October 12, 2011

To: Markus Gylling
ePub 3.0 Working Groups

From: Jan Wright (jancw@wrightinformation.com)
David Ream (davеream@levtechinc.com)
Members of the American Society for Indexing's Digital Trends Task Force (DTTF)
(Mary Harper, Charlee Trantino, Michele Combs, Pilar Wyman, Joshua Tallent, Cheryl Landes)
Email: dttf@asindexing.org
Website: http://www.asindexing.org/i4a/pages/index.cfm?pageid=3647

Contents

Index Functionality in ePub

This proposal is divided into two parts: (1) this background segment that identifies the anatomy of indexes, workflows for indexing, and suggestions for usability in an on-screen environment; (2) our first attempt to outline the index functionality.

Background

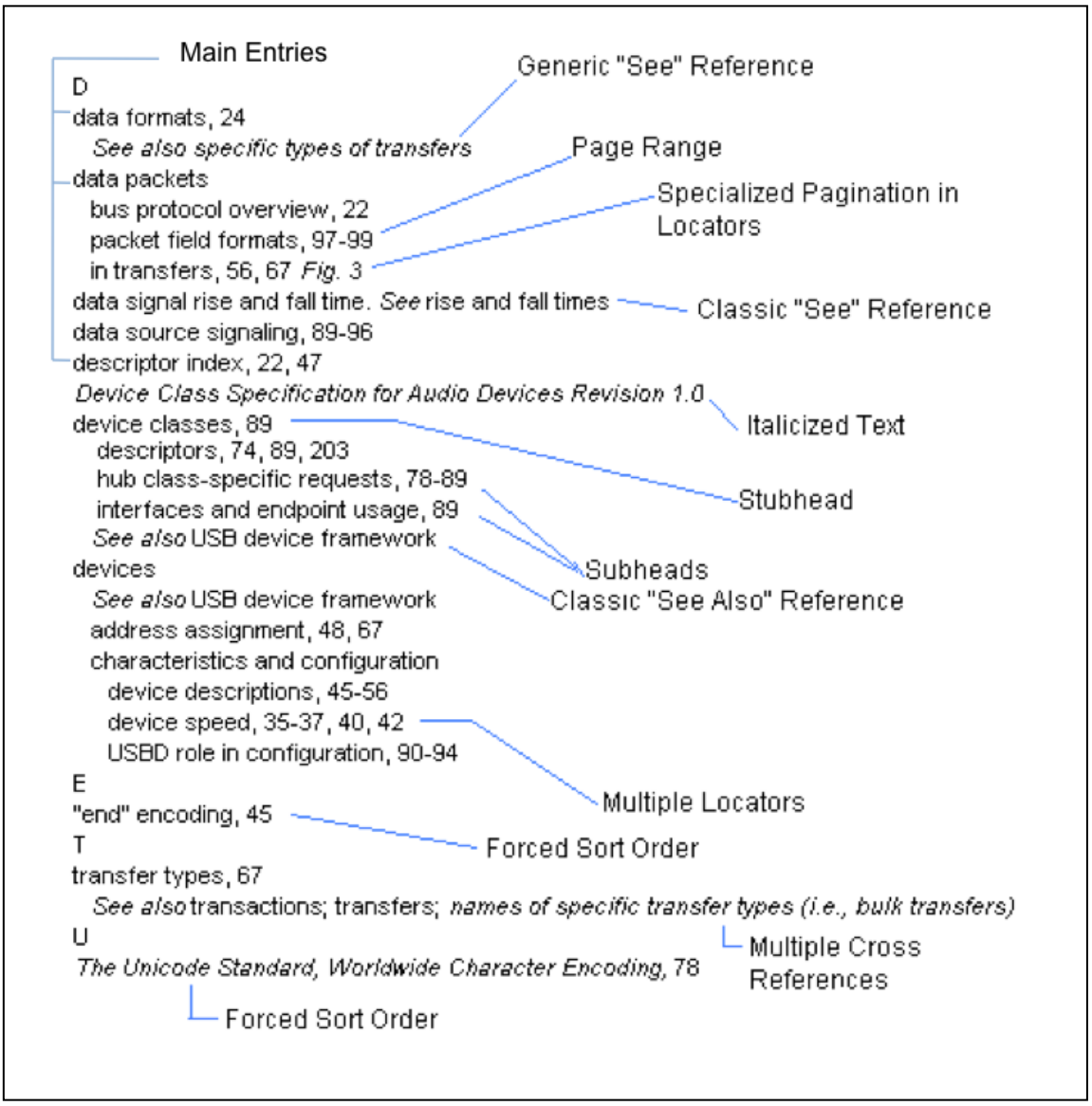
Most current eBooks are displaying their index as a chapter at the end of the book, with or without linking. This file should be developed into a more effective feature, in the same way that a book-specific glossary could. We recommend the user interface combine search, index, and glossary into a single screen that would allow the user to have access to all semantic actions needed in one place.

The index file can be linked into the text by the publisher through the use of anchors inserted into the text, each uniquely named or numbered. The index should use these anchors as the basis for links into the text at precise locations, and should also allow contextual display of a specified number of words following the anchor, to show the term in context.

Small screen indexes have been available for online Help systems for 20 years, and ePub can adopt some of the best practices and usability measures from these systems.

Index Anatomy

The following diagram shows all the pieces of a standard print index. In most cases, indexes are created in a database format using dedicated indexing software, with each entry making up a record. The records are then manipulated, consolidated and formatted into the traditional displays seen on paper or in PDF or eBook chapters. There are a variety of other methods for creating indexes, which are discussed in more detail in the Publisher Workflow section of this document. For simplicity's sake, we are looking here at the basic features of print-based indexes, and identifying the elements. Some of these index features will not survive the transition to eBooks:

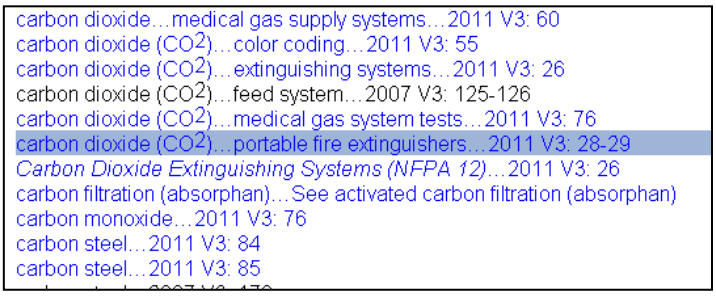


Anatomy of indexes: terminology can vary (main entries/main heads, subentries/subheads)

Note that index elements can consist of

- solitary entries, such as “descriptor index”
- two-part entries, such as “data packets:bus protocol overview”
- three-part entries (sub-subheadings), such as the section under “devices”
- specialized entries for cross-referencing such as “see” and “see also”

This traditional structure means the individual records have varying numbers of fields for a complete entry, and employs attributes to indicate specialized entries. Special codes set the sort order to ignore symbols or unnecessary words, such as quotation marks in the “end” entry and “The” in the last entry.



Raw index records before compiling. Note that the main entry is repeated in each record

Usability Best Practices

Print

With print books, case studies showed that when a book had two indexes, an author-only and a subject-only index, most readers did not find the second index and had unsuccessful search experiences. One index is the best practice. A fallback is to have multiple indexes included as separate chapters in an eBook, but combine the entries into a file for the main index display.

Online Help Indexing and Search

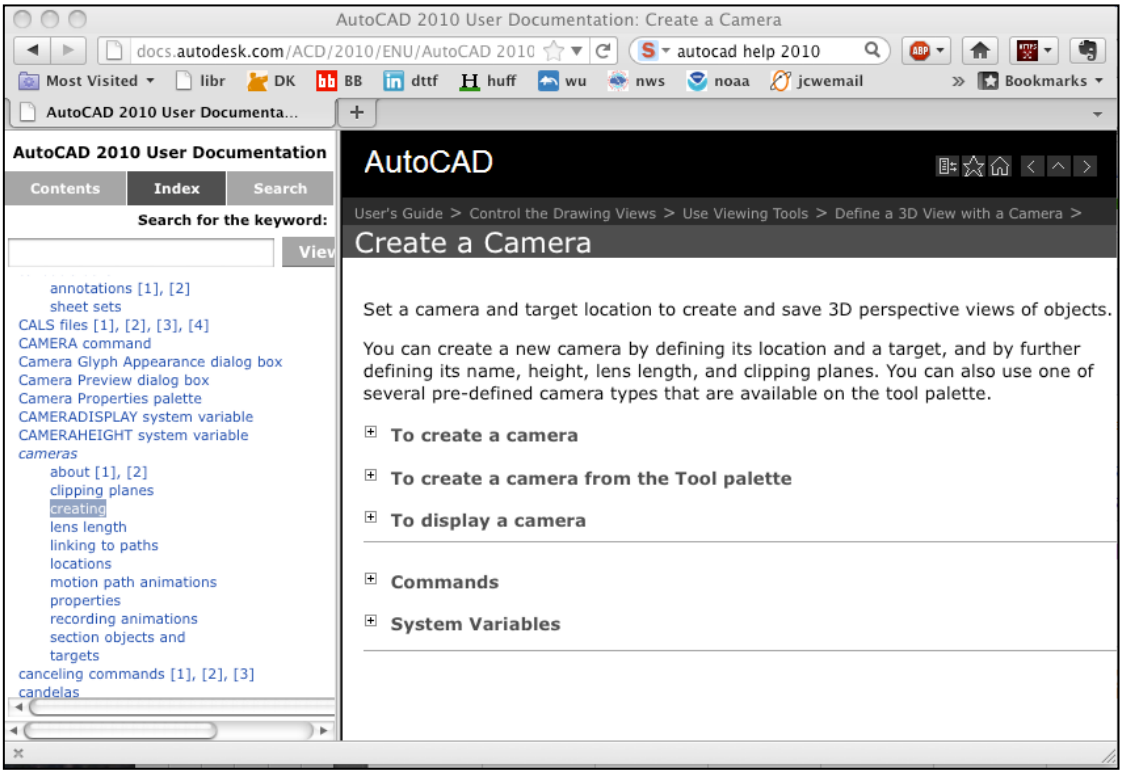
At Visio Corporation in the late 90s, the effectiveness of online Help search vs. index was researched. The results showed the users did not understand the difference at that time between full-text search and the index. On a small screen, the user would not scroll down past the bottom of the first results window. When they used search, their results were not as good as when they used the index.

The large publisher of law materials, Bureau of National Affairs (BNA), conducted a usability study (results are available at http://dl.dropbox.com/u/2248375/Using%20BNA%20Indexes%20study.pdf) to "detect and quantify any disparity between text searching and research aided by indexes. Its study was conducted at law schools and compared success rates and completion times for text searchers with those of index users. Law students completed a series of research tasks using the online version of United States Law Week. They answered half of the questions using text searches. For the other half they used the online index."

BNA further describes how "in the BNA Usability Study, index users had an 86 percent success rate while text searchers had only a 23 percent success rate. The study included both single answer and more complex research tasks. Results for the various types of tasks confirmed many limits of text searching. Text searching can be successful in locating proper names or an isolated piece of information involving very specific facts. However, for most legal research tasks, using an index provides more relevant and complete results."

Best practice take-aways are that we can expect a Google-experienced population will likely not easily make the distinction between Search and Index, and may not scroll down more than one screen. We can expect, though, from the use of taxonomies and breadcrumbs on the web that the display of structured and indented information showing broader, narrower, or related concepts helps the reader choose.

At Microsoft Corporation in the mid 90s, usability studies conducted on a two-pane online Help index display, index on the left and topic on the right, showed proof that the "scent of information" allowed the user to jump from index to the right pane, even if that pane contained more choices and not yet actual information. Leaping from left pane to right with terms in context would be a much more findable "scent of information."



Web-based example of online Help index with type-ahead entry box

Book-like Chapter-based Indexes vs. Search

Barnum, Henderson, Hood, and Jordan (available at http://dl.dropbox.com/u/2248375/STC_index_usability.pdf) summarize their findings comparing use of PDF-hyperlinked book-like indexes and search:

- The hyperlinked index is the more effective look-up tool
- The hyperlinked index is the more efficient look-up tool
- Users considered the search version slightly more engaging
- Users considered the search version slightly more error tolerant
- Users considered both the search and the hyperlinked index versions easy to learn
- Users slightly preferred the search version to the hyperlinked index version
- The average time spent performing a look-up task in the hyperlinked index version was less than the time spent completing a task of similar difficulty in the search version

Search Processes and Returning to Try Again

We need to keep in mind that readers turn to a non-fiction book in several mental states:

- having never read it and wanting to see if needed information exists in it

- having read it, and needing to retrieve information that they know exists
- formulating a concept to be found and guessing the words to use
- reformulating their question as they learn more and change focus, to another term, to a narrower term, or to a broader term

Search can be helpful in several stages here, but there are crucial stages where access to an index's pre-analysis of a text saves a reader time and frustration.

- New users who don't know the phrases the author uses aren't well-served when they can't browse for alternative terms to describe their concept.
- A simple text search gives no clues as to where or how a specific occurrence of a keyword relates to other occurrences.
- Indexes frequently reference conceptual discussions that may be skipped over by a text search because a particular keyword is not present in the section of text, only a synonym or an assumption that the reader in linear mode will understand the aboutness (i.e., Hemingway cats are polydactyl cats, but a section of text using only the Hemingway terminology will be missed by a search for "polydactyl").
- A text search will find only the term the reader typed in, unless a stemming mechanism is in place. An index, by virtue of having similar terms preceding and following an entry, allows the reader to know that these terms also exist, and can create happy accidents of retrieval.
- People who are browsing casually to see if a name is mentioned can use Search as well, but if they want to see how extensive the coverage is on a topic, Search in its current form on an eReader is not the tool for them.

Users also change their focus and their specificity during the search process as they learn more about the concept they are interested in: a person who originally wanted just a general overview may focus in on an aspect, and then may want to return to an index to see if that aspect is covered, and in what depth. The Search feature can help them, but again, this is like a one-word Google search: the user must wade through the results one by one.

Having an index is like having a conversation with someone who has read the book. "Oh yes, it's there, and look at all the rest of the related material I found as well, some of which may be what you are looking for!" That discussion is missing from eBooks, and we fear that as more and more eBooks are published without working indexes, that conversation may cease to exist, as it did in early online Help indexes.

Recommendations

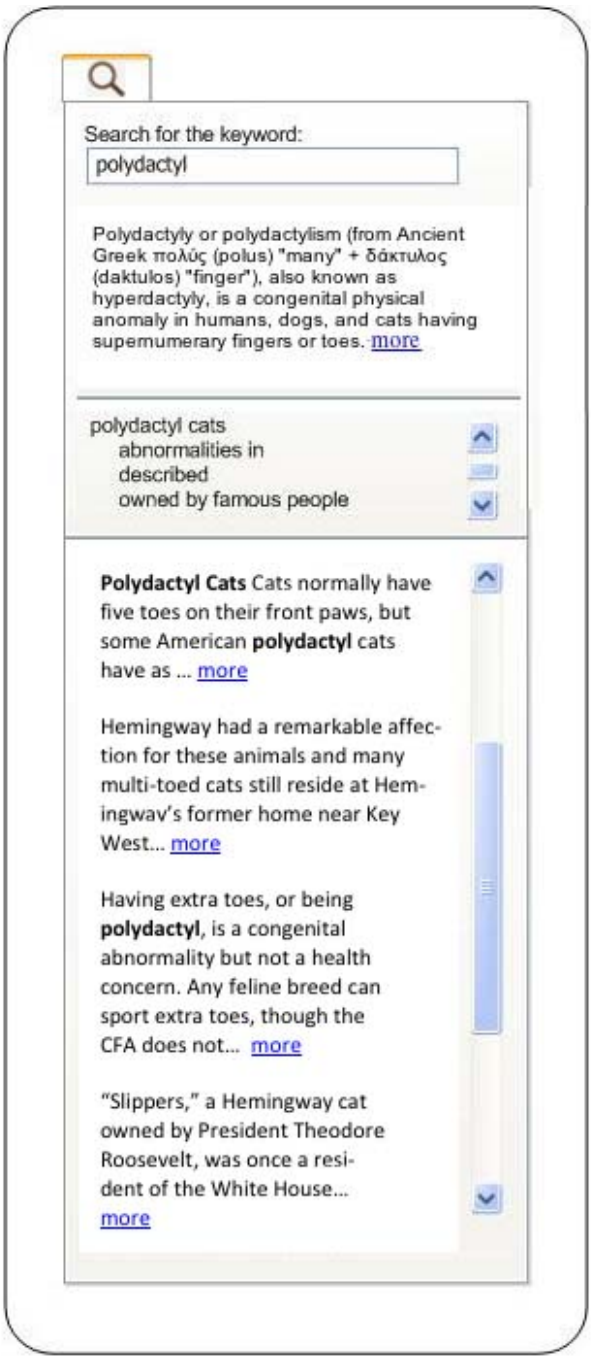
We recommend that the index should be displayed from a separate file containing the index entry terms, linked to anchor points in the text. A print-like chapter is the fallback, showing a traditional book-page display, with locators hyperlinked into the text.

The user experience we envision is one in which the user highlights a word, or the user clicks on a magnifying glass icon or Search option and starts typing, and then:

- a combined screen display shows the definition from the glossary if one is available, or a fallback definition from an installed local dictionary or a Web dictionary

- a screen displays key entries for the index, if available, or
- full text search results are displayed if no index is available

Choosing an entry, or the full text search, should show the concepts for the term in context with a specified number of contextual lines displayed to allow the user to choose a location in the book to research or go to.



(Developed from current search interfaces, online Help interfaces, mobile Help interfaces, and Google Books by Jan Wright, Pilar Wyman and Cheryl Landes)

The index+search should remember the last searched term until a new one is entered by the reader.

Implementation of index display is dependent on the manufacturers of eReaders, but considering the usability issues of early online Help Find and Index functionality, and the data showing that multiple locations can confuse users, a display system like the one above is recommended. In the user's mind, the magnifying glass will become the one point they can count on for researching, defining, or resources for working with the text. Each piece of the interface, glossary, dictionary, index, should be a separate file that is called at time of use. At some point in the development of semantic interlinkings, we may be able to display further connections as topic maps or taxonomic structures, but as a baseline, a file specification for these resources should be developed.

Best practices for online or onscreen indexes limit the number of levels presented to just two: main entries and subentries. Depending on the interface, most online Help indexes limit cross referencing, and in some cases it is not used at all. Developing the index entries under each synonym take the place of cross referencing in these cases. Print page folios limit the length of an index, but onscreen indexes can be fuller and include full coverage at synonyms.

An ePub index needs to be capable of displaying four out of the six basic components listed below. Cross referencing, and its associated switches or attributes will be dependent on eReader functionality for an interactive display, but should be retained for the fallback chapter-style index. How these components should be or could be encoded in XHTML or XML is open for discussion. Multiple solutions are possible, requiring different levels of eReader technology to implement them, and differing levels of tool development on the publishing side to produce the files.

	Entry unique ID (this can be defined by vendor or software)	**Main Entry Field** (can have duplicated contents: not unique)	**Sub Entry Field**	**Locator Field** (can have multiple anchors)	**Switches/ Attributes**	**Cross Reference Target Field**
Plain entries of one level	*1001*	*Carbon monoxide*	(Nothing here: it is a simple main entry)	Anchor point within text	Off	
Entries with subheads	*1002*	*Carbon dioxide*	*medical gas supplies*	Anchor point within text	Off	
Cross references	*1003*	*Carbon filtration*		Anchor point within index file	On (to turn on See behavior)	*Activated carbon filtration*

Additional fields could be added if manufacturers want to include a system for indicating targeted page ranges in the text, based on highlighting text from one anchor point to another.

Some mechanisms for stemming and conflating words should be built into glossary, search, index and thesaurus functionality. As defined in Wikipedia, "stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form—generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root… Many search engines treat words with the same stem as synonyms as a kind of query broadening, a process called conflation."

An example of stemming would be the word "booking." A stemmed search would also include "book," "books," and "booked." When a reader highlights a word, or types a word, a mechanism should be provided to conflate the search. The provision of synonym lists or a lookup table would be the simplest step.

This issue may already be of concern to the dictionary group, as shown by the lemma notations in the spec outline at https://code.google.com/p/epub-revision/wiki/DictionariesGlossaries#Glossaries.

Sorting order, or alphabetization, will be an issue for multi-word main entries, an issue that surfaced in early online Help systems, and will most likely surface in ePub as well. Whether or not space characters and symbols are considered as characters, or are ignored, determines the sort. Two systems are in use: word-by-word, which honors the space, and letter-by-letter, which ignores it. *NISO TR03: Guidelines for the Alphabetical Arrangement of Letters and Sorting of Numerals and Other Symbols* can be a guideline. We recommend establishing that a space character precedes any other character in alphanumeric arrangements. Hyphens, dashes of any length, and slashes should be treated as space characters. A short example from Nancy Mulvany's *Indexing Books* shows the difference:

Word-by-word	**Letter-by-letter**
TYPE-ADF command	TYPE-ADF command
Type font	Typeface
Type foundry	Type font
Type metal	Type foundry
Typeface	Type metal
Typeset	Typeset

eReader manufacturers will need to know about this issue in order to present sorted indexes appropriately to their users.

Publisher workflow issues

It is most likely not the place of the ePub 3.0 Working Group to consider the current publishing workflows publishers use to get indexes into their products. But some information and background could be helpful to understand their workflow issues, and their output needs.

Indexes usually cannot be built until after the content of a piece is nearly complete. This is often a surprise to new publishers, who feel that the index for each chapter could be completed at the same time as the chapter is, without any further revision. Unfortunately, it doesn't work that way. The unique process of indexing requires that all references to a subject throughout a book be evaluated as a whole, so that terminology that best expresses each piece of the subject can be refined when compared to later references. Merging of very similar terms happens when all the similar terms are gathered into the compiled index.

Many people believe that an index is completed when the entries have all been created, but it takes time to then edit the rough index into a concise, merged, congruent and meaningful tool. Synonyms and main terms are evaluated and the semantic structure of cross references is analyzed and solidified at the end of the process.

Indexes are incorporated into publications in a variety of ways, and each method will need to find a path to both the fallback chapter-like index and the index data file:

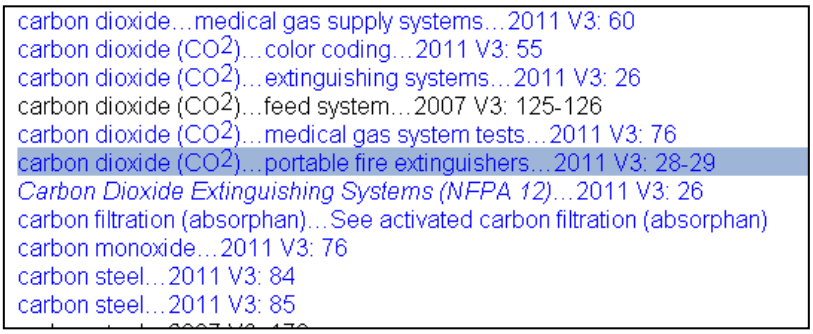
Standalone

Some are written as standalone files, referring to frozen page numbers in galleys, and have no active links to the text. These are currently being converted to ePub linked indexes with macros that hyperlink the page numbers and take the reader to the top of what was the print page. They are useful to a point, but can confuse the reader who has enlarged the font size, meaning that the actual term may be on a second page and not visible on the screen. The outputs from these files can be a standard book-like word processed file, a CSV file, or a tab-delimited file. XML content tags or styles can be applied to each level.

The file output needed to transform a standalone index to an ePub index would be: (a) a print-like chapter file with hyperlinks at least to page level, if not to paragraph level, and (b) indexing to anchor points instead of to page numbers, and converting to an XML, CSV or tab-delimited output file.

carbon dioxide
    medical gas supply systems…2011 V3: 60
carbon dioxide (CO^2)
    color coding…2011 V3: 55
    extinguishing systems…2011 V3: 26
    feed system…2007 V3: 125-126
    medical gas system tests…2011 V3: 76
    portable fire extinguishers…2011 V3: 28-29
Carbon Dioxide Extinguishing Systems (NFPA 12)…2011 V3: 26
carbon filtration (absorphan)…See activated carbon filtration (absorphan)
carbon monoxide…2011 V3: 76
carbon steel…2011 V3: 84

Standard output of standalone indexing

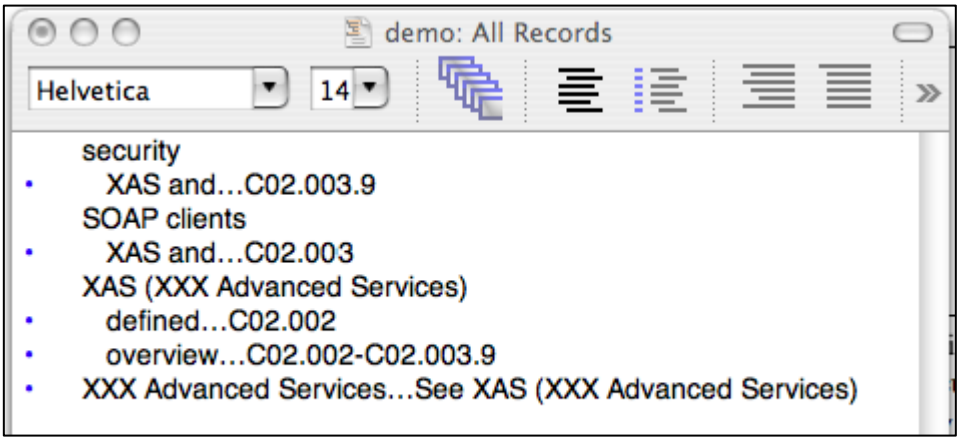


CSV or tab-delimited output of standalone indexing

Anchored

These are standalone index files that use anchor points or unique IDs in the content at the paragraph level for locators. These can easily be updated to refer to ePub anchor points, and can lead the reader to the paragraph for the entry. Again, these are not actively interconnected. Changes to one file can require changes to the other. These files would look the same as the samples above, but the locators would be replaced with anchor point codes.

The file output needed to transform an anchored index to an ePub index would be: (a) a print-like chapter file with hyperlinks to the paragraph level, at the anchor, and (b) converting to an XML, CSV or tab-delimited output file with the needed fields.



Indexing to unique IDs

Embedded

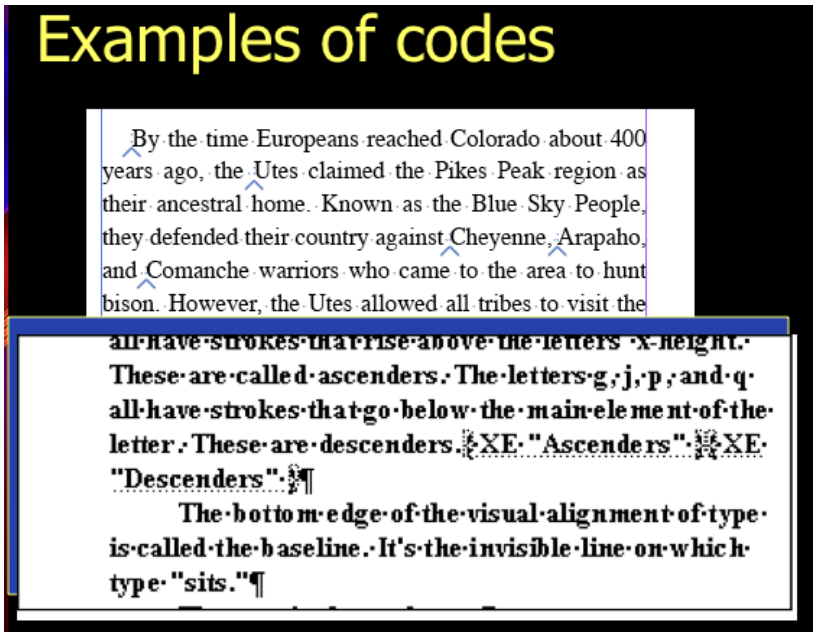
These are indexes that are generated by putting index codes for each entry into the publisher's chapter files. An example of tools that have embedding modules are InDesign, Frame, Word, and OpenOffice. The DocBook DTD, used by many publishers who create their books in XML, includes the <indexterm> and five child elements for creating index entries in DocBook XML files. Each of these tools uses a differing system for encoding the index entries, and compiling the final index; in the case of XML, each publisher has their own production process and specialized tools. If a publisher is outputting print, PDF, and ePub, this kind of indexing coding may be in the files, and the publisher may want to retain the indexing codes for future editions. A new system that includes adding anchor points at each index code's

location in the content, and linking the compiled index to the anchor points, will be needed. A similar process currently works in PDF exports for InDesign and Frame, and should be a matter of programming the tools to make it work.

The DocBook Project (which creates and disseminates the DocBook DTD and associated style sheets) has a beta version of style sheets to produce a complete EPUB book from a DocBook instance: http://sourceforge.net/projects/docbook/files/epub3 . At this time, it outputs the index as a separate file; it uses the nearest preceding section title as the text for the locator rather than page number. Clicking on the link in the index takes you straight to the exact location in the text. This locator style can be very confusing to readers, as it clutters a chapter-like display with long text strings that are hard to format into readable indents. But the mechanism is on the right path.

Publishers using embedded indexing need to have information about what they can currently do to convert an index from embedded codes to the index file needed in the ePub for display. Tab-delimited or CSV-delimited exports are not natively built into these page layout tools. Some publishers have further modified standard tools like Word or Frame to create proprietary systems, such as Cambridge University Press's CUP-XML indexing system.

The path to transform an embedded index to an ePub index would be: (a) a print-like chapter file with hyperlinks to the paragraph level, at the anchor, and (b) a separate export that can output an XML, XHTML, CSV- or tab-delimited output file with the needed fields.



InDesign markers (above) and Word XE fields (below)

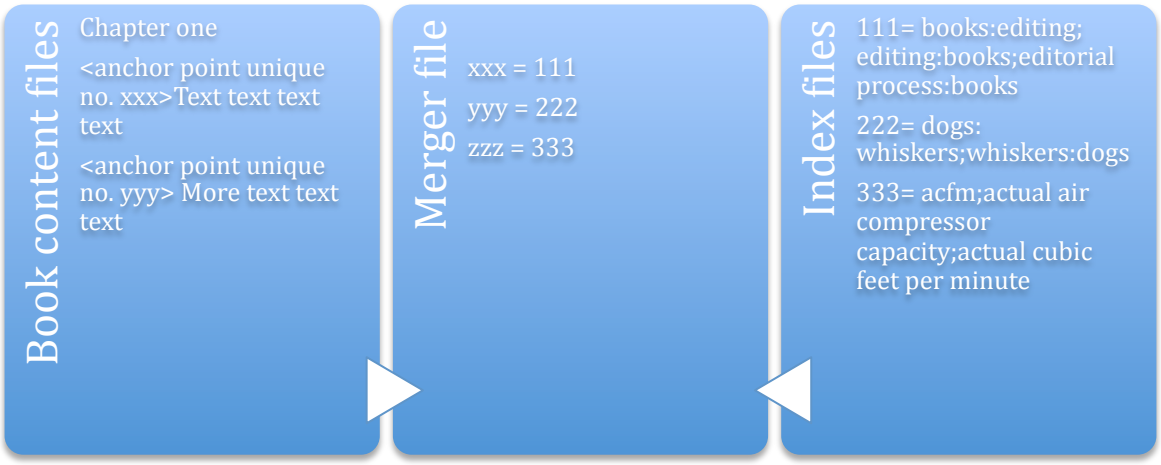
Dynamic

Also known as "meeting in the middle." Three files are involved:

- Content files with anchor codes or unique IDs
- Merger file, which connects anchor codes to sets of terms in the index file

- Index file, which groups similar concepts and synonyms into one unique id set

This style of indexing is exemplified by Microsoft's TIM tool and by taxonomy structures for large web sites. It acts more as a matchmaker, pulling locations and associations of terms together for display. The use of this system allows changes to happen in the book content files and the index file at the same time, removing the need for one group, either editorial personnel or indexing personnel, to have sole access to files for editing and updating.



The path to transform a dynamic index to an ePub index would be: (a) a means of exporting a print-like chapter file with hyperlinks to the paragraph level, at the anchor, and (b) a separate export that can output an XML, XHTML, CSV or tab-delimited output file with the needed fields, most likely pulled from the merger file.

All of these differing publishing workflows need a path to convert their content to ePub. Since each publisher is following a different path, many macro tools or small utilities will be needed to get from where the industry is now to a fluid workflow into ePub indexing.

Once the index file structures needed for displaying indexing in ePub 3.0 are established, the workflow tools can become a reality, since the output format and file structure needs to be established first.

ePub Index Working Draft

Use Cases

Accessibility

- Screen reader displays glossary definitions if available
- Screen reader displays a second section of index entries matching the user's entry data and an additional 3 lines of the index above, and 6 lines of the index below. This index section should have scrolling capabilities.
- Screen reader displays hits in context: lines of surrounding text with the term bolded.

User

- User reads as in a normal ebook
- User searches for terms or concepts
- User reads definitions (if available) and explores usage and thesaurus and returns to text, or:
- User selects items in index, and reviews displayed locations with terms in context.
- User is able at any time to set the default level of detail returned/displayed from the index.
- User chooses a specific hit from the index display, is taken to the paragraph, and term is near the top of display, or in the case of preserved page layouts, highlighted; or
- If no index results are available, user continues to type and full text search results are given for the term, displaying hits in context.
- User returns to reading mode until the need for search is repeated.
- If user re-enters search mode without highlighting a word, the last search term is still available, reducing user's work to retrieve the same term; or
- User enters search mode with new typed or highlighted term, and process begins anew.

Publication

- Publication contains a master index file linked to anchor points.
- Publication contains a chapter-like hyperlinked index at the end of the book for browsing, or as fallback.
- Publication may contain multiple chapter-like index files: name, product, subject
- Publication contains list of stemmed words (optional if the reading system can supply a universal list or the logic to perform stemming on the fly).
- Publications that are part of a series or collection will contain a file with links to other anchor points in other volumes.

User Agent / Reading System

- UA displays book text based on user interaction
- UA displays information not based on user interaction
- Sidebars, pop-ups, or help panels can be used

- UA parses dictionary/glossary/index information embedded in non-resource publication and provides alternate access or navigation. Stemmed words list can be incorporated.

Fallbacks

A chapter-based hyperlinked book-style index should be provided as a fallback. The hyperlinks should link to anchor points in the text.

Necessary Lexical/Terminological Concepts

- Concepts followed by '+' are baseline minimum requirements for basic functionality.

Indexes

- Headings
  - Main entry +
  - Subentry +
  - Locator +
    - Single locator +
    - Multiple locator strings +
    - Ranging
- Cross references
  - See-style cross references from unused terms to used terms +
  - See-also style cross references from used terms to other broader or related terms
  - Generic cross references from used terms to broad categories of terms (for example, "Commands. *See specific names of commands*")
- Sorting mechanisms
  - Alpha characters +
  - Numeric characters +
  - Symbol characters +
- Word stemming mechanisms (could be pulled from dictionary or glossary)
  - Gerund forms
  - Plural forms
  - Adjective or adverb forms

Indexing Issues

In the IDPF Wiki, it states the following:

"Indexing method *should not* be specified in 3.0, however spec *should* recommend that any resource used for indexing be placed into META-INF.

- Resources placed in META-INF may be unmanifested.
- Resources should be named with a leading reverse DNS qualifier e.g. org.idpf.resource.XXX"

https://code.google.com/p/epub-revision/wiki/DictionariesGlossaries#Indexing_Issues

Our team would like to understand the reasoning here, and have some input on this.

The index file should be referenced in the Navigation Document. As a navigation feature, the index should be a <nav> element, i.e., <nav pub:type="Indexmaster"> (*Note*: we are using the

term “indexmaster” to distinguish it from any other “index” types that appear elsewhere in the specification. Other terminology or naming schemes could be used.)

Package declaration

A publication can declare that it is a standalone index resource using the following syntax:

```
<meta property="publication.type">Indexmaster</meta>
```

Samples

(TBW – actual code samples. This will be developed after the project is accepted and work starts.)