Embedded indexing

James Lamb

Microsoft® Word embedded indexes have never really made much impression on the world of professional indexing. They are lots of work and have limited functionality and so clients have often been persuaded that they don't really need them. But this may be about to change.

When the latest version of Microsoft® Word was released, a function, unnoticed by some, was the ability to export XML files. Furthermore, Word's embedded indexing tags are successfully included in the XML file. Microsoft is starting to push Office/XML as the front-end to almost everything, and so publishers, many of whom are developing XML-based systems, will start to look at embedded Word indexes as a preferred medium for transporting index information.

In any case, embedded indexes can have advantages and can be worthwhile, if creating them is sufficiently easy. Embedded indexes are not complicated, and in today's world it would be wise for any indexer to have an understanding of the issues.

This article explains simply how an embedded index works; the advantages and limitations of embedded indexes; how to create an embedded index using Microsoft Word, and briefly describes the software available to make embedding practical.

What is an embedded index ?

In a conventional index each heading has a collection of locators indicating the locations to which it refers. In an embedded index this is reversed – each location has a collection of headings to which the location is relevant. These collections of headings are not usually visible in the document, but can be made visible, and so are regarded as being buried beneath the surface of the document – in other words, embedded. Neither are the headings usually visible in the printed version of the document, but on request the computer program will use the locations of these embedded headings, combined with the current pagination, to build a conventional index. This ability to build the conventional index on request provides the advantage of an embedded index – if changes are made to the document which affect the pagination, then the program can rebuild the index to match the new pagination.

Advantages of an embedded index

From the publisher's point of view this means that a text can be indexed and issued in one format, say hardback, and repaginated to a different format, say paperback, without the cost of reindexing. The new format could be completely different, such as changing the text into HTML for publishing it online or issuing it on CD-ROMs. This is referred to as *re-purposing*. In this case, the format of the final display index may even be changed. Rather than having page numbers, it could have hypertext links: simply click on the link and you are taken straight to the location in the text.

Furthermore, because the indexing information is stored in the document itself, documents can be cut up and reassembled. For example, a publisher might take key chapters from a book to issue a condensed version (as with Gibbon's Decline and Fall of the Roman Empire), may have extra chapters added to cater for new developments in the subject matter, or may even cull chapters from different books to create a compilation book. There has been concern expressed recently in the United States, and previously elsewhere, about the weight of schoolbooks that children must carry. Compilation books tailored to specific courses or schools could provide an answer.

Another advantage, which affects everyone in the book production process, is that of timing, referred to as workflow. The creation of the index need no longer wait until the final page numbers are assigned. Indeed, even processes that affect the content of the material, such as copy-editing, can be carried out in parallel to the index creation.

Disadvantages of an embedded index

This all comes at a price – the editing of an index is dependent on context, and with an embedded index, the indexer cannot see the final index to know that context. For example, if locations are close to each other in the text, a page break occurring between them will result in two locators in the final index, but under a different pagination they might appear on the same page, resulting in a single locator in the final index. A string of seven locators should be broken up using subentries, but in one pagination the heading may have four locators, and in another pagination eight – the indexer cannot know and so cannot get it right for all possible paginations.

There are more problems if documents are cut up and reassembled, as described above. Headings can be dependent on the overall subject of the book in which they appear: for example in a biography the heading 'education' would refer to the education of the biographee, rather than education generally. Synonyms can result in incomplete sets of locators, say, some under 'farming' from one book and some under 'agriculture' from another. Cross-references, unnecessary in the individual books, become essential in the compiled book; contrariwise, cross-references that are present might no longer lead anywhere, as their target might

have been removed from the current text. It is likely, however, that few people other than indexers, will understand these problems, or that that the benefits will be seen as overwhelmingly advantageous, so the use of embedding is set to rise.

Why Microsoft® Word?

Getting practice in embedding is not difficult – most of us will already have the software necessary. The free OpenOffice word processor, for example, has all that is required, but Microsoft Word will be the software most commonly requested by clients. Where a publisher does not provide a Word version of the proofs, provided you have an electronic copy, say a pdf, it is very simple to create a Word copy with the correct pagination.

How to embed in Microsoft® Word

To explain the detail of embedding in Word, I shall use a familiar piece of text, a portion of *The Walrus and the Carpenter* by Lewis Carroll, but have a page break between each stanza. Figure 1 shows the text with its final embedded index entries displayed. If you were to press the (Show/Hide) button, these embedded entries would disappear from view, leaving the plain text.

Point locators

The simplest type of entries is those that point to a specific point in the text. In reality, of course, most indexes entries refer to more than just a single character, but for names in scholarly references, for example, this may be sufficient.

At the end of the poem there is a reference to the author, and I want to have an index heading 'Carroll, Lewis' which has a locator that refers to this page. In a conventional index I would create a heading and find the page number on which the name appears. For an embedded index, I need to attach the heading to that location in the document. In Word, the format of this embedded index entry will be: {XE "Carroll, Lewis"}. This is known as an XE, or index entry, tag.

To create an XE tag:

- Select the text 'Lewis Carroll'.
- From the menu choose Insert> Field.
- Select Field Name 'XE' from the very bottom of the scrolling list and click the button labelled 'Mark Index Entry'.
- In the 'Main Entry' box you will see the text selected, 'Lewis Carroll'. You need to invert this, so retype the text as 'Carroll, Lewis'.
- Click the button 'Mark'.

The 'Mark Index Entry' dialogue box will stay open, and the XE entry will show in the text, immediately following the text highlighted.

This might seem like a tremendous hoo-ha in order to create one index entry, but that is simply because it is. Now that you have an index entry, however, you can edit its contents and copy and paste it just like normal text. So a much easier way to create the next entry is to copy, paste and edit this one.

Cross-references

I also want to have a cross-reference appear in the index from the author's real name and so we create an entry {XE "Dodgson, Charles Lutwidge" \t "see Carroll, Lewis"}. The text in quotes can be anything you wish, so you can use *See* or *see also* or *see under* etc., and there is no validation to force, or ensure, that the preferred heading actually exists.

Cross-references are not linked to specific places and so can be placed anywhere in the text. It is sometimes thought that it is good to keep the cross-ref as close to the original entry as possible – on the basis that, if the text is subsequently modified and the text containing the 'Carroll, Lewis' entry deleted, then the cross-ref would be removed too. In practice, however, most index headings will appear in multiple places in the document so this doesn't work, and it is better to keep all the cross-ref's together, so they can be found more easily.

Range locators

As we said, most index entries actually refer to a passage of text. Often, when indexing conventionally, however, we don't think about that: we simply identify the passage, often a paragraph, as starting and finishing on the same page and give it a single page number. With embedded indexing, however, we cannot be sure where the pages start and end, so most entries should be ranges.

In order to create an entry referring to a passage we must first create a bookmark to define the text range. To create the bookmark:

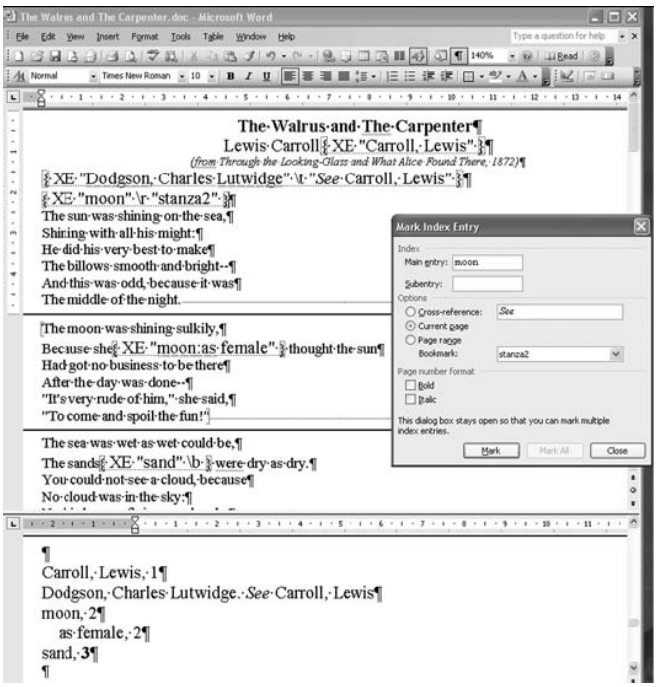

Figure 1

- Select the text, say stanza two, by dragging, multiple-clicking etc.
- From the menu Insert> Bookmark
- Type in a name for the bookmark, say 'stanza2' – this must be unique (but be careful not to create names which might get in the way of any naming scheme which your client might be using)
- Click the button 'Add'

Now that we have created the bookmark we can create the index heading referring to it: {XE "moon" \r "stanza2"}. When this appears in the index, if the entire range appears on a single page in the current pagination, then a single page number will be shown, but if the start and end are on different pages, then a page range will be shown.

Given the significant extra work in creating range locators it is tempting to use point locators instead, particularly when the range is small and sited comfortably in the middle of a page. However, subsequent editing can add in more text, tables, illustrations or even footnotes which might cause the passage to move, so an embedded index should have ranges for each topic, no matter how small, otherwise the full flexibility, and therefore some of the benefit, of embedding is lost.

Italics and bold

For including italics, bold and special characters, in the headings themselves, simply include them in the index entry and they will appear in the index. For font colours, the colouring of the entry in the index is taken from the first occurrence in the text, so if subsequent entries are coloured differently they do not create different entries. Font sizes are not taken through to the index entries.

For the locators, only italics and bold are allowed, by including an \i or \b flag in the XE field: {XE "sand" \b}. There is no mechanism to do anything else, such as adding a suffix *ill.* to indicate an illustration, for example.

Subheadings

Including subheadings is straightforward, separating the heading from the subheading using a colon, so: {XE "moon:as female"}. Up to 6 levels of subheadings are allowed.

Sorting

The final thing we may wish to do is to override the default sort. Word defaults to word-by-word, but you may require letter-by-letter. Even if you never use letter-by-letter, you will want, at some time, to put subheads in chronological order or to sort 'McDonald' under 'Mac'. Although this is not to be found in the help texts, this is done by appending the sort term to the heading, separated by a semi-colon, so: {XE "McDonald;MacDonald"}.

Adding the index

Once all the XE entries have been created, the index itself must be added. This is done by using the menus: Insert> Field and selecting 'Index' from the scrollable list. This brings up a window with some parameters which allow the layout of the index to be tweaked. If the index entries are changed in any way, or the pagination is changed (for example, by opening the document on a machine other than the one on which it was created), then the index should be rebuilt by right-clicking on it and selecting Update Field.

Multiple sequences

If we wish to have more than one index sequence in a document, perhaps Authors and Subjects, then each XE should have a /f flag added with a single character identifier. So: {XE "Carroll, Lewis" \f "A"} and {XE "moon" \f "S"} and then including the \f "A" in the index tag will include only those entries generated from XE tags also containing the matching flag. Entries without a flag are treated as having the flag '1'.

Techniques

This, then, allows you to do everything which can be done with Word indexes, although the process is rather cumbersome. Word's answer is to provide the ability to create entries for all occurrences of a text throughout the document with a single click (using 'Mark All' rather than 'Mark' in the sequence under Point Locators above), and, one stage further, to pre-load all the index headings in a separate automark/concordance file which will then automatically 'Mark All' for each heading.

Real Indexing

The above is quite practical for producing indexes, *provided* that you know exactly what is to be the final text of the index heading at the time you create the entry. This does happen, when producing an index to some form of catalogue, where product titles are clearly defined, but for most indexing, involving human evaluation and judgement, this would be unusual. Professional quality indexes require an editing pass, where the index entries can be made consistent and tailored within their context in the final index, which may amount to as much as one third of the overall work. To change an index heading in an embedded index involves changing the XE tag at every location at which that heading appears in the text – even with careful use of global replaces this is still very time consuming and a major distraction from the process of editing.

Software

One might have expected there to be a plethora of software to help make this easier, but there is actually very little. It would seem that it is only indexers who see it as inadequate.

First there are the standard indexing packages such as SKY and CINDEX. With these you create your index within the package, using page numbers from the current document pagination. This has the advantage that you can use the software with which you are familiar, with their auto-flipping, spellchecking, shortcuts etc. which can so speed up data entry. You can then do all the editing of the index in the package and only embed it into Word when complete.

To embed the index into Word then, you display your

index in page number order and, moving through your index and the document in parallel, drag the heading from the indexing package into the correct location in Word. This pastes the XE field into Word. Both packages will include the sort override information and SKY will take the bold/italic locator flag too. Thus there is at least one drag operation for every location (average 3 per heading?). A major drawback with this technique is that ranges must still be named and created manually, and then the XE entry, after it has been dragged into place, must have the /r information added manually.

Next, IndexAssistant is a utility, costing $10, which improves on Word's built-in dialogue window, making easy those editing functions which otherwise require playing around with global replaces, and, significantly, making creation of range locators faster too.

The best known software is probably DEXter, costing $199.95 [see Jon Jermey's review in *The Indexer*, Vol 24, No. 3, pp. 167–8, and Frances Lennie's letter on page 223 of this issue – *Ed*], which gives you a separate, tabular index entry window, where the index is created, a fully formatted display of the index to work with, and, only when the index is finished are the entries actually embedded. Dexter provides functionality only usually included in the stand-alone index packages, such as automatically swapping 'Apples: in pie' around to become 'Pie: apples in', handles easy creation of range locators, and sort options such as letter-by-letter.

My own contribution to the field is WordEmbed ($110: see page 251 for review), which uses a different approach altogether. By installing WordEmbed, selecting a range or point and pressing ctrl-shift-\ you are given a special locator to paste into your own, familiar indexing software, e.g. MACREX. (The first version of WordEmbed used Word's own page and line numbers as locators, but an 'undocumented feature' in Word meant that this is not unique – it is possible to have two places in one document with the same page and line number!) You then create and edit the index in your own software and once it is complete, press a button and the embedded entries and ranges are automatically created and entered into the Word document.

Why bother?

Embedding can be time consuming, but it is not really complicated, and, as we have seen, there are software solutions which can mean that it can be done with almost no extra effort and time.

Embedded indexes in Word are not suitable for documents which require specialist locators, but, having said that, there are an awful lot of books out there which do only need simple/bold/italic page numbers.

Even when the client is not demanding an embedded index, if there is a risk of last minute changes, the indexer may wish to embed for their own protection. But, more and more, the client will be demanding embedding and, more and more, Word will be the medium of choice. It would be wise to investigate the options now, so that, when that request comes, you have an informed answer.

References and acknowledgements

The examples here use Word 2003.
IndexAssistant, US$10: for ordering see
`http://jambient.com/indexassistant/`
DEXter, US$199.95: for ordering see
`http://www.editorium.com/dexter.htm`
WordEmbed, US$110: for ordering see
`http://www.jalamb.com/wordembed.html`

James Lamb has a degree in Computer Science and Mathematics from London University, worked for over 20 years as a senior IT technician and team leader, much of that time for dealing rooms of international banks, and became a full-time, professional indexer in 2004. He is DBA/Developer of the ASI Web-Indexing SIG and an Accredited Indexer with the Society of Indexers (SI). Email: `james@jalamb.com`