View Index

ViewIndex

for the BBC Microcomputer

ACORNSOFT

Contents

Introduction

ViewIndex is an automatic indexing system designed especially to be used in conjunction with the VIEW word processor.

When VIEW is used to create substantial documents, particularly in education, business or professional fields, an index of the contents is often required. To prepare this manually is a laborious and often repetitive task, particularly when alterations are made.

ViewIndex searches through text files on disc, automatically collecting selected words or phrases to be included in the index, and placing them together with references in the desired order and format in a VIEW text file.

If required, several indexes may be merged to create a single overall index.

An index may be based on page number or section number.

There are four simple stages in the creation of an index using ViewIndex:

– Preparing the text file (or files) containing the document to be indexed, by using VIEW in the normal way, but with words or phrases to be included in the index indicated in a special way.
– Creating an intermediate index file on disc, by using VIEW with the ViewIndex printer driver. This file is a list of index entries and references as they were found.
– Using the ViewIndex sort program, which converts the contents of the intermediate index file into a completed index text file in the desired format.
– Editing the index, using VIEW to add any modifications or refinements that may be required.

This guide describes how each stage is carried out, and example files are included on the disc.

1 Preparing VIEW text files for ViewIndex

1.1 The template

When creating an index for a document, you must specify which kind of index references you are going to use. These may be either page numbers or section numbers (like 4.7.1). ViewIndex can index a document using either type of reference.

A template, or pattern, allows you to specify which type of index references you are going to use, The template must be inserted at the beginning of the first text file or form the first text file itself. The ViewIndex disc includes two files containing ready-made templates which allow you to index either by page number or by section number. These files are called TPP and TPS respectively and can be used when required. Their use is fully explained later.

The following section is for those who wish to know more about how templates are designed and who may wish to design their own.

1.1.1 Designing a template

If you design a template, place it before the text files to be indexed. The template can form the first text file itself or can be inserted at the beginning of the first text file.

A template consists of number registers which form the pattern for the type of reference you want to use. Number registers are indicated by a vertical bar (¦) before a letter. For example, ¦ **P** means that **P** is a number register rather than an ordinary letter. For a more detailed explanation of number registers, see the *VIEW Guide.*

The following shows you step by step how the templates included on the ViewIndex disc were designed.

1.1.2 Indexing by page numbers

A template may contain up to five number registers, but if you are indexing by page numbers then only one is necessary ie ¦ **P**, where P stands for page.

In order to enter the template into a VIEW file:

–   Enclose the template within two highlight 2 characters.

```
*¦P*
```

–   Follow the template by your chosen filename (eg I.INDEX).

```
*¦P* I.INDEX
```

–   Surround the template by the following sequence of edit commands, so that ViewIndex can recognise it as a template. Edit commands are explained more fully in the *VIEW Guide*.

```
HT 2 149
    *¦P* I. INDEX
HT 2 129
SR P 0
PE
```

The first line sets highlight 2 to 149.

The second line consists of the template enclosed by highlight 2 characters, followed by the filename to be used for the intermediate index (in this case I.INDEX.)

The third line resets highlight 2 to its normal value.

The fourth line resets a register, in this case **P** for page indexing.

The last line is a page eject, which resets the line count which has been altered by the template.

1.1.3 Indexing by section numbers

If your indexing is based on section numbers, then the template will contain more than one register. In this case, place separators between registers. In the template ¦ **C . ¦ S . ¦ T** fullstops are used as separators but any character except a comma can be used.

In order to enter the template into a VIEW file:

–   Enclose the template within two highlight 2 characters.

```
*¦C.¦S.¦T*
```

– Follow the template by a filename (eg I.INDEX).

```
*¦C. ¦S. ¦T*I.INDEX
```

– Surround the template by the following sequence of edit commands, so that ViewIndex can recognise it as a template. For a more detailed explanation of edit commands see the *VIEW Guide.*

```
HT 2 149
  *¦C.¦S. ¦T* I.INDEX
HT 2 129
SR C 0
PE
```

The first line sets highlight 2 to a value of 149.

The second line consists of the template enclosed by highlight 2 characters, followed by the filename to be used for the intermediate index. (Here I.INDEX.)

The third line resets highlight 2 to its normal value.

The fourth line resets the chosen chapter number register.

The last line is a page eject, which resets the line count which has been altered by the template.

1.2 Setting up section numbers

If you have chosen to use page numbers in your index, you do not need to read this section

If you are setting up section numbers you need to tell VIEW when each chapter, section and subsection begins. You can do this in the following way:

Before each chapter:

– Press EDIT COMMAND **RETURN** IC **RETURN**

Before each section:

– Press EDIT COMMAND **RETURN** IS **RETURN**

Before each subsection:

– Press EDIT COMMAND **RETURN** IT **RETURN**

Thus, when indexing by section number, the document text would have this sort of format

```
     Text.................
     .....................
IC                                     New chapter
     1.0.0                             Chapter reference
     .....................
IS                                     New section
     1.1.0                             Section reference
     .....................
IT                                     New subsection
     1.1.1                             Subsection reference
     .....................
IS                                     New section
     1.2.0
     .....................
IT                                     New subsection
     1.2.1
```

In fact IC, IS and IT are macro calls, which cause each type of reference to be incremented. Macros are used since they provide the easiest way to increment section numbers. (The use of macros is explained fully in the *View Guide*.) In this case, the following three macros have been used:

```
DM IC           Macro name
SC C ¦C+1       Increment chapter register by 1
SR S 0          Set section register to zero
SR T 0          Set subsection register to zero
EM              End macro

DM IS           Macro name
SR S ¦S+1       Increment section register by 1
SR T 0          Set subsection register to zero
EM              End macro

DM IT           Macro name
SR T ¦T+1       Increment subsection register by 1
EM              End macro
```

If you are using your own macros be sure to include them, together with the template information, at the start of the document text. However this is not

necessary if you are using the above three macros since they are included in the file TPS, which also contains the template necessary for indexing by section number.

1.3 Preparing the document text

A document which requires an index is usually substantial; too long to fit into memory and comprising a large number of pages. You can create such a document with VIEW using two different methods: as a single large text file using the EDIT function or as a set of separate independent text files which you then print in sequence to produce the final document.

ViewIndex will work on documents created by either of these methods, but there must be sufficient room on the disc to create an intermediate index file and the total length of all filenames to be processed at one time must not exceed 100 characters (including separating spaces).

ViewIndex will create indexes with references based on document page numbers or section number depending on your requirements. In either case you must first identify the text entries to be included in the index by enclosing them in special codes.

1.3.1 Creating index entries in text files

Within the document text, you must identify the words, phrases or other item to be included in the index by enclosing them within double highlight 1 codes so that ViewIndex can identify them, as shown below:

```
The __VIEW__ system for word processing
```

The word 'VIEW' has been identified as an index entry.
Note that several highlight 1 codes appear as a single line on the screen.

Highlight 1 codes (ie _) are usually used to identify text to be underlined, the first code indicating the start of underlining, the second indicating the end of underlining. Double highlight 1 codes have no effect on printing the document but are used by ViewIndex to identify the start and end of index entries.

Because of the 'on-off' effect described above, you can still use highlight 1 codes normally for underlining. In the above example if the word 'VIEW' were to be underlined on printing as well as included in the index you would use six highlight 1 codes as shown below:

```
The ___VIEW___ system for word processing
```

The first two highlight 1 codes are found by ViewIndex and identify the start of an index entry but have no effect when the document is printed. The third single code has no effect in ViewIndex but will start text underlining when printing. The fourth code will stop underlining when printing and the fifth and sixth codes indicate the end of the index entry to ViewIndex.

The same principle applies if the index entry and underlining required do not correspond exactly in the same piece of text. Suppose 'VIEW' were still to be an index entry but the phrase 'VIEW system' were to be underlined, the text would have these highlight codes:

```
The ___VIEW__ system_ for word processing
```

As another example suppose that both the word 'VIEW' and the phrase 'word processing' were to be index entries and the whole text were to be underlined, the highlight codes would look like this:

```
_The __VIEW__ system for __word processing___
```

The first, code starts the underlining, the double pairs of codes around 'VIEW' and 'word processing' define them as index entries, and the last code turns off underlining.

Index entries may be words or phrases, and may carry over between lines or pages. You must define all occurrences of an entry to be included in the index using pairs of highlight codes as described. ViewIndex will merge all exactly similar entries into a single-line index entry with several references in the completed index.

You may highlight words or phrases to be included as index entries whilst creating the document, or do this after the document is finished. In the latter case, you can use 'VIEW's SEARCH facility to find all occurrences of a word or phrase. To do this, go into command mode and type, for example:

```
SEARCH/biodegradable foodstuffs/
```

The cursor will move to the first letter of the first occurrence of 'biodegradable foodstuffs' in the text. You can enclose the phrase in double highlight 1 codes (as described above). Then press NEXT MATCH (**CTRL** *F1*), which moves the cursor to the next occurrence of the phrase.

1.3.2 Length of index entries

The maximum length for an index entry using ViewIndex is 50 characters. However, the practical maximum length to use depends on the document and output format that you require when producing the final index. If you are using a full page width layout there will be plenty of space for index entry and references; if the final index is to be structured as several columns per page then the maximum index entry length used in the document should be limited accordingly.

1.3.3 Extended highlight sequences – a special case

If you have enclosed an index entry in double highlight 1 codes and you are also going to enclose it in highlight 2 codes for printing purposes (usually to produce bold text) you must either break up the sequence by leaving a space after the first highlight, 2 code, as shown below:

```
The * __VIEW__* system for word processing
```

or use the codes in the following sequence:

```
The __*VIEW__* system for word processing
```

This is because ViewIndex interprets unbroken sequences of highlight codes which begin with a highlight 2 code as extended highlight sequences (used by some printer drivers) and ignores them.

If you do not use the space or specific code order this could result in the index entry search getting out of phase and ordinary text being taken as index entries.

1.3.4 Reversing the roles of highlights 1 and 2

If you want to enclose words or phrases within highlight 2 (*) rather than highlight 1 (_) characters, this can easily be achieved by including the following edit commands in your template file:

```
HT 2 128
HT l 149
_¦P_I.INDEX    (for a page index)
HT 1 129
SR P 0
PE
```

8

In this case, the template is now enclosed within highlight 1 characters, and all words or phrases to be included in an index should be surrounded by double highlight 2 characters.

Note that when printing the text, as the roles of the highlights are now reversed, a phrase enclosed in highlight 1 characters will now normally be in bold type and a phrase enclosed in highlight 2 characters will be underlined.

2 Creating an intermediate index file

When you have inserted all the appropriate highlight codes in the document and added the template and macros, if designing your own, you simply create the intermediate index file.

Insert the ViewIndex disc and load the ViewIndex printer driver called 'INDEX' with the command:

PRINTER INDEX **RETURN**

This sets up VIEW to create an intermediate index file instead of printing a document, so a printer need not be connected.

Next, use the PRINT command with all the text files containing words or phrases to be included in the index:

`PRINT (templatefile) (textfile1) (textfile2)...`

where `(templatefile)` is either TPS, for section references, TPP, for page references, or your own file containing template and macros; `(textfile1)`, `(textfile2)`, etc, are VIEW text files.

It is important to PRINT all the text files for the index at one time so that the references will be correct. Only the first template read is used, any others are ignored.

ViewIndex processes all the text files in sequence, extracting the index entries and creating the intermediate index on disc, in a file with the specified name. The entries extracted are displayed on the screen as they are written to the intermediate index file.

If the display begins to show ordinary text rather than proper entries, it means that you have made a mistake in your highlight coding. Either an index entry has not been ended properly with double highlight 1 codes or highlight 1 and 2 codes are not in the correct sequence (see section 1.3.3). If this happens, identify the position of the error from the last correct index entry, stop processing by pressing **ESCAPE**, correct the error and restart.

10

The intermediate index file has a special format and cannot be used with VIEW.

3 The ViewIndex sort program

Once you have created the intermediate index file successfully, the document text files are no longer required for building an index. ViewIndex puts the extracted entries in alphabetical order and merges like items together (the intermediate file is simply a list of entries and references as they were found).

Load the ViewIndex sort program in the following way:

- Press **SHIFT**
- While holding down **SHIFT**, press and release **BREAK**
- Release **SHIFT**

A menu with four options is displayed. Option 1 allows you to create a final index from an intermediate index file. Option 2 allows you to merge up to four indexes into a single large index. Option 3 allows you to enter VIEW and Option 4 ends the program

3.1 Create index

When this option is selected a series of prompts is given about the format of the final index. The prompts are explained below;

```
Insert disc containing
intermediate index file
and enter filename:
```

Insert the disc containing your intermediate index file if this is not on the ViewIndex disc.

Enter the name of the intermediate index file you have created. If you simply press **RETURN** the default name of 'I.INDEX' will be assumed, which will be the name of your intermediate index file if you are using TPP or TPS.

```
Enter filename
of final index:
```

Reply with the name of the file to contain the completed index here. A default name of 'O.INDEX' is assumed if **RETURN** is pressed.

```
Ignore case ? (Y/N)
```

Your response to this prompt affects the way the index entries are sorted. If you enter 'N', the index entries will be sorted into order taking upper case and lower case characters into account; for example, 'SALT' and 'salt' will appear separately in the index. If 'Y' is entered, the case will be ignored and entries will be mixed together.

```
Justify reference numbers ? (Y/N)
```

The index format is affected by your response to this prompt, which controls where the references will be placed on an index line. If you enter 'N' the first reference will immediately follow the index entry; entering 'Y' will result in the reference(s) being right justified on the index line. If you enter 'Y' a further prompt is given requesting the line length to use for justification.

```
Line length ?
```

This must be between 10 and 80 characters. The shorter the line, the more lines will fit in the file containing the final index.

```
Should the first letter
  of each entry be:

  1.  upper case
  2.  lower case
  3.  unchanged?
```

The index entries in the intermediate index file may start with a mixture of upper and lower case. ViewIndex will automatically convert all the index entries to upper or lower case first letter, or leave them as they were originally, depending on which option you choose here.

After you have answered the prompts, the maximum number of index entries that can be processed (which depends on the line length chosen) is displayed, and the program begins processing the intermediate index file. A prompt will appear at the bottom of the screen: `Insert disc to contain final index.` You can insert another disc if you want to or continue to use the same

disc. There are four stages to the program and the following messages are displayed as each stage is reached:

Reading index
Sorting index
Merging multiple entries
Writing sorted index
Finished

The total number of entries in the final index is also displayed.

3.1.1 Index format

Examples of the two types of final index format are given below:

Unjustified

```
....
....
Hangman 138
Leap Years 134,141
Man 171
Month 126
Persian 46,97,114,145,151,180,189
Polygon 39
....
....
```

Justified

The same index with references right-justified on a 30 column line would look like this:

```
....
....
Hangman                      138
Leap Years            134,141
Man                          171
Month                        126
Persian         46,97,114,145,
    151,180,189
Polygon                       39
....
....
```

14

If the references exceed the line length, they continue on subsequent lines preceded by four spaces.

3.2 Merge multiple indexes

This option allows you to merge up to four existing indexes of a similar format to produce a single overall index. For example, you may have prepared a document as separate sections, each with its own index. All files containing indexes to be merged must be on the same disc. The prompts are described below:

```
Insert disc containing
files to be merged, and
enter filenames:
```

Insert the disc containing the index files to be merged if they are not on the ViewIndex disc, then enter their filenames in the table displayed. In order to edit the filenames, you can move the cursor up and down by using the arrow keys (↑ and ↓). When you have entered the filenames, press **RETURN** until the cursor leaves the bottom of the table. A second prompt will now appear.

```
Enter output filename:
```

You should enter the name of the file to contain the overall index here. If you simply press **RETURN** the default name of 'M.INDEX' will be assumed.

The index files are then merged to produce the overall index.

Merging the two indexes below would give the final index shown overleaf.

```
Index 1                 Index 2
....                    ....
....                    ....
Leap Years 134,141      Leap Years 164,181
Man 171                 Man 191
....                    ....
....                    ....
```

Overall index after merging

```
....
....
Leap Years 134,141
         164,181
Man 171
    191
....
....
```

3.3 Editing the final index

The final, sorted index is a VIEW-compatible file, and can be edited or printed out by VIEW like any text file. However, the size of the file depends on the number of entries, whether the entries have been right-justified, and the chosen line length It may be too large to fit into memory, in which case, the EDIT function must, be used to examine or alter it.

The completed index, which must have heading and formatting commands added as needed, should he examined before printing as there are instances where the index may need to be tidied up, such as when zeros occur in section references.

For example, the reference '2.0.0' may occur, which indicates that an entry is at the beginning of chapter 2 but before section 1 and subsection 1. ViewIndex does not remove the zeros in case you actually wish to have this kind of reference. You can use VIEW's REPLACE command to delete the zeros.

Type:

```
REPLACE/.0.0/    /
```

This causes the cursor to pause at each occurrence of **.0.0**, replacing it with four spaces if you press 'Y'.

Also, if a series of related index entries beginning with the same word or phrase are grouped together, you may wish to alter all entries after the first into subentries.

Another possible case is where an index entry has split across page boundaries and the page number has been inadvertently incorporated into the entry.

If a final index with more than one column per page is required, it must be created by physically cutting up the final index printout and creating a new layout. This may mean the pagination of the index needs to be altered before printing, to fit the final layout.

4 Example of an index

Included on the ViewIndex program disc are three VIEW files called 'TPS', 'TPP' and 'TEXT'. Files 'TPS' and 'TPP' are ViewIndex template files which you can use in conjunction with text file 'TEXT' to produce an example index. File 'TPS' creates intermediate index file 'I.INDEX' and produces an index based on section numbering; 'TPP' creates 'I.INDEX' and produces a page number index. File 'TEXT' contains part of the text used to produce this guide with the page length and text altered slightly. Several phrases have been enclosed within double highlights for inclusion in the index.

Examine these files with VIEW and use them to try out the ViewIndex system. Try altering the index entries in the text and producing a slightly different index, or inserting a new subsection anywhere in the text and see how the reference numbering is automatically updated both in the document text and index. You can also try out the different index format options.

1. Type `*WORD` to enter VIEW and load the ViewIndex printer driver 'INDEX' by typing:

```
PRINTER INDEX
```

2. Use the `PRINT` command to create an intermediate index:

```
PRINT TPP TEXT
```

or

```
PRINT TPS TEXT
```

3. Start the ViewIndex sort program by using **SHIFT** **BREAK** as described in chapter 3. After the first prompt, press **RETURN** which gives the default intermediate index filename I.INDEX. After the second prompt press **RETURN** once more, which gives the default final index filename O.INDEX. Then respond to the questions which follow, depending on the index format you require.

18

4. Select the VIEW option and examine the final index, which will be called 'O.INDEX'

5 Error messages

The following error messages may occur when using the ViewIndex program

`File not found`	The file specified cannot be found by the ViewIndex program.
`Incorrect template`	The template defined before the document text contains no number registers. It should contain between one and five.
`Too Long`	This error message is displayed if an index entry of more than 50 characters is found by the sort program; the message is followed by the first 50 characters of the entry.
	If the entry displayed is not one of the words or phrases which you highlighted but is part of the rest of the text, then the two most likely mistakes are:
	– Underlines have been typed instead of highlight 1 characters. – A highlight 2 is immediately followed by a highlight 1 character and not separated by a space. (See section 1.3.3.)
`Too many references`	The index entry named has too many references.

Other error messages can be found in the *BBC Microcomputer System User Guide* or the *Disc Filing System User Guide*.

Index

22