Jumping on the embedded indexing bandwagon – or should I?

Lucie Haskins

Do you need to master embedded indexing before your clients ask for it? And what are the pros, cons and practicalities of doing so? Lucie Haskins, who teaches and practices embedded indexing, offers an authoritative overview.

The seeds for this article came from an indexer's email inquiry on the importance of learning how to index in InDesign (a popular desktop publishing program). Because I felt her question needed a thoughtful and comprehensive answer, I responded from the more global perspective of getting on the embedded indexing bandwagon (and all that involved) while also recognizing and addressing the 'why bother now' aspect. (This response also provided the basis of my 2015 presentation to ISC/SCI conference attendees.)

What's the big deal with digital publishing?

There has been a lot of talk about digital publishing in the past few years. We hear the market is exploding for ebooks. That they will overtake print books and bookstores will go out of business. That libraries will go the way of the dinosaurs. That indexes (and indexers) are no longer needed because online search does the job so much better.

Who and what are we indexers to believe? Is there any hope for us or should we all be looking for careers outside of publishing? Is there any way we can prepare for the changing times or have we already missed the boat?

It's definitely a scary time if we listen to all the hype. But as far as I have observed, while digital publishing has gained momentum, it has not replaced print publishing. Nor do I expect it to.

However, it's always a good practice to reflect on how we might want to sharpen or augment our services and skill sets. And this changing environment provides us with a great opportunity to reflect on where we each might fit in this brave new world.

Why should I bother with embedded indexing now? Will I be left behind if I don't?

Have clients expressed an interest in digital publishing? Does any of this interest you?

If you didn't answer yes, then consider the importance of understanding this emerging arena in order to guide your clients when they come to you with questions. Even if you are not interested in learning embedded indexing yourself, wouldn't it be helpful to have enough knowledge about it to point your clients to others who can help them?

I have been working with embedded indexing products (FrameMaker, Word, InDesign, HTML and XML) since I began indexing back in 2000. The majority of my embedded indexing work has been for computer book publishers, with a few outlier clients in other specialty areas. I also do a lot of back-of-the-book (BOB) indexing. My BOB indexing clients provide me with projects in business, cookbooks, math, economics, statistics, travel, memoirs and history.

Whenever I hear about the trend toward ebooks and the need to jump on the embedded indexing bandwagon, I think of my long-term BOB indexing clients and try to visualize them moving toward ebooks and embedded indexing. I already have them as a BOB client, so it would be an easy transition for them. They would already be comfortable with my indexing abilities and professional competence. We would just be changing to a different indexing process and delivery system.

But even with that relationship securely in place, I can't imagine it, at least in the near term. These clients of mine are so established with their processes that it would take a major shakeup to get them out of their comfort zone. That shakeup may be coming but they aren't budging right now.

I know that learning new software and adapting work methods to fit new software constraints can be quite involved. So, I can understand why people would say 'Why bother now?' Perhaps the writing is on the wall and those who don't learn will be left by the wayside. But there are so many hurdles to overcome when entering this arena that the decision to do so should not be made lightly.

I think indexers should only learn embedded indexing methods if they have clients who are making those grumbling noises and asking questions. One of the difficulties in learning something like embedded indexing is that the knowledge and training rapidly leaves us unless it is practiced regularly and consistently. If there are no clients on the immediate horizon, it seems like you are doing the right thing but prematurely. So why purchase an expensive product and attend time-consuming training when it might not pay off for a while?

At the same time I recognize that we are entering a new era and that ebooks will only grow in market saturation. And market saturation and consumer demand will drive more and more attention to ebooks and all things related to them. I hope that as things progress and evolve, the path will become clearer on how all this is going to shake out and what role embedded indexing will have in all this. But right now this still looks like the Wild Wild West to me, with so many formats and so many processes and so many solutions out there. How are we to untangle them and make sense of it all?

My Microsoft TIM (Tagging and Indexing Tool) indexing experience closely mimicked the embedded indexing issues we currently face. My clients didn't understand what they needed to do to provide their materials in the proper format. They didn't know what to do with the output I delivered. And they didn't know what they were responsible for and what I was responsible for in the process. In fact, they didn't understand the process at all. And because I was new to the TIM process for those first few projects, I didn't have a clear understanding of all of that either. It frightened me that my clients looked to me as the expert with all the answers when I was struggling just as much as they were. It took a lot of persistence and research to figure out how the whole process worked, what they contributed to the process, and my role in it. And I didn't have any experts to call on because Microsoft had disbanded its TIM team. Those exasperating experiences taught me the importance of gaining critical foundational knowledge, and the importance of understanding all aspects of the process, because clients often don't and their ignorance can cause serious complications.

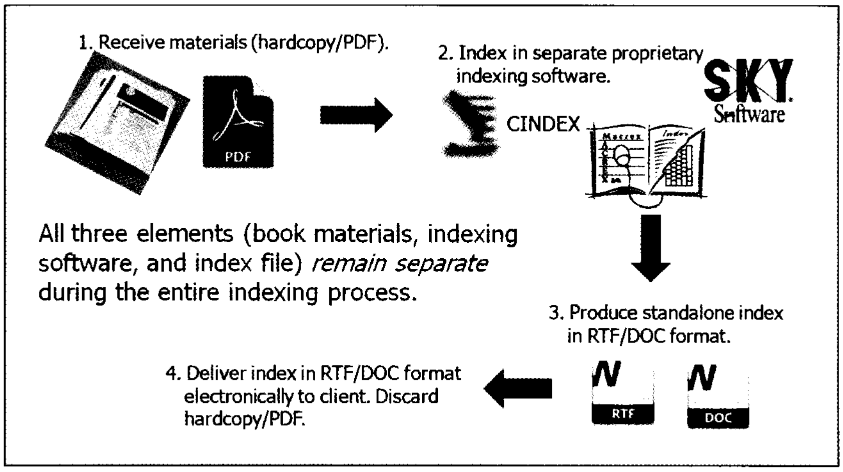

1. Receive materials (hardcopy/PDF).
PDF
2. Index in separate proprietary indexing software.
CINDEX
SKY Software
All three elements (book materials, indexing software, and index file) *remain separate* during the entire indexing process.
3. Produce standalone index in RTF/DOC format.
RTF
DOC
4. Deliver index in RTF/DOC format electronically to client. Discard hardcopy/PDF.

Figure 2 The back-of-the-book indexing process

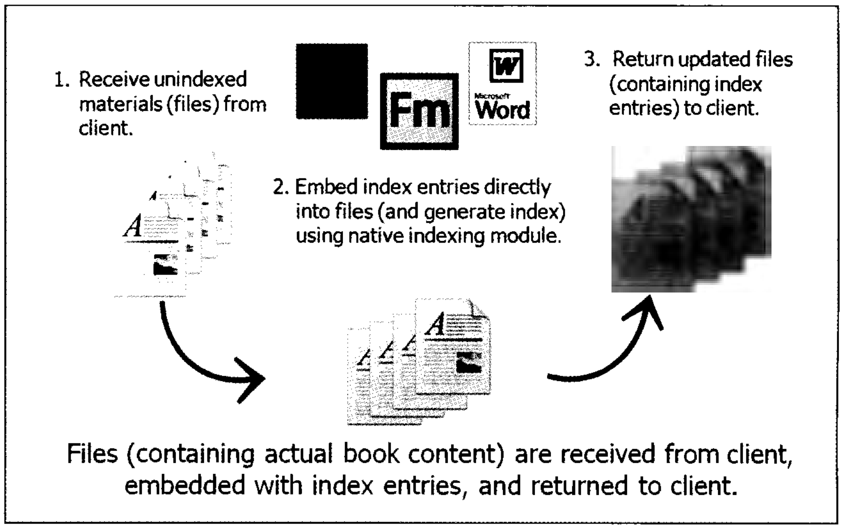

1. Receive unindexed materials (files) from client.
Fm
Microsoft Word
3. Return updated files (containing index entries) to client.
2. Embed index entries directly into files (and generate index) using native indexing module.
Files (containing actual book content) are received from client, embedded with index entries, and returned to client.

Figure 3 The embedded indexing process

What is embedded indexing?

Dictionary definitions state that to embed is to fix into, to insert into. As in inserting (or embedding) an index entry into the book content itself.

Embedded indexing describes the concept of embedding index entries into the book content. This is the 'umbrella' that includes all the various methods and formats that it supports. Additional discussion requires that we drill down into the specifics of deliverable formats (FM, INDD, RTF/DOC, XML, HTML and so on) and programs/mechanisms (FrameMaker, InDesign, XML editors and the like) used to provide those deliverables to our clients. (See Figure 1.)

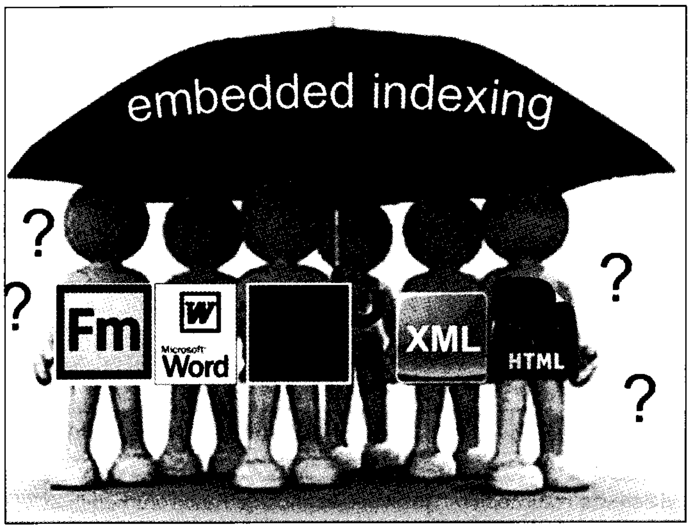


Figure 1 The embedded indexing 'umbrella'

How is embedded indexing different from back-of-book indexing?

In the BOB indexing process (Figure 2), the indexing process is completely separate from the book's content. We never receive the actual files containing the book's content. Instead we receive hard copy or PDF files and use those materials to index from. We use proprietary indexing software to create the index. And the final deliverable (the index) is transmitted to the client as a standalone RTF or DOC file. We do not return the client's hard copy/PDF files.

Compare that with the embedded indexing process (Figure 3) where the client sends us the LIVE unindexed files containing the book's content. Depending on the embedding indexing program/mechanism required, we used that program/mechanism to modify the client files as we build the index and embed the index entries directly into the file. We return the modified files (which now contain the index entries) to the client.

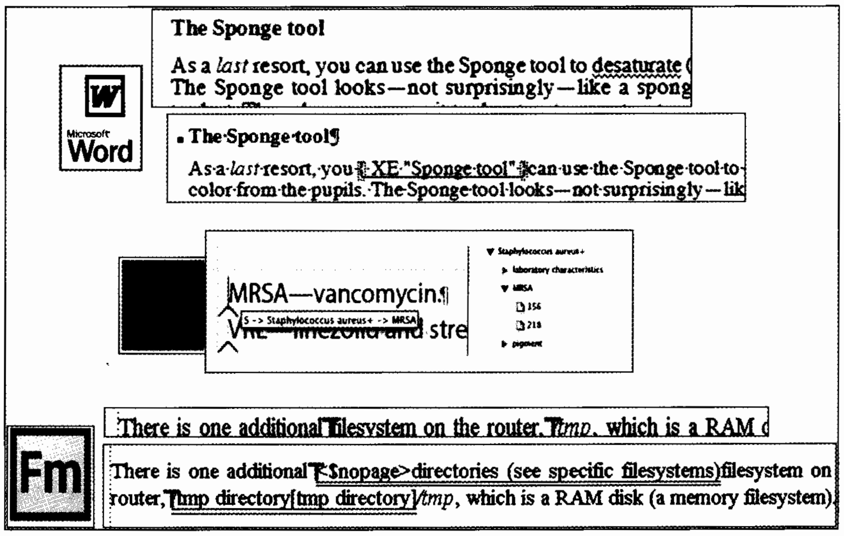
Table 1 compares the differences between the BOB and the embedded indexing processes.

Note that the embedded indexing process can have two variations:

- Software dependent embedded indexing requires the indexer to use the same desktop publishing program as the client. Figure 4 contains examples in this category.
- Software independent embedded indexing (such as HTML and XML formats) allows indexers to use XML editors or HTML editors of their choice. Figure 5 contains examples in this category.

Desktop publishing (DTP) indexing modules lack many features we enjoy with our dedicated indexing software we used for BOB projects, but the benefits they provide are the reason they continue to gain ground. Table 2 lists these limitations and benefits in more detail.

Various third-party developers have correctly evaluated the need for tools to enhance DTP indexing module inadequacies, and have provided solutions for select DTPs. Table 3 provides additional information about some of these third-party utilities.



The Sponge tool

As a *last* resort, you can use the Sponge tool to desaturate (
The Sponge tool looks—not surprisingly—like a spong

▪ The·Sponge·tool¶

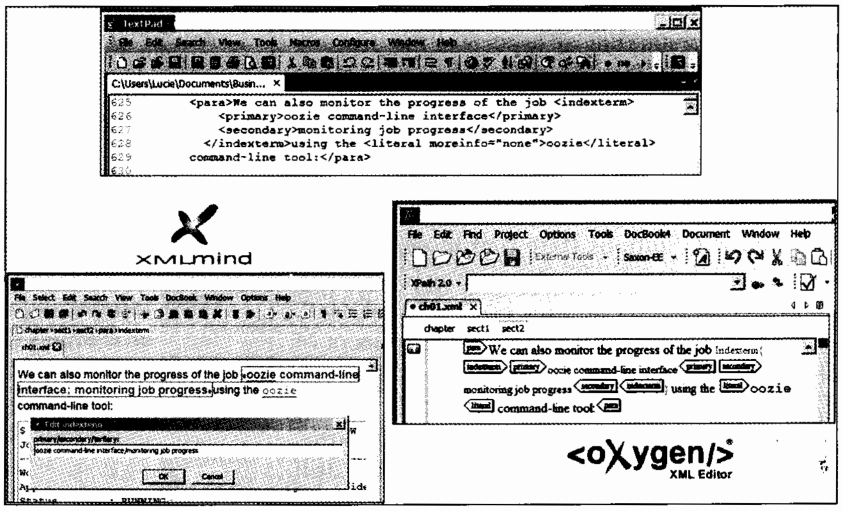
As·a·*last*·resort,·you{·XE·"Sponge·tool"·}can·use·the·Sponge·tool·to·
color·from·the·pupils.·The·Sponge·tool·looks—not·surprisingly—lik

MRSA—vancomycin.¶
S -> Staphylococcus aureus+ -> MRSA
stre

There is one additional filesystem on the router. /tmp, which is a RAM d

There is one additional <$nopage>directories (see specific filesystems)filesystem on router, tmp directory[tmp directory]/*tmp*, which is a RAM disk (a memory filesystem).

Figure 4 Software-dependent examples


TextPad
File Edit Search View Tools Macros Configure Window Help
C:\Users\Lucie\Documents\Busin...

```
<para>We can also monitor the progress of the job <indexterm>
   <primary>oozie command-line interface</primary>
   <secondary>monitoring job progress</secondary>
 </indexterm>using the <literal moreinfo="none">oozie</literal>
command-line tool:</para>
```

XMLmind
File Select Edit Search View Tools DocBook Window Options Help
chapter › sect1 › sect2 › para › indexterm
ch01.xml

We can also monitor the progress of the job oozie command-line interface; monitoring job progress using the oozie command-line tool:

Edit indexterm
primary/secondary/tertiary:

oozie command-line interface/monitoring job progress

OK
Cancel
File Edit Find Project Options Tools DocBook4 Document Window Help
External Tools
Saxon-EE
XPath 2.0
ch01.xml
chapter sect1 sect2

para We can also monitor the progress of the job indexterm primary oozie command-line interface primary secondary monitoring job progress secondary indexterm using the literal oozie literal command-line tool para

Figure 5 Software-independent examples

Practices specific to embedded indexing

Because embedded indexing requires manipulating the client files, additional work practices must be in place to ensure proper file handling. We need to be vigilant in what we do. We are working with the LIVE files here and if we damage files, it can be difficult to remedy.

Access control and time constraints

- Hot potato (golden file)
  With all embedded indexing projects, clients entrust their live files to us so we can embed the index entries within them. That means we have the most recent version of everything they have done within the files: all

Table 1 Indexing process comparisons

Process used	**Back-of-the-book**	**Embedded indexing (software dependent)**	**Embedded indexing (software independent)**
Client materials needed	Printouts or PDFs	Client files formatted as FM (FrameMaker), RTF (Word), INDD (InDesign), etc. as appropriate.	Client files in HTML or XML format
Software used for indexing	Indexer's dedicated indexing software (CINDEX, Macrex, Sky, etc.)	FrameMaker, Word, InDesign, etc. desktop publishing programs as appropriate	XML Editor of choice for XML files (oXygen, XMLMind, etc.) HTML editor of choice for HTML files (TextPad, etc.)
Index format delivered to client	Standalone RTF or DOC files	Updated client files in FM, RTF, INDD, etc. format as appropriate.	Updated client files in XML or HTML format as appropriate
Materials modified and returned to client	Not necessary	Client files modified and returned in same file format originally provided by client.	Client files modified and returned in same file format originally provided by client

Table 2 Limitations and benefits of DTP indexing modules

Limitations	**Benefits**
Creating index terms: No index preview No autocomplete of index entries Tiny marker box size May be difficult or impossible to handle special strings (page range, italics, bold) Cross-reference restrictions	*Creating index terms:* Autogeneration of entries possible with some DTPs (from key words, paragraph tags, character tags, permuting/rotating marker text)
Editing terms: Limited or no change propagation within index entries No index preview No display/viewing index entries within content No temporary grouping of index entries	
Work process: Requires 50–100% more time than BOB indexing File access needs to be controlled Index must be regenerated in separate step to recognize changes	*Work process:* Can start indexing before final pages Concurrent proofreading Potential reuse in future editions Potential reuse in other formats

the content, formatting and images. They trust us to complete our task and to return the files without damage or inadvertent modifications. This is a huge responsibility and needs to be recognized accordingly.

- **Version control systems**
  To ensure that our changes don't get overlaid by others' work, clients use version control systems to lock out other users and to handle file merges appropriately. Each client has procedures customized to their environment. It is up to us to learn those procedures and follow them to the letter. Because the procedures can be difficult to remember, I create cheat sheets to exhaustively document each step of the client process for handling files. Cheat sheets are life- and time-savers.
- **Time restrictions**
  We have the hot potato: the LIVE files. That means everyone else is unable to work on the files while we have them. Time is precious. We need to stay on task and to meet the index deadlines. What else is new?
- **Proofreading concurrently**
  There's always pressure to shorten the publishing cycle, so many clients schedule proofreading at the same time as indexing. The proofreaders work from hard copy and simply notate the errors (capitalization, wording, typos) until they regain access to the files. That means I typically find a lot more inconsistencies in the unproofed files than with BOB indexing projects.

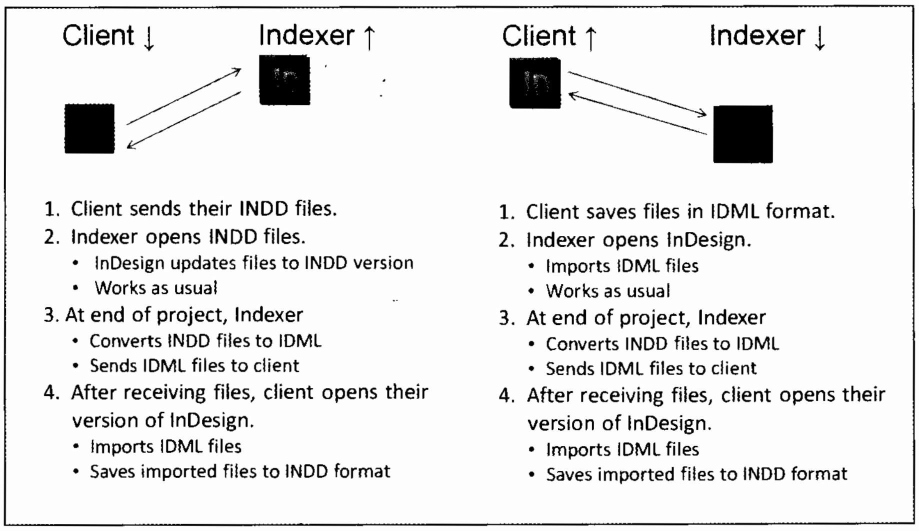
File breakout for FrameMaker or InDesign projects

Is the book's content combined into one huge file? Or are book chapters broken out as separate files?

Separate files are better than one combined file. In the multiple years I have been indexing in InDesign, I have only had two or three clients who had one huge INDD file for me to index from. That's a dangerous game to play, as file corruption is something that is much more difficult to recover from if the entire book – rather than one chapter file – becomes corrupted.

Table 3 Third-party utilities augmenting indexing functionality

Format/DTP	**Third-party plugin(s)**	**Additional information**
FrameMaker	IXgen	Frank Stearns Associates `www.fsatools.com/`
Microsoft Word	DEXembed	The Editorium (Jack Lyon) `www.editorium.com/dexembed.htm`
	WordEmbed	James A. Lamb `www.jalamb.com/wordembed.html`
InDesign	Kerntiff Publishing System (KPS plugins)	`www.kerntiff.co.uk/products-4-indesign/indexutilities`
HTML		
XML	IXMLembedder (in development)	Leverage Technologies `www.levtechinc.com/publishing-indexing-products/utilities/IXMLembedder.asp`



1. Client sends their INDD files.
2. Indexer opens INDD files.
   - InDesign updates files to INDD version
   - Works as usual
3. At end of project, Indexer
   - Converts INDD files to IDML
   - Sends IDML files to client
4. After receiving files, client opens their version of InDesign.
   - Imports IDML files
   - Saves imported files to INDD format

1. Client saves files in IDML format.
2. Indexer opens InDesign.
   - Imports IDML files
   - Works as usual
3. At end of project, Indexer
   - Converts INDD files to IDML
   - Sends IDML files to client
4. After receiving files, client opens their version of InDesign.
   - Imports IDML files
   - Saves imported files to INDD format

Figure 6 Considerations when the client and indexer have mismatched versions of InDesign

Software versions for FrameMaker or InDesign projects

Knowing which version of the program the client is using is essential. InDesign is InDesign and FrameMaker is FrameMaker – except when your client has one version and you have another.

InDesign and FrameMaker don't play nicely between versions, so issues arise when version mismatches occur. There are ways to hand off files between versions but they come with some risks and extra effort for your client. (See Figure 6 for an InDesign example. FrameMaker mismatched version issues are similar to InDesign issues.) But why would a client agree to extra file manipulation if they could easily get another indexer who has the version needed for the project?

File downloading/uploading

DTP files occupy megabytes and typically exceed email attachment size allowances. The following file-sharing solutions are more effective and efficient for distributing files for these projects:

- FTP (FileZilla, PuTTY, etc.)
- client repositories (SharePoint, Egnyte, OneDrive, Microsoft SkyDrive, etc.)
- web-based file-sharing sites (Dropbox, etc.)
- proprietary client systems.

Version control and compliance is a serious responsibility. It bears repeating that these files are the hot potato. And it says a lot that the clients trust you to do what you need to do. Just follow your client's procedures. They will lay them out for you. Create that cheat sheet.

Client guidelines

Clients might have guidelines on how to handle:

- Index entry placement:
  - Not at the beginning of paragraphs (where formatting marks might exist).
  - Not within figures or notes or sidebars (as they can become distanced from content and page ranges would be adversely impacted).
  - Not in tables (which might not handle index entries correctly).
- File name changes to easily distinguish indexed files from unindexed files.
- Page ranges:
  - Just for the starting page of discussion
  - Only if discussion is at least three pages.
  - Force adjacent pages (but noncontiguous discussion) to show as page range.
- Markup for bold/italics: InDesign does not handle italics or bolding. For clients who need italicized entries, I mark up the affected entries and send the highlighted list to the client to apply manually.

What work is there for embedded indexers?

From my perspective, the market continues to evolve and the potential is unlimited. Computer book publishers have occupied this arena for decades, but there are publishers in other specialties who already produce books in InDesign or FrameMaker or Word. They have been content with standalone BOB indexes and have not seen the need to switch to embedded indexes. These clients would be the easiest starting point to find opportunities to expand in embedded indexing arenas.

How much will it cost?

- Consider the **time** investment for learning
  - new software
  - indexing processes
  - client practices.
- Consider the **money** investment for
  - new software (DTP packages, XML/HTML editors, third-party plugins)
  - software version upgrades
  - potential equipment upgrades to handle more sophisticated programs and larger files: memory and processor upgrades, larger/multiple monitors, reliable/fast Internet connections.

How will you learn?

Comprehensive training can a challenge to find and might involve online research. Some starting points to consider:

- the ASI website (asindexing.org):
  - A three-part online webinar on Indexing in InDesign (full disclosure: I'm the instructor).
  - DTTF (Digital Trends Task Force) maintains a plethora of information on trends in digital publishing.
  - Various chapter/regional/national conferences occasionally provide embedded indexing workshops.
- UC Berkeley had an Embedded Indexing course (but its status is unknown at this time).
- Resources pages on my (and other indexers') website provide step-by-step articles on indexing in FrameMaker, Word, or various plugins.
- SI embedded indexing course at www.indexers.org.uk/index.php?id=557
- Individual indexers known for their expertise might be amenable to providing one-on-one tutoring (it never hurts to ask!).

Conclusion

Remember, knowledge is power. I hope that now that you have more information, you can answer my original question: Should I jump on the embedded indexing bandwagon?

What do you think? Should you?

Lucie Haskins became an indexer in 2000 after a long career in corporate America, with roles spanning the computer industry and management consulting. She specializes in embedded indexing and in computer- and business-related topics, and has presented workshops on various indexing topics, instructed for UC Berkeley's online indexing course, and served ASI in chapter and national positions as well as on its webmaster team. Email: luciehaskins@gmail.com *Website:* luciehaskins.com

Does embedded indexing have a future?

Cheryl Landes

Cheryl Landes, an indexer who specializes in working with new technologies, discusses the ways in which indexing tools and techniques have changed over the years, and suggests, on the basis of her recent experience with clients, that traditional embedded indexing may be becoming a thing of the past.

While the concepts behind the craft of indexing have remained the same over the years, the techniques and tools we have to create indexes are changing. The advent of ebooks in the publishing industry accelerated this change. At the same time, the growing popularity of ebooks has forced publishers to look at new ways to produce books and indexes in an efficient, cost-effective manner. Embedded indexing is an effective way to tag content from an original source and create books in various formats that can be read on a variety of devices or in print. Single-sourcing content like this reduces labor and production costs and increases profits for the publisher.

Traditional embedded indexing meant tagging content in a word processing program, such as Microsoft Word, or a desktop publishing package, such as Adobe InDesign, QuarkXpress and Adobe PageMaker. However, when publishers want to single-source content, all of these programs are limited in their abilities to create the different types of formats readers want. As a result, publishers are turning to new, often custom, methods of creating and producing content, and indexers are also using different methods to create entries.

This article focuses on the future of embedded indexing, based on my experiences and observations as an indexer who works with new technologies. My primary indexing target market is technical materials, but I have also indexed hundreds of trade books since 1991, the year I started indexing. On the technical side, the majority of my work is embedded indexing, creating keywords for online help systems and web-based content, and optimizing content for search. In the 1990s and early to mid-2000s, my non-technical (trade book) clients provided hard-copy (printed) manuscripts or PDFs, and I used my dedicated indexing software (Macrex) to create the indexes. I delivered my final indexes in either Microsoft Word or rich text format, and a graphic designer laid out the index in a desktop publishing program.

When ebooks became popular, my non-technical publishing clients began asking me to create embedded indexes in InDesign. They sent me the original manuscript files, created in InDesign, and I would insert the index entries in the files with InDesign's tagging features. When I finished, I would return the tagged files to the graphic designer, who would check the layout of each of the files in InDesign, make adjustments as necessary, and regenerate the index in InDesign before the files went to final production.

During the past five years, however, I have noticed a sharp decline in my clients requesting traditional embedded indexing. They are switching to content management systems, where the layout for the books is created separately from the content through templates, more commonly known as stylesheets. These stylesheets are applied when the content is ready to publish. With this process, the publisher receives the manuscript from the author in a word processing or text file, or more commonly during the past couple of years, the author enters the text directly into the content management system. Then the publisher tags the content with the appropriate style names to match the stylesheet, such as the book's title, main headings, subheadings, paragraphs, bulleted lists and other layout elements. Some of the newer content management systems I use have the tags set up so that the author can even apply them, and then the publisher checks the layout before the content is published. Most of the systems I have used are either XML-based or in a wiki, but XML is the most popular because of its flexibility.

A simplified version of the publishing process in an XML-based content management system is shown in Figure 1.

When clients hire me to index books created in these content management systems, I create the entries either directly in the raw content by tagging or by a method outside the content, which is integrated into the content later. The two most common methods I use now are described below: directly tagging content in a repository, and indexing by paragraph numbers.

Directly tagging in content repositories

When a client asks me to create an index directly in a content repository, they give me a user name and password to access the content. By the time I start indexing the content, the client has added the tags for headings, paragraphs, bulleted and numbered lists, and other layout elements to match the stylesheets for each type of output (ebook, print and so on). The client provides special instructions for me to create tags, such as the format and name of the tags to use, separators for main entries and subentries, and formatting for cross-references.

Table 1 shows an example of tags used in indexing wiki content for one client.

After I finish inserting the index tags in the content, I notify my client. Then the client compiles the tags into a formatted index and sends the index to me to review. Then I edit the index by making changes in the tags. After I finish, I notify the client again, and the client compiles the index again for a final edit pass. After I finish the final edits, the book goes into production.

Indexing by paragraph numbers

Indexing by paragraph numbers is becoming the most popular method of creating indexes for single-sourced content. With this method, my clients number each paragraph with XML tags in the content repository and generate PDFs of each chapter. I index the chapters from printouts of the PDF using Macrex. Instead of using page numbers, I use paragraph numbers. If a discussion spans more than one paragraph, I index the paragraph numbers as a page range. A detailed example of this method is shown in Figure 2.

My index for this chapter, using paragraph numbers as reference locators, was created in Macrex:

Baker City, Indian raid near, 52–58
Carlisle farm, Indian raid on, 52–58
cats, as target of jokes, 52–58
feathers on a cat, 56–57
felines. See cats
Indian raids, jokes pulled during, 52–58
jokes, cat as target of, 52–58
molasses, cat dunked in, 56–57
Mouser (cat), 52–58
Native American raids. See Indian raids
Oregon, Indian raids in, 52–58
raids by Indians, jokes pulled during, 52–58
Upper Willow Creek, Indian raids at, 52–58

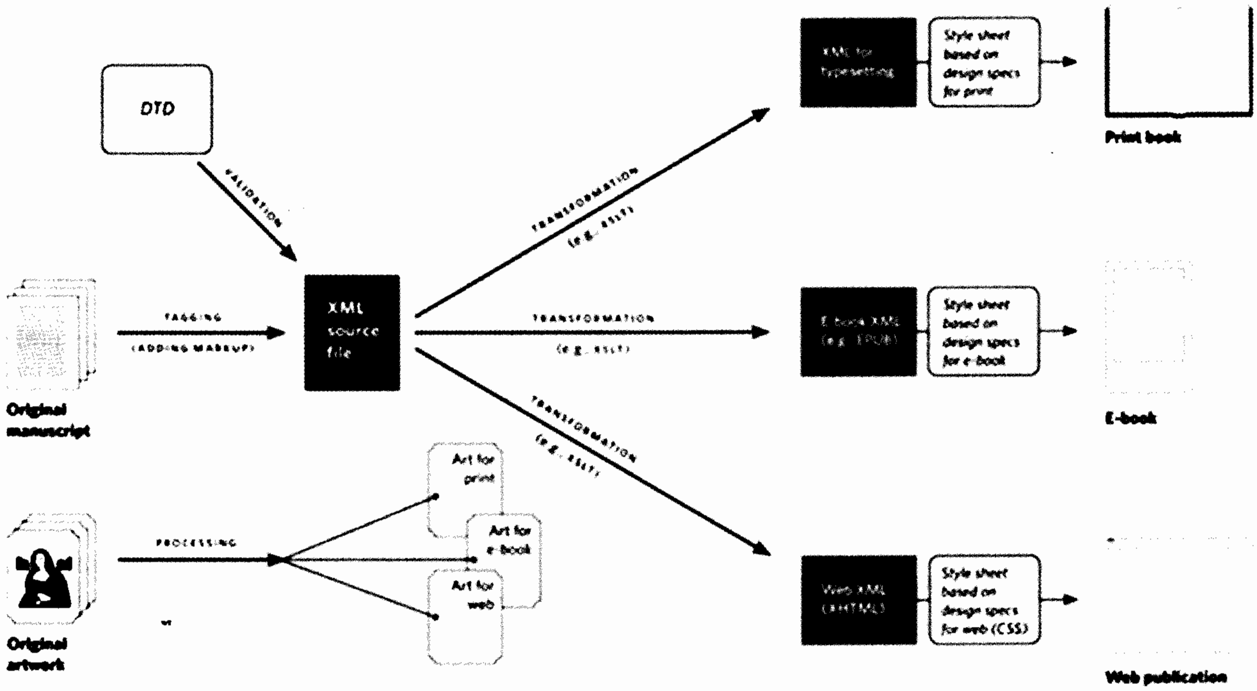


Figure 1 A simplified version of the publishing process in an XML-based content management system

Source: Chicago Manual of Style Online, 16th edition, www.chicagomanualofstyle.org/tools_workflow.html

Table 1 An example of tags used in indexing wiki content

Single entry, no page range	\{in main entry; subentry; sub-subentry} Example of a tagged single entry: \{in cats; sleeping patterns of}\{in sleep patterns, cats}Cats sleep an average of 15 hours a day. Displays in the book as: cats sleeping patterns of, 43 sleep patterns, cats, 43
Page ranges	Start range with \{is id; main entry; subentry; sub-subentry} End range with \{ie id} The indexer needs to provide an ID number for each page range so that the correct ranges display in the index. My client uses a two-digit chapter number, followed by a two-digit entry number, such as: Start range - \{is id **ch01-01**; main entry; subentry; sub-subentry} End range - \{ie id **ch01-01**} (bold shown above for emphasis – not used in the tagging) Example of tagged page range: \{is id ch06-01; Sand Island, fishing at}\{is id ch06-02; Columbia River; Sand Island, fishing at}\{is id ch06-03; Washington (state); Sand Island, fishing at}Sand Island, a small piece of land in the mouth of the Columbia River, had been a menace to navigators but a blessing to \{is id ch06-04; fishing, Sand Island}fishermen for decades. Navigators dreaded sailing in this area, because constant erosion and silt deposits continually changed the shape and location of the island. Fishermen saw the waters surrounding Sand Island as a gold mine. Average \{is id ch06-05; salmon fishing, Sand Island} salmon catches ranged from 300 to 500 tons each season, the largest in the world. .... Sand Island became a memory 36 years later. When the north and south jetties were built on the Columbia River in 1932, the island was completely covered with water. However, alluvial deposits eventually caused the island to resurface in a smaller form, closer to Cape Disappointment. The salmon runs have never vanished, and fishermen return for their share every year.\{ie id ch06-01}\{ie id ch06-02}\{ie id ch06-03}\{ie id ch06-04}\{ie id ch06-05} Displays in the index as: Columbia River Sand Island, fishing at, 15-18 fishing, Sand Island, 15-18 salmon fishing, Sand Island, 15-18 Sand Island, fishing at, 15-18 Washington (state) Sand Island, fishing at, 15-18 (Source of text: Cheryl Landes, *Those wild Northwest days*. Victoria, BC: Trafford Publishing, 2006.)
Cross-reference: See	\{ic term; term-to-see} Example of a tagged See cross-reference: \{ic felines; cats}Cats sleep an average of 15 hours a day. Displays in the book as: felines see cats
Cross-reference: See *also*	\{ia term; term1, term2, term3, ...} Example of a tagged cross-reference: \{ia felines; tigers}Tigers sleep from 16 to 20 hours a day. In the tropics, they often sleep in water holes to protect themselves from insects. Displays in the book as: felines see *also* tigers

The compiled index for this chapter, which is published in the book:

Baker City, Indian raid near, 5–6
Carlisle farm, Indian raid on, 5–6
cats, as target of jokes, 5–6
feathers on a cat, 6
felines. See cats
Indian raids, jokes pulled during, 5–6
jokes, cat as target of, 5–6
molasses, cat dunked in, 6
Mouser (cat), 5–6

```
<H1 n='50'>A Cat of a Different Feather</H1>
<p n='51'>Thousands of stories exist about the lives of American pioneers, but not many are told or written about pioneer cats.</p>
<p n='52'>Mouser was a cat that lived on the Carlile family farm at Upper Willow Creek near Baker City in northeastern Oregon. His life was the same as that of any average farm cat of 1878, until he became the target of a joke during an Indian raid.</p>
<p n='53'>At that time, Indian raids were common in northeastern Oregon. When tribes went on the warpath and began to raid the countryside, riders on horseback tried to warn their fellow settlers before the attack. The homeowners then fled, taking refuge wherever they could in the brush, timber, or mountains. They did not have time to gather their belongings.</p>
<p n='54'>Therefore, when Indians moved in on Upper Willow Creek, the Carliles left Mouser in the house as they fled for safety. When they returned, they found the house ransacked, as expected. But they also saw a small, strange, four-footed creature dash past them. Feathers were scattered everywhere.</p>
<p n='55'>When Mr. Carlile managed to get near the animal, it meowed at him. It was Mouser!</p>
<p n='56'>Closer investigation unraveled the mystery of Mouser's new attire. When the Carliles fled, they left a sealed keg of molasses in the kitchen. It was not open. Apparently, the taste did not appeal to the Indians. Instead, they had grabbed Mouser, dunked him in the molasses, and turned him loose. As an added touch, it seemed, they ripped open the feather bed so the feathers would stick to the cat's body.</p>
<p n='57'>Of course, it was impossible to salvage either the molasses or the bed. But after a long, wild chase, the Carliles finally succeeded in catching the frantic Mouser. They sheared and bathed him, and eventually, he became a normal-looking cat again.</p>
<p n='58'>Fortunately, Mouser witnessed no more Indian raids. He lived out the rest of his life as a happy, carefree cat on Upper Willow Creek.</p>
```

Figure 2 Example of a tagged chapter by paragraph number

Source of text: Cheryl Landes, *Those wild Northwest days*, Victoria, BC: Trafford Publishing, 2006.)

Note: Paragraph numbering conventions vary by publisher.

Native American raids. See Indian raids
Oregon, Indian raids in, 5–6
raids by Indians, jokes pulled during, 5–6
Upper Willow Creek, Indian raids at, 5–6

After I finish indexing, I create a Word file of the index and send it to the client. Then the client runs a script that migrates the index to the content management system, embeds the index entries into the content by paragraph number, and translates the paragraph numbers into page numbers. When migration is complete, the client compiles the tags into a formatted index and sends the index to me to review. If I need to make any corrections, I edit my Macrex file and send the client an updated index in Word. The client updates the index using the same migration process, compiles the index, and the book goes into production.

Will embedded indexing survive?

Based on my experiences, embedded indexing is losing popularity for trade indexes. I have not received any requests to create an embedded index during the past two years. Clients now ask me to use the paragraph indexing method described above. I believe as technologies change, this process will become the preference for indexing. Indexers also find the process faster and more efficient, because they can use their dedicated software and index in an environment in which they are familiar.

Cheryl Landes founded Tabby Cat Communications in Seattle in 1995. She has indexed since 1991, when she began her career as a technical writer at a marine transportation company in Seattle. Today, she is a technical and marketing communications consultant and specializes as a findability strategist. She has published five books and contributed chapters about findability to The language of content strategy *(XML Press, 2014) and* The language of technical communication *(to be published in 2016 by XML Press). Cheryl currently lives in Vancouver, Washington.*
Email: clandes407@aol.com

A look at classification and indexing practices for elementary school children: who are we really serving?

Valerie Nesset

As indicated in the findings of a larger study investigating the information-seeking behavior of Grade 3 students it is asserted that traditional classification and indexing methods used in school libraries and print reference materials targeted at young students often do not address young searchers' unique information needs and searching behaviors.

The findings from a broader study of the information-seeking behavior of Grade 3 students conducted in an elementary school located in a suburb of Montreal, Quebec, Canada indicate that the classification of print materials within an elementary school library may not be intuitive to young searchers. Furthermore, once a book was found, often the indexing practices did not facilitate information retrieval within the text itself. In order to address this problem, librarians and indexers need to be aware of the rapid changes in intellectual development that occur during a child's tenure in the first six grades of elementary school and how they can affect their information-seeking behavior.

The participants in this particular study were searching for information for a project on Canadian animals in winter using both print and electronic resources. This paper concentrates on the former. In the school library, it was observed that the students were not in the habit of consulting the card catalog before searching for materials but instead preferred to browse the shelves, which in the non-fiction section were identified with simplified labels (such as Animals and Insects). As a result, they did not rely on the call numbers for locating suitable books; rather, they tended to use visual cues (such as book covers and shelf labels).

For this particular study, at the teacher's request, relevant books were removed from the school library, shelved by call number and placed on a small bookcase in the classroom. This resulted in a rather artificial environment as it meant that it was not possible to observe the students search for print resources in the actual library. (Cooper (2002a) employed a similar technique in her study of the information-seeking behavior of 7-year-old children.) Yet, although it was somewhat artificial in terms of a public library situation, in the classroom setting this was a common practice and occurred even within the school library. For example, sometimes a teacher would request the library staff to select and remove books on a specific subject from the regular collection and place them together for easy access by the students in the class.

Even though there were relatively few books on the subject (under 100), observation over a period of weeks