/5b` are all acceptable page numbers.

For the purpose of ordering, a text string is derived from each page number, by making up each group of digits into at least three digits (by adding leading zeros), by leaving the letters unchanged, and by discarding the other symbols. The resulting strings are sorted into dictionary order, 0.....9 coming before A.....Z which comes before a.....z.

Roman numerals comprising the letters i, v, x, l or I, V, X, L are also permitted. They are assumed to refer to introductory pages and are therefore ordered before Arabic numerals.

Because of the unrestrictive nature of page number formats, it is possible to specify a 'page number' as a piece of text. Thus

```
ICL\ - see International Computers
```

in the input file would produce

```
ICL    - see International Computers
```

in the index.

Output file

The output file produced by INDEX is not the index itself, but a file which can be processed by the text formatting program LAYOUT to produce the index. The reason for having a two-stage process is that it enables the user to modify details of the format of the index; for example, the width of each total entry, the amount of indentation of substrings.

The collation sequence for index items is as follows: space, non-alphanumerics ('.' ':' ',' etc.), digits, letters. Initial non-alphanumerics in an entry are ignored for the purpose of collation, unless the entry consists solely of non-alphanumerics. Thus .OUT appears with the O's rather than before the A's.

Example

If a file IND has been prepared in the manner described above, then an index could be obtained from it by the following commands:

```
Command: INDEX IND/ INDL                          ('L' for LAYOUT)

Command: LAYOUT INDL,, X2700/ DRAFTD, FINALD      ('D' for document)
```

Note that LAYOUT is accessible via directory ERCLIB.GENERAL on EMAS and BUSH, and via ERCLIB:GENERAL on EMAS-A and EMAS-B.

If the user does not alter INDL before calling LAYOUT, then the following formatting conventions apply:

- Width of each entry: 2.9"
  Thus if the document line length is 6" or more, the index can be presented as two columns per page. Note however that the use of INDEX results in a *single* column - it would be necessary to use scissors and glue to place two columns side by side. (This may seem rather primitive, but in my experience it is the simplest way of achieving the desired effect.)

- Tab positions: 0.25",0.5",0.75"
  This means that each successive substring of an entry in the index is indented by 0.25".

- Left margin: 0

- Font used: 0 (default). On most output devices this corresponds to a 12cpi or 10cpi fixed-pitch font. It can be changed by inserting a font selection command (e.g. $02) at the start of the LAYOUT input file.

- Output device assumed: line printer (default). The user can specify the required device as the third parameter to LAYOUT (see example above), or by use of an entry point implying a specific device (e.g. DPLAY for GP300 printer output).

New input file, ordered alphabetically

Once an input file has been prepared and processed by INDEX, and the result processed by LAYOUT, the user normally finds errors or omissions in his index. Omissions are easily handled, by suitable additions to the INDEX input file, but errors can be less easy to correct, since the references are reordered by INDEX and the input line in error can be difficult to find.

For this reason a second output file can be generated by INDEX. It contains the same information as the input file but the order of presentation is that of the index (i.e. alphabetical). The example below should clarify this.

Notes

- The production of a new input file by INDEX is optional. It is only carried out if a third file is specified when INDEX is invoked (`'new input 1'` in the command description).

- Within the new input file, text strings are omitted where possible (i.e. if the same string or substring is already defined). This form has been found to be most convenient since, for example, a mis-spelt reference only appears once and therefore only needs to be corrected once.

New input file, ordered by page number

The new input file described above is useful when an index for an unchanging document is being developed, but not when material is being added to or removed from the document. In this case the page numbers of the document may have to be changed, and it is therefore better to have the index file entries in page number order. INDEX will generate such a file if a fourth filename ('new input 2' in the command description) is specified when it is invoked.

Notes

- Either, neither or both new input files can be generated by INDEX, as required by the user. Where the version ordered by page number is required but the alphabetically-ordered one is not, the call of INDEX has the form

```
Command: INDEX(input/ output, , new input 2)
```

- Unlike the previous new input file, in this file no text strings are omitted.

Example

A test input file:

```
Kangaroo\ 22
first string,second string\A4-10
Kangaroo\ 50
*Kangaroo\13
  Kangaroo\9-62
Kangaroo\    xiv
first string\7-10
first string,second string\27
,,third string
,2nd string,3rd string\18-4
Kangaroo\77
*new first string,second string
zzzz
```

If the above file were called TESTIND, and the following commands were given:

```
Command: INDEX TESTIND/ TESTINDL, NTESTIND1, NTESTIND2

Command: LAYOUT TESTINDL/TESTINDD
```

then the files involved would be as follows:

TESTINDL, the file produced by INDEX:

```
$a LINE=2.9"
$a TAB=0.25",0.5",0.75"
$b1 first string$r0 7-10
$b0$t1 2nd string
$b0$t2 3rd string$r0 18-4
$b0$t1 second string$r0 27,A4-10
$b0$t2 third string$r0 27
$b1 Kangaroo$r0 13,xiv,9-62,22,50,77
$b1 new first string
$b0$t1 second string$r0 77
$E
```

TESTINDD, the file produced by LAYOUT:

```
first string              7-10
  2nd string
     3rd string           18-4
  second string       27,A4-10
     third string           27

Kangaroo 13,xiv,9-62,22,50,77

new first string
  second string             77
```

NTESTIND1, the 'new input 1' file produced by INDEX:

```
first string\7-10
,2nd string,3rd string\18-4
,second string\27
,\A4-10
,,third string\27
*Kangaroo\13
\xiv
\9-62
\22
\50
\77
*new first string,second string\77
zzzz
```

NTESTIND2, the 'new input 2' file produced by INDEX:

```
Kangaroo\ xiv
first string\ 7-10
Kangaroo\ 9-62
*Kangaroo\ 13
first string, 2nd string, 3rd string\ 18-4
Kangaroo\ 22
first string, second string\ 27
first string, second string, third string\ 27
Kangaroo\ 50
Kangaroo\ 77
*new first string, second string\ 77
first string, second string\ A4-10
zzzz
```