

---

# Using DocBook5

To Produce PDF and ePub3 Books

Murray Maloney

## Abstract

Production notes from an academic book project designed to yield variations of a single work in print, ePub, and PDF book forms, between 2012 and 2016. The work is a multi-disciplinary study of the art and science of organizing. Murray contributed and edited content, participated in the design, and coordinated production.

He will talk about what they wanted to do, what they did in each successive edition, what didn't get done, what worked, what didn't, what they would do differently in retrospect, and what they would do the same, if they had it to do over.

## Table of Contents

Introduction .....	2
Book as Organizing System .....	2
The Structure of a Book .....	3
Front Matter .....	3
Body Matter .....	4
Rear Matter .....	6
Composition of Book Components .....	6
XHTML5 Composition .....	7
ePub3 Composition .....	9
Printed Page Composition .....	12
Copy fitting .....	13
Plain Old Text .....	14
Text Blocks .....	14
Paragraphs .....	14
Sidebars .....	18
Admonitions .....	19
Display Lists .....	21
Simple List .....	21
Itemized List .....	22
Ordered List .....	23
Glossary List .....	23
Variable List .....	25
Task and Procedure List .....	26
Tables, Figures, Examples .....	28
Informal table .....	28
Formal Table .....	29
Figure .....	29
Informal figure .....	30
Example .....	30
Informal example .....	32
Hypertext .....	32
Cross-references .....	32
Citations .....	36
Footnotes .....	37
Glossary terms and references .....	41
Links .....	43
Local Transclusion .....	43
Semantic Text .....	44
Why? .....	44

Lexical and Typographic Elements .....	45
Editorial Semantics .....	47
Bibliography and Citations .....	48
Glossary .....	50
Index .....	53
Index Terms in Paragraph Content .....	54
Index Terms in Info Elements .....	54
Index Processing .....	56
XHTML5 Output .....	57
Accessibility .....	61
Book Navigation .....	61
Bibliography Navigation .....	62
Glossary Navigation .....	62
Index Navigation .....	63
eBook Readers .....	63
Implementing Interactions .....	65
The Quiz .....	68
TDO Viewport .....	70
TDO Toggle .....	72
Build Process .....	72

## Introduction

The Discipline of Organizing is an academic work edited by UC Information School Adjunct Professor, Robert J Glushko, with contributions from a diverse collection of educators and students around the world. Between 2010 and 2012, he assemble and organized material from his course notes and slide decks, along with contributions from other professors and some graduate students. His goal was to produce an academic book in print and epub2 formats for publication by MIT Press. The first edition was published by MIT Press in 2013, in hard-cover print and ePub2 forms. The second through fourth editions were published for O'Reilly Media. PDF format editions of the third and fourth editions were made available for use for accessibility.

Murray Maloney joined the project in Spring of 2012, tasked with content editing. Later in 2012, he assumed responsibility for producing the first edition print and epub editions, using O'Reilly's Atlas system, which was subsequently used to produce an ePub3-only edition in 2014. In 2014 and 2015, Murray produced print and epub3 editions using customized XSLT scripts created by Bob Stayton and further developed by Jirka Kosek.

The following material is an elided version of Murray's Production notes. In its complete form, it was a hidden chapter in the book. He relied upon this chapter to test formatting, copy-fitting, floats, hypertext capabilities, The examples that illustrate DocBook5 input and XHTML5 output were intended primarily for the benefit of the production editor and the XSLT wizards. Many figures and examples have been removed because they would not have worked without the custom production pipeline, CSS, and JavaScript..

## Book as Organizing System

This book *qua* book project began for me in April 2012, when Bob Glushko showed me a draft of Chapter 1 of his then current project. I sat down and edited it that night; I handed it back to him in the morning. He asked if I would like to be his copy editor, and I responded that I would do it only if I could also be part of the entire production process, and build a suitable glossary, index and bibliography to accompany what I was starting to think of as “a manual for the obsessive compulsive.”

Bob and I have decades of experience working with hypertext and markup, so we saw single-source publishing to multiple print and ebook editions as an achievable goal. Although I had not been actively working in publishing for over a decade, I was interested to learn how one would go about making a book today. We started with ten chapters written and edited in Microsoft Word because, as a word processing tool, it was familiar to all of the writers. File sharing, editing, and annotating

was accomplished trivially over Dropbox. Bob and I have been exchanging our files over Dropbox since May 2012. We have been encouraged to adopt Git as our source repository, but have not been persuaded to do yet. We have lost files, but only temporarily, and we have managed to overcome occasional asynchronous saves by using oXygen's diff tools. So, the authors wrote while I edited copy and started planning how we would encode the book. Over the summer and fall of 2012, the textual content of the chapters and sections of the book coalesced.

Given our backgrounds, biases, connections, and knowledge of relevant standards we quickly narrowed in on DITA and DocBook5 as likely host languages to effect a single-source encoding of the book contents with structural and semantic markup. DITA was a serious contender over the summer of 2012, and many hours were spent studying up on its tag set, installing and testing its toolkit, under the tutelage of Eliot Kimber. Ultimately the decision to use DocBook5 was made based on the pragmatics of the availability of the Atlas publishing system, with which we could then use DocBook5 sources to produce PDF, epub2, and HTML manifestations of the book. Having O'Reilly Media behind the system was a big plus. (Producing epub3 books would come later.)

After the contents of the ten chapters were converted from Microsoft Word files to DocBook v4.5 files, I spent a couple of weeks cleaning up behind the conversion team, and then began making attempts to build the chapters of the book under Atlas. We discovered that there were some things that we expected out of DocBook5 that were not directly supported, or that could only be supported by customizing the XSLT transformation, and we also found out that there were other things that were supported. We adapted to our situation, in part by enlisting the aid of Bob Stayton, and we made the most of the standard tools at our disposal.

Transforming the DocBook5 chapter files into a book involves conversion into a set of roughly equivalent XHTML5 documents. Media-sensitive CSS style sheets provide formatting rules for the page layout, paragraphs, headings, images and on. The build process manufactures the table of contents and the index, and the rest of the book, whether fixed format print pages or variable layout ebooks is based on transformation of the source files. Aside from the layout of large text components in the reader's viewport, we had to identify and highlight some of the underlying meaning within the paragraphs and endnotes, by encoding titles and citations, acronyms and abbreviations, index entries. In order to develop a useful glossary, we would have to identifying term definitions for inclusion in the glossary, which would later become helpful when we decided to build in a quiz feature.

## The Structure of a Book

A book is a book, is a book. Whether it is presented as a manuscript or a bound volume, or through an electronic interface, a book is a book, with characteristic structural qualities that serve the needs of someone in the publishing chain, whether the typesetter, the publisher, the editor, the designer, the writer, or the interested reader.<sup>1</sup>

A book is a collection of conceptual elements that are typically arranged in progressive pattern, with a cover and title page at the beginning, followed by preliminary material of a legal and editorial nature, table(s) of contents, optionally some introductory forewords and prefaces, some chapters, which form the sum and substance of the work, potentially some ancillary material in appendixes, a bibliography, a glossary, index(es), and a terminal colophon.

This simplified subset of the DocBook5 model corresponds with the conventions of the Chicago Manual of Style. So, we started with these well-understood and time-tested architectural models of a book.<sup>2</sup>

## Front Matter

The front matter of the book includes the cover; half-title, title, and series pages; legal, copyrights, and publishing credits; a table of contents, and, optionally, some introductory material, such as a foreword, a preface, an abstract, and a dedication.

---

<sup>1</sup>The Art of the Book [<http://thevarsity.ca/2012/02/05/the-art-of-the-book/>]

<sup>2</sup>*The Chicago Manual of Style* and *DocBook5*.

In print, front matter is characterized by the use of roman numerals in folios, and the absence of folios on the title pages. The front matter is not considered to be part of the intellectual work; it is typically included to create context.

### Example 1. Front Matter: XHTML5

```
<body epub:type="frontmatter"
  data-type="book"
  epub:title="The Discipline of Organizing"
  epub:subtitle="... Edition"
  data-audience="LIS | Museums | Archives | ..."
  data-vendor="MITP | ORM | UC | ..."
  data-userlevel="Professional | Professor | Instructor |
    Graduate | Undergrad | Novice |
    Librarian | Curator | Archivist | ... "
  data-os="osx | linux | windows | jvm | ...
    ios | android | ...
    firefox | chrome | safari | opera | ...
    iBooks | Calibre | Lucifox | Readium | ..."
>
<section id="..."
  title="..."
  class="..."
  epub:type="abstract | acknowledgments | dedication |
    foreword | introduction | preamble | preface |
    halftitlepage | titlepage | seriespage |
    copyright-page
    toc | lot | loi "
  data-type="foreword | introduction | preface | dedication |
    halftitlepage | titlepage | seriespage"

  data-audience="..."
  data-userlevel="..."
  data-os="..." > [...]
```

## Body Matter

Chapters are the core content of the book. Chapters are typically ordered according to a model of progressive disclosure of information about a given topic or set of topics that comprise the intellectual value of the work. The other major components of the book are, essentially, metadata related to the sum or substance of the chapters. (The bibliography, glossary and index, for example, could not exist except in support of the content of the chapters. The front matter has no purpose but to introduce the chapters.)

Chapters contain subordinate sections, down to fourth-level numbered headings. When source files are transformed into XHTML5, the storage granularity shifts from chapter-level to the immediate children at the top section level. Thus a DocBook5 chapter `ch1.xml` with two subordinate sections is transformed into `ch1.xhtml` and `ch1s02.xhtml` in the ePub archive.

Chapters 1-12 must be included in any media representation of this work. However, discipline-specific variants of the chapters may accompany each of the editions of the work. Chapters are `Chaptern.xml`, where *n* is 1-12.

### Example 2. Chapter model: DocBook5

```
<chapter xmlns="http://docbook.org/ns/docbook" version="5.1"
  xml:id="PRODNOTES"
```

```
        userlevel="Editor Marker Producer Publisher" >
<title>Production Notes</title>
<info>[...]</info>
<para>...</para>
    [...]
<para>...</para>
<section xml:id="PRODNOTES.1">
    [...]
</section>
<section xml:id="PRODNOTES.2">
    <section xml:id="PRODNOTES.2.1">
        <section xml:id="PRODNOTES.2.1.1"> ... </section>
    </section>
</section>
</chapter>
```

### Example 3. Chapter model: XHTML5

```
<body epub:type="bodymatter"
    data-type="book"
    epub:title="The Discipline of Organizing"
    epub:subtitle="... Edition"
    data-audience="LIS | Museums | Archives | ..."
    data-vendor="MITP | ORM | UC | ..."
    data-userlevel="Professional | Graduate | ..."
    data-os="..." >
<section id="PRODNOTES"
    title="Production Notes"
    class="chapter Markup"
    epub:type="chapter"
    epub:label="Chapter"
    epub:ordinal="12"
    epub:title="Production Notes"
    epub:subtitle="..."
    data-type="chapter"
    data-audience="..."
    data-userlevel="..."
    data-os="..." ><p>...</p><p>...</p>
<section id="PRODNOTES.1"
    title="Production Notes"
    class="section"
    epub:type="subchapter"
    epub:label="Section"
    epub:ordinal="13.1"
    epub:title="Production Notes"
    epub:subtitle="..."
    data-type="sect1"
    data-audience="..."
    data-userlevel="..."
    data-os="..." ><p>...</p><p>...</p>
</section>
<section class="section" id="PRODNOTES.2"
    epub:type="subchapter" data-type="sect1">
    <section class="section" id="PRODNOTES.2.1"
        epub:type="division" data-type="sect2">
        <section class="section" id="PRODNOTES.2.1.1"
            epub:type="" data-type="sect3"> [...] </section>
    </section>
```

```
</section>
</section>
</body>
```

## Rear Matter

The rear matter of a book may be composed of ancillary and supporting materials. Supporting material is usually provided in the form of one or more afterwords, appendixes, a bibliography, a glossary, an index, and a colophon.

### Example 4. Rear Matter: DocBook5 and XHTML5

DocBook5: [...]

```
<body epub:type="rearmatter"
  data-type="book"
  epub:title="The Discipline of Organizing"
  epub:subtitle="... Edition"
  data-audience="LIS | Museums | Archives | ..."
  data-vendor="MITP | ORM | UC | ..."
  data-userlevel="Professional | Graduate | ..."
  data-os="..." >
  <section id="..."
    title="..."
    class="..."
    epub:type="afterword | appendix | conclusion |
      credits | colophon"
    epub:title="..."
    epub:subtitle="..."
    data-type="afterword | appendix | conclusion |
      acknowledgments | epilogue | colophon"
    data-audience="..."
    data-userlevel="..."
    data-os="..."
  > [...]
```

```
</section>
</body>
```

## Composition of Book Components

We start with a folder full of DocBook5 XML files, as described in the preceding section. The composition process is controlled by a `make(1)` file that accepts a group name as a parameter. That group name is one of many recorded in a configuration file described below.

The first step in composing an instance of this work is to consolidate and profile its constituent parts. Consolidation of the book involves reading the root `book.xml` file and resolving all of the `XInclude` references to the content files. Concurrently, we profile the content according to the settings of the profiling attributes: `@audience` on footnotes, paragraphs, and sidebars; `@condition` on media-specific content; and, `@userlevel` on other edition-specific content.

The build process is guided by a configuration file that lists all of these categories, and subsequently defines named groupings of one or more values from audience and userlevel. The build process begins with a group name as a parameter, determines which footnotes, paragraphs, and other content is to be included and excluded, and processes the DocBook5 source files to create a single DocBook5 file with the edition-specific subset of the book content.

The next step is an XSL transformation of the XML DocBook5 content into an XHTML5 equivalent, including: resolution of all cross-references and transclusion links; copying of images, figures,

pictures, scripts, and stylesheets; and, creation of the table of contents and index. The transformation process is aware of downstream requirements, so it includes appropriate attributes and values in the XHTML5 content that it produces, such as accessibility cues, navigation metadata, and ePub content type information. The result is a set of navigable XHTML5 documents, chunked at the section level.

Next, media-specific metadata files are packaged with the XHTML5 files as an ePub3 archive.

The process for creating printed pages is slightly different. Following an initial transformation to XHTML5, a page formatting engine, guided by a sophisticated CSS-Print stylesheet, resolves cross-references with page numbers, and lays out a properly hyphenated and justified text, with page headers, and folios using Arabic and roman numerals as appropriate. We have even been able to incorporate dictionary-style page headers into the glossary.

Arranging the pagination of paragraphs, footnotes, images, and other display elements to is a necessary task in any effort at book composition.

## XHTML5 Composition

The transformation process that we use to build this book creates a set of XHTML5 content files in a single folder. Included in that folder are css and js files to control appearance and behavior, respectively. The presentation of the book as an XHTML5 hypertext document is controlled by the browser, with guidance from the stylesheets and scripts. Whether hypertext links are blue and underlined or formatted otherwise is a function of the stylesheets. How a hypertext link behaves is a function of the browser.

Books viewed in a browser necessarily have a variable layout. We cannot know, for example, which typeface or size you and your browser have negotiated to be your default, nor how narrow or wide the browser viewport on your display. One of the features of web browsers is that they adapt to needs of the user. As a result, the stylesheets tend to be mere guides. In contrast, printed books, necessarily, have a fixed layout.

The root of the XHTML5 book is `index.html`. All of the XHTML5 files include a head and a body. The head includes links to the stylesheets, scripts, the table of contents, and to the previous and next resource. Previous and next links enable the reader to follow a linear path through the book regardless of ones' entry point to the book. Within the body of the document are intra-book links related to cross-references, citations, and so on, as well as external links.

In aid of precise understanding of the XHTML5 form of the electronic books, we employed the XHTML5 Vocabulary [<http://www.w3.org/1999/xhtml/vocab>] which “describes the items that are available in the default XHTML5 vocabulary space.” The vocabulary includes categories for “document metainformation, role attribute values, and items from the Accessible Rich Internet Applications Vocabulary (ARIA) vocabulary.” While somewhat useful, we found that ePub offered a much richer vocabulary.

We briefly considered the WAI-ARIA Roles Model [<http://www.w3.org/TR/wai-aria/roles>] and decided quickly that it was more about working with GUI interfaces than it was about the content of the book. In future editions, we expect to offer user interfaces to help the reader customize their experience of the book. To ensure that those interfaces are accessible, we will have to study this and seek guidance.

### Example 5. Book component: XHTML5

The basic structure of book components in the XHTML5 files generated by this process is modeled below. An html document has a head and a body child; the head contains link, meta, and script children; the body contains a header, a section, and a footer.

```
<html>
  <head>
```

```
<link [...] />
<meta [...] />
<script [...] > {... } </script>
</head>
<body data-type="book">
  <header>[...]</header>
  <section>[...]</section>
  <footer>[...]</footer>
</body>
</html>
```

At the granularity of chapters and sections, pages commonly include a header and a footer, whose content varies according to media type and capability. We discuss the composition of printed book page headers and footers later. In ebooks, or in web pages, the header and footer appear at the top and bottom of the page, respectively. Suggested uses include table of contents, navigational cues, and user interface controls.. The content of the header and footer could be computed during the build process, while the big giant brain holds the entire tree and all of its related links, or by post-processing the ePub3 archives and XHTML5 files. So, we could have a unique header and footer in each file.

The head of these XHTML5 documents contains LINK elements with information about top level components of the book, and sections in the current chapter. Scripts interact with this information to produce the navigational menus that are discussed later.

### Example 6. LINK element: XHTML5

```
<link rel="this chapter" href="ch01.xhtml" title="Chapter 1. ..." />
<link rel="this section" href="ch01s02.xhtml" title="1.2. ..." />

<link rel="section" href="ch01.xhtml#section-1.1" title="1.1. ..." />
<link rel="section" href="ch01s02.xhtml" title="1.2. ..." />
<link rel="section" href="ch01s03.xhtml" title="1.3. ..." />
<link rel="section" href="ch01s04.xhtml" title="1.4. ..." />

<link rel="next" href="ch01s03.xhtml" title="1.3. ..." />
<link rel="prev" href="ch01.xhtml#section-1.1" title="1.1. ..." />
<link rel="last" href="ch01s04.xhtml" title="1.4. ..." />
<link rel="up" href="ch01.xhtml" title="Chapter 1. ..." />

<link rel="chapter" href="ch01.xhtml" title="Chapter 1. ..." />
<link rel="chapter" href="ch02.xhtml" title="Chapter 2. ..." />
<link rel="chapter" href="ch03.xhtml" title="Chapter 3. ..." />
[...]
<link rel="chapter" href="ch11.xhtml" title="Chapter 11. ..." />

<link rel="cover" href="cover.xhtml" title="Cover" />
<link rel="titles" href="index.xhtml" title="Titles" />
<link rel="copyright-page" href="index.xhtml#Credits" title="Credits and Copyright" />
<link rel="dedication" href="index.xhtml#Dedication" title="Dedication" />
<link rel="foreword" href="pr01.xhtml" title="Foreword" />
<link rel="preface" href="pr02.xhtml" title="Preface" />
<link rel="abstract" href="pr03.xhtml" title="Abstract" />

<link rel="acknowledgments" href="pr04.xhtml" title="Acknowledgments" />

<link rel="bibliography" href="bi01.xhtml" title="Bibliography" />
<link rel="glossary" href="go01.xhtml" title="Glossary" />
<link rel="index" href="ix01.xhtml" title="Index" />
```



```
<link rel="colophon" href="co01.xhtml" title="Colophon" />
```

### Example 7. META element: XHTML5

We keep track of the count of footnotes in a set of meta elements; scripts use this information to manage presentation.

```
<meta name="notecount.LIS" content="3" />
<meta name="notecount.Law" content="1" />
<meta name="notecount" content="3" />
```

## ePub3 Composition

Like the XHTML5 book, the ePub3 book's format is somewhat variable. Laptops, tablets and smart phones come in all shapes and sizes, users can choose between portrait or landscape modes, day or night reading modes, smaller and larger font sizes, single or multiple column paged formatting, or full-width continuous scroll viewport. The reader software has more control over how you see this book than anything that we can do.

The good news is that the simplicity of the ePub3 format means that we can concentrate on delivering XHTML5 content enabled with semantic markup and scripts.

An ePub3 archive is a zip file whose constituent members must include a standard set of control files and a set of content files. Most publishers require that you validate ePub archives with `epubcheck3`. The archive contains a `META-INF` with `container.xml` and an `OEBPS` folder with the payload of the book's XHTML5 files and supporting image, style sheet, JavaScript files; notable among them are:

bk01-toc.xhtml	This is an XHTML5 file that contains a table of contents in ePub3 compliant NAV list format, with hidden entries below the top level of the book. The list can be expanded by applying CSS and/or JavaScript.
container.xml	Is the starting point for the ePub package.

### Example 8. ePub container.xml

```
<container xmlns="urn:oasis:names:tc:opendocument:xmlns:container" version="1.0">
  <rootfiles>
    <rootfile full-path="OEBPS/content.opf" media-type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

content.opf	This is an epub3 required data file that contains publication metadata in Dublin Core format, a manifest of files, a spine that maps the manifest items into a linear path, and a guide that references the cover and the table of contents.
index.xhtml	The preferred entry point to the book.
toc.ncx	This is an ePub3 required data file that contains navigation map that describes a linear path through the book.

### Example 9. ePub toc.ncx

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<ncx xmlns="http://www.daisy.org/z3986/2005/ncx/" version="2005-1">
  <head>
    <meta name="cover" content="cover"/>
    <meta name="dtb:uid" content="Z666.5.D57 2013"/>
  </head>
```

```
<docTitle>
  <text>The Discipline of Organizing</text>
</docTitle>
<navMap>
  <navPoint id="d0e1" playOrder="1">
    <navLabel>
      <text>The Discipline of Organizing</text>
    </navLabel>
    <content src="index.xhtml"/>
  </navPoint>
  <navPoint id="d0e1174" playOrder="2">
    <navLabel>
      <text>Foreword to the First Edition</text>
    </navLabel>
    <content src="pr01.xhtml"/>
  </navPoint>
  <navPoint id="d0e1307" playOrder="3">
    <navLabel>
      <text>Preface</text>
    </navLabel>
    <content src="pr02.xhtml"/>
  </navPoint>
```

In aid of precise understanding of the electronic books, we employ the EPUB 3 Structural Semantics Vocabulary [http://www.idpf.org/epub/vocab/structure/] which “defines a set of properties relating to the description of structural semantics of written works. While the vocabulary is generally host language agnostic, it has been constructed primarily to enable semantic inflection of elements in the HTML vocabulary.” The vocabulary includes categories for document partitions and divisions, titles and headings, sections and components, front matter, navigation, paragraphs, lists, tables, figures, notes and annotations, sentence types, credits, references, complementary content, learning and testing, reference sections, bibliography, glossary, index, colophon, and even pagination markers.

The EduPub Structural Semantics [http://www.idpf.org/epub/profiles/edu/structure/] “introduces additional usage requirements and restrictions not defined in the formal definitions in” EPUB 3 Structural Semantics Vocabulary [http://www.idpf.org/epub/vocab/structure/].

These vocabularies have parallels with DocBook5 and the Chicago Manual of Style.

### Example 10. TOC Nav: ePub3

The TOC nav is provided as the main Table of Contents for the work. It represents the reading order and hierarchical structure of the work, as nested lists of hypertext titles that lead to the corresponding components of the work. There must only be one nav with epub:type="toc" in the navigation document.

```
<nav id="nav-toc"
  class="toc"
  title="Table of Contents"
  epub:type="toc">
  <h2>Table of Contents</h2>
  <ol>
    <li><a class="xref" rel="foreword" rev="xref"
      href="..." title="Foreword">Foreword</a></li>
    <li><a class="xref" rel="preface" rev="xref"
      href="..." title="Preface">Preface</a></li>
    <li><a class="xref" rel="chapter" rev="xref"
      href="..." title="Chapter 1: ....">Chapter 1: ....</a>
    <ol hidden="">
      <li><a [...]>1.1: ...</a> </li>
      <li><a [...]>1.2: ...</a>
```

```
<ol hidden="">
  <li><a [...]>1.2.1: ...</a> </li>
  <li><a class="xref" rel="section" rev="xref"
    href="..." title="1.2.2: ...">1.2.2: ...</a> </li>
</li>
</li>
<li> [...] </li>
<li> [...] </li>
<li><a class="xref" rel="index" rev="xref"
  href="..." title="Index">Index</a></li>
</ol>
</nav>
```

### Example 11. Landmarks Nav: ePub3

The landmarks nav is provided for readers to present through their UI.

```
<nav id="nav-landmarks"
  class="landmarks"
  title="Landmarks"
  epub:type="landmarks">
  <h2>Landmarks</h2>
  <ol>
    <li><a class="xref" epub:type="toc"
      rel="toc" rev="xref" title="Table of Contents"
      href="bk01.xhtml#nav-toc">Table of Contents</a></li>
    <li><a class="xref" epub:type="loi"
      rel="loi" rev="xref" title="List of [Figures | Equations | Examples | ...]"
      href="bk01-toc.xhtml#lo-figures">List of Figures</a></li>
    <li><a class="xref" epub:type="bodymatter"
      rel="chapter" rev="xref" title="Foundations ..."
      href="ch01.xhtml">Foundations ...</a></li>
    [ ... ]
    <li><a class="xref" epub:type="bodymatter"
      rel="chapter" rev="xref" title="Roadmap ..."
      href="ch10.xhtml">Roadmap</a></li>
    <li><a class="xref" epub:type="bodymatter"
      rel="chapter" rev="xref" title="Case Studies"
      href="ch11.xhtml">Case Studies</a></li>
    <li><a class="xref" epub:type="bibliography"
      rel="bibliography" rev="xref" title="Bibliography"
      href="bi01.xhtml#lo-tables">Bibliography</a></li>
    <li><a class="xref" epub:type="glossary"
      rel="glossary" rev="xref" title="Glossary"
      href="go01.xhtml">Glossary</a></li>
    <li><a class="xref" epub:type="index"
      rel="index" rev="xref" title="Index"
      href="ix01.xhtml">Index</a></li>
  </ol>
</nav>
```

### Example 12. Page List Nav: ePub3

“The page-list nav element is a container for pagination information. It provides navigation to positions in the content that correspond to the locations of page boundaries present in a print source being represented by the EPUB Publication. The page-list nav element is optional in EPUB Navigation Documents and must not occur more than once. The order of li elements contained within a page-list nav structure must match the order of the actual pages inside each targeted EPUB Content Document

and must also follow the order of Content Documents in the Rendition's spine. The page-list nav element should contain only a single ol descendant (i.e., it should be a flat list, not a nested structure of navigation items).” (From ePub3 Content Documents [<http://www.idpf.org/epub/301/spec/epub-contentdocs.html#sec-xhtml-nav-def-types-pagelist>].)

```
<nav id="nav-pagelist"
      class="pagelist"
      title="Page List"
      epub:type="page-list">
  <h2>Page List</h2>
  <ol>
    <li> ... </li>
    <li> ... </li>
    <li> ... </li>
  </ol>
</nav>
```

## Printed Page Composition

We need to consider printed page composition. For the most part, you can think of the book having a frame, within which are its margins, its headers and footers, and a text area into which the content flows. There are traditions and guidelines, many of which have survived.

(We will discuss typographic devices that are employed within a page, and how we encode them, later.)

Margins	Left and right, top and bottom, recto and verso.	
Columnar Flow of Page Elements	Columnar flow is proceeding as expected.	
	One-Column	Main text body.
	Two-Column	Glossary page format.
	Three-Column	Index page format.
Chapter Page Headers and Footers	Running headers and footers in Chapters as follows:	
	Verso Header	Book Title
	Verso Footer	Page Number spacer Chapter #—Title
	Recto Header	Book Subtitle (Edition)
	Recto Footer	Section # - Title spacer Page Number
Front Matter Page Headers and Footers	Running headers and footers as follows:	
	Verso Header	Book Title
	Verso Footer	Page Number spacer [Major Section Title]
	Recto Header	Book Subtitle (Edition)
	Recto Footer	[Major Section Title] spacer Page Number
Bibliography, Glossary, Index Page Headers and Footers	Running headers and footers, in dictionary style, as follows:	
	Verso Header	First entry (.biblioentry[id], .glossterm, .primary)
	Verso Footer	Page Number spacer [Major Section Title]

Recto Header	Last entry
Recto Footer	[Major Section Title] spacer Page Number

Doing dictionary style page headers is a tough trick in the book making trade, which is why I wanted to attempt it with this book. In the first two editions, the stylesheet machinery was not up to the complexity of the bibliography and index markup. We pulled it off for the Glossary, though. In the later editions, with the help of Hakon Lie, we managed to get dictionary style headings for the Index.

## Copy fitting

Fitting text onto pages, whether for print media or for presentation through the viewport of an electronic device, remains a challenge. Even when faced with a paragraph that perfectly conveys its author's true intent, we may struggle with understanding due to unfortunate placement on the page. A paragraph may break near the end of a page, leaving a widow on one side and an orphan on the other. Figures and other display elements might force premature page breaks, or float away from their related contexts. Footnotes have to be gathered for presentation at the bottom of a page or at the end of a logical partition.

### Widow and Orphan Control

Managing the layout of text blocks so that they remain together on the same page, or even the same column, is a valuable typographic technique. Ideally, we want to avoid having a heading on one page and its subordinate paragraphs on the next, but it happens. We have learned to accept that today's ePub readers are not equipped to perform high-quality copy fitting.

In the print edition, we went through the book carefully, adding bits of white space here and there to force a paragraph, or a heading, or a figure onto the next page. We spent two days making small adjustments until we had something that was good enough to carry our names. Some will notice an occasional widow or orphan; alas, one can only endeavor to use the tools at hand to best advantage. I am satisfied enough to say that CSS has performed quite well. In spite of that, we needed to perform detailed pagination testing and copy fitting for print media.

### Floats

Several book sub-components have the quality that their position floats in the flow of the book. For example, many of the smaller sidebars in this book will shift position on a tablet display when the font size is increased or decreased.

In the print edition, we went through the book and found floats that not only needed their copy fitted, but the position of floats had to be tuned so that they were collocated with related text. Generally, an author or editor want related sidebars to be visible while reading the main point in the text flow. A misplaced sidebar, figure, or example can detract from the message and therefore the perceived quality of the work. As a consequence, we observe that page layout tools have not yet reached a level of typographic sophistication such that we can avoid the need to perform pagination testing.

### Endnotes

Chapter notes, section notes, paragraph notes, and pop-up notes are different presentation modalities for what we commonly know as footnotes.

In the print edition, footnotes are positioned as endnotes for each chapter.

In the early electronic editions, footnotes are positioned as endnotes for major sections of each chapter. The hypertext nature of footnote markers would permit collection of all endnotes into a single repository, but keeping all of the endnotes that relate to a given section at the end of that section creates an efficiency for the both the human and the electronic reader, by eliminating the need to load the footnotes and then reload the chapter.

**In newer electronic editions, footnotes are positioned as indented paragraph(s) immediately following the paragraph containing its reference. CSS and JavaScript will provide interactive control over footnote visibility.**

The selection of endnotes in a given edition is a function of the set of disciplines upon which that edition is based, which is controlled in a configuration file. The Professional Edition includes all of the discipline-specific endnotes, while the Core Concepts Edition contains none. Among the collection of graduate-level academic editions, for example, “Sensemaking” includes Philosophy, Linguistics, and Cognitive Science.

## Plain Old Text

Our principal concern with text is that it is expressible through the use of Unicode characters, that its lexical and semantic sub-components can be identified and distinguished, and that the presentation system is able to render those characters in a reasonably predictable facsimile of the author/editors' intent. For our purposes, the minimum threshold for printed and electronic visual presentation can almost be accomplished through the use of open source serif, sans serif and mono-spaced fonts with bold, italic and bold-italic variants in a range of sizes from 8 point to 36 point.

We assume that most readers will want to choose their preferred typefaces while reading the ebooks, so we limited our range to serif for text, sans-serif for labels and headings, and monospaced for code. Likewise, we limited our range of emphasis to italic, bold, and bold italic. We have avoided the use of colored text, except to distinguish between external links and in-book cross-references.

In order express some of the characters used in this book, we require a Unicode font, as in our reference to the classic Japanese film *Kumonosu-jo* (#####) (*Throne of Blood*) [(Kurosawa 1957)].

## Text Blocks

### Paragraphs

The paragraph is a container for a group of one or more sentences. The principal goal of the paragraph is to be readable and to distinguish itself from surrounding paragraphs and other compositional elements. That is, one should be able to readily discern the beginning and ending of a paragraph.

In Western languages, the flow of text in a paragraph is from left-to-right, until the end of the line is reached and then recommencing at the beginning of a new line. Many composition systems employ hyphenation and line justification to manage word breaks and white space within lines. In some presentations, paragraphs may include highlights, such as numbering and leading capital letters. While ornate styling of paragraphs may be suitable in other genres, we chose to avoid employing such typographic devices in an academic work.

**Example 13. Paragraph: DocBook5**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Urt enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
<para>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua. Urt enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute
    irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
    nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
    culpa qui officia deserunt mollit anim id est laborum.
```

```
</para>
```

An alternate encoding for paragraphs, which is discussed elsewhere, is to add `@audience` or `@userlevel` values to designate the paragraph for profiling by the build process when producing custom editions

```
<para userlevel="Graduate">Lorem ipsum dolor sit amet, ...</para>
```

**Example 14. Paragraph: XHTML5**

```
<p>Lorem ipsum dolor sit amet, ...</p>
```

```
<p class="LIS" data-audience="LIS" data-userlevel="Graduate">Lorem ipsum dolor sit amet, ...</p>
```

## Headings

Chapter and section headings are numbered, as are formal examples, figures, and tables. Sidebars, admonitions, and informal examples, figures and tables are not numbered. We chose to use identifiers that reflect the heading type and its numerate prefix. Rather than relying on an automated process to number chapters and sections, we chose to make the labels explicit, thereby facilitating look-up and cross-referencing.

Chapter and section headings are numbered in all media. Formal table, figure and example headings are numbered in all media. Informal tables, figures and examples lack headings in all media. Sidebar headings are unnumbered in all media.

**Example 15. Heading: DocBook5**

```
<chapter xml:id="chapter-1"
    xmlns="http://docbook.org/ns/docbook" version="5.1">
  <title>Foundations for Organizing Systems</title>
  <para>...</para>
```

```
<section xml:id="PRODNOTES.3">
  <title>Text Blocks</title>
  <section xml:id="PRODNOTES.3.1">
    <title>Paragraphs</title>
    <para>...</para>
```

```
<sidebar xml:id="sidebar-1.2.1-Concert-Tickets">
  <title>Concert Ticket</title>
  <para>...</para>
```

```
<example>
```

```
<title>Paragraph example: DocBook5</title>
<literallayout>...</literallayout>
</example>
```

Technical aside: In this example, the inclusion of the DocBook5 namespace declaration and version identifier reflects the fact that `<chapter>` element is the level of granularity at which we edit this book. We use the DocBook5 namespace declaration in order to signal to any XML tools that this document contains elements in that namespace. Our editing tools, for example, can apply CSS style sheets and schema-based validation scenarios, while our build tools can apply associated XSLT transformation rules.

## Example 16. Heading: XHTML5

```
<section id="PRODNOTES"
  title="Chapter 13. Production Notes"
  class="chapter"
  epub:type="chapter"
  epub:label="Chapter"
  epub:ordinal="13"
  epub:title="Production Notes"
  epub:subtitle="..."
  data-type="chapter" >
<div class="titlepage">
  <div>
    <div>
      <h1 class="title"><span epub:type="label">Chapter</span>
        <span epub:type="ordinal">12</span>
        <span epub:type="title">Production Notes</span>
      </h1>
    </div>
  </div>
</div>

<section class="section"
  epub:type="subchapter"
  epub:label=""
  epub:ordinal="13.3"
  epub:title="Text Blocks"
  epub:subtitle="..."
  data-type="sect1"
  id="PRODNOTES.3"
  title="13.3. Text Blocks">
<div class="titlepage">
  <div>
    <div>
      <h2 class="title"><span epub:type="label"></span>
        <span epub:type="ordinal">12.3</span>
        <span epub:type="title">Text Blocks</span></h2>
    </div>
  </div>
</div>

<section class="section"
  epub:type="division"
  epub:label=""
  epub:ordinal="13.3.1"
  epub:title="Paragraphs"
  epub:subtitle="..."
  data-type="sect2"
```



```
        title="13.3.1. Paragraphs"
        id="PRODNOTES.3.1">
<div class="titlepage">
  <div>
    <div>
      <h3 class="title"><span epub:type="label"></span>
        <span epub:type="ordinal">12.3.1</span>
        <span epub:type="title">Paragraphs</span></h3>

    </div>
  </div>
</div>
[...]
```

**A Bridge Heading**

Sometimes a set of paragraphs really needs to be set apart with headings, but it seems hard to justify adding new hierarchical sections. That is where a bridge heading can be useful.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Urt enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi urt aliquip ex ea commodo consequat.

## Another Bridge Heading

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat endente non proident, feugiat in culpa qui officia deserunt mollit anim id est laborum.

### Example 17. Bridge Heading: DocBook5

```
<bridgehead>A Bridge Heading</bridgehead>

<para>Sometimes a set of paragraphs really needs to be set apart with headings,
  but it seems hard to justify adding new hierarchical sections.
  That is where a bridge heading can be useful,
  as in the <xref linkend="Preface"/>.</para>
<para>Lorem ipsum dolor sit amet, ...</para>
<bridgehead>Another Bridge Heading</bridgehead>
<para>Duis aute irure dolor ...</para>
```

### Example 18. Bridge Heading: XHTML5

```
<h4 class="bridgehead">A Bridge Heading</h4>
<p>Sometimes a set of paragraphs...</p>
```

```
<p>Lorem ipsum dolor sit amet, ...</para>
<h4 class="bridgehead">Another Bridge Heading</h4>
<p>Duis aute irure dolor ...</p>
```

## Sidebars

We use sidebars extensively throughout the book. In the Web World, they are more commonly known as asides. We use sidebars as editorial asides, to draw the readers attention to a related topic. Some are designed as “Stop and Think” exercises, others highlight an important definition, still others contain captioned graphic images. Some sidebars are designated by discipline and user level for inclusion/exclusion. Some sidebars are designated for inclusion only in the Academic Editions. Some sidebars are designated for inclusion only in the Instructor's Edition.

### Example 19. Sidebar: DocBook5

```
<para>Some sidebars, as you see,...</para>

<sidebar><title>Another Sidebar</title>
  <?dbhtml sidebar-width="45%"?>
  <?dbhtml float-type="right"?>
  <para>Sidebar paragraph. Lorem ipsum dolor sit amet, ...</para>
</sidebar>

<para>This text is only here...</para>
<para>This text is only here...</para>
```

You would be right to wonder what that `<?dbhtml stuff` is. It is a processing instruction that tells the processor some information that is not available through DocBook5 attributes. (It seems kinda silly to allow `@scale` on an image and not allow float placement information, but there it is. Separation of presentation from content, or so they say. It can be taken too far, and has in this case. So that is why we have to resort to using processing instructions rather than the sidebar element's attributes.)

We designate quite a few sidebars as “Stop and Think” exercises for the academic editions only:

```
<sidebar xml:id="StopAndThink-1.3.1.2-AnalogyTests"
  userlevel="Editor Professor Instructor Graduate Undergraduate">
  <title>Stop And Think: Organizing System Analogy Tests</title>
  <para>Wild Animal Park is to Natural History Museum as A City is to... ?</para>
  [...]
</sidebar>
```

### Example 20. Sidebar: XHTML5

```
<aside id="StopAndThink-1.3.1.2-AnalogyTests"
  title="Stop And Think: Organizing System Analogy Tests"
  class="sidebar"
  epub:type="sidebar"
  data-type="sidebar">
  <div class="titlepage">
    <div><div>
      <div class="sidebar-title">Stop And Think: Organizing System Analogy Tests</div>
    </div></div>
  </div>
  <p>Wild Animal Park is to Natural History Museum as A City is to... ?</p>
  [...]
</aside>
```

## Admonitions

Admonitions are a class of typographic device that is used frequently in technical publishing, from consumer product manuals to industrial, scientific, and chemical catalogs, the pattern has become recognizable. Earlier incarnations of admonitions in text used the annotation, **N.B.** or **n.b.**, as an abbreviation of the Latin *nota bene*. The heading text and color combinations may vary, but the message is clear: “**Pay attention to this!**” It is possible to associate icons with the various category of admonition, but we decided to avoid the bling. We only use admonitions for editor's notes, important navigation advisories, and production warnings, but we present them all here for your edification.

A note:

### **Note**

In this chapter, we observe that notes are intended for editorial annotation, as at the beginning of the bibliography, glossary, and index..

A tip:

### **Tip**

Here's a tip for you: Don't take any wooden nickels.

An important note:

### **Important**

Pay attention to this...

We use important admonitions as a container for way-finding information in many chapters.

A caution:

### **Caution**

Be careful about this...

A warning:

### **Warning**

*Caveat lector!*

## Example 21. Admonition: DocBook5

```
<para>A note:</para>
<note>
  <para>This is interesting...</para>
</note>

<para>A tip:</para>
<tip>
  <para>Here's a tip for you: Don't take any wooden nickels.</para>
</tip>

<para>An important note:</para>
<important>
```

```
<para>Pay attention to this...</para>
</important>
```

```
<para>A caution:</para>
<caution>
  <para>Be careful about this...</para>
</caution>
```

```
<para>A warning:</para>
<warning>
  <para>Caveat lector!</para>
</warning>
```

## **Nota bene**

The heading of any of these admonitions can be substituted thus:

```
<note><title>Nota bene</title>
  <para>The heading of any of these admonitions can be substituted thus:</para>
  [...]
</note>
```

## **Example 22. Admonition: XHTML5**

```
<p>A note:</p>
<div class="note"
  data-type="note"
  epub:type="notice"
  title="Note">
  <h3 class="title">Note</h3>
  <p class="editorial">In this chapter, we observe that notes are intended for editorial annotation,
    as at the beginning of the bibliography, glossary, and index..</p></div>
```

```
<p>A tip:</p>
<div class="tip"
  data-type="tip"
  epub:type="notice"
  title="Tip">
  <h3 class="title">Tip</h3>
  <p class="editorial">Here's a tip for you: Don't take any wooden nickels.</p>
</div>
```

```
<p>An important note:</p>
<div class="important"
  data-type="important"
  epub:type="notice"
  title="Important" >
  <h3 class="title">Important</h3>
  <p class="editorial">Pay attention to this... </p>
  <p class="editorial Markup">We use important admonitions as a
    container for way-finding information in many chapters.</p></div>
```

```
<p>A caution:</p>
<div class="caution"
  data-type="caution"
  epub:type="notice"
  title="Caution">
```

```
<h3 class="title">Caution</h3>
<p class="editorial">Be careful about this...</p>
</div>

<p>A warning:</p>
<div class="warning"
  data-type="warning"
  epub:type="warning"
  title="Warning">
  <h3 class="title">Warning</h3>
  <p class="editorial"><span class="foreignphrase"
    ><em class="foreignphrase">Caveat lector!</em></span>
  </p>
</div>

<div id="..."
  title="Nota bene"
  class="note"
  epub:type="notice"
  data-type="note" >
  <h3 class="title">Nota bene</h3>
  <p>The heading of any of these admonitions can be substituted thus: </p>
  [...]
</div>
```

## Display Lists

In writing and speech, we often relate two words or terms with a conjunction or disjunction, such as “and” or “but”. When we add a third or fourth word, term, or expression, we use a comma to distinguish the members of a list; we can also use a semi-colon, when we want to distinguish among a list of clauses. However, our cognitive ability to recognize the distinct members of a complex set of inter-dependent clauses can become strained; we witness this often in legal, statutory, and tax documents; it seems as though the writer is testing our patience and our ability to recognize subtle nuance.

We use display lists when we want to signal sharp distinctions that may be harder to achieve in running text of a paragraph or set of paragraphs.

### Simple List

The simple list is a construct whose members are simple text strings; that is, the members may contain text and phrase-level markup, but no paragraph-level constructs. In this work, we have used it mainly for its typographic effect, not as a logical container.

A simple list:

```
first member
second member
third member
```

#### Example 23. Simple list: DocBook5

```
<simplelist>
  <member>first member</member>
  <member>second member</member>
  <member>third member</member>
</simplelist>
```

**Example 24. Itemized list: XHTML5**

```
<p>A simple list:</p>
<table style="border: 0; " class="simplelist" epub:type="list">
  <tr><td>first member</td></tr>
  <tr><td>second member</td></tr>
  <tr><td>third member</td></tr>
</table>
```

## Itemized List

An itemized list is a construct whose list items may contain paragraph-level elements. The list items are presented in the same order as they appear in the source files; in an itemized list we assume that the order of the list items is not relevant. We use itemized lists to distinguish among a series of related paragraphs whose relatedness may not be as obvious if presented in running text. Itemized lists are the most commonly used form of display list in this work.

An itemized list:

- first bullet
- second bullet
- third bullet

**Example 25. Itemized List: DocBook5**

```
<itemizedlist>
  <listitem>
    <para>first bullet</para>
  </listitem>
  <listitem>
    <para>second bullet</para>
  </listitem>
  <listitem>
    <para>third bullet</para>
  </listitem>
</itemizedlist>
```

**Example 26. Itemized list: XHTML5**

```
<p>An itemized list:</p>
<div class="itemizedlist" epub:type="list">
  <ul class="itemizedlist" style="list-style-type: disc; ">
    <li class="listitem" epub:type="list-item">
      <p>first bullet</p>
    </li>
    <li class="listitem" epub:type="list-item">
      <p>second bullet</p>
    </li>
    <li class="listitem" epub:type="list-item">
      <p>third bullet</p>
    </li>
  </ul>
</div>
```

&lt;/div&gt;

## Ordered List

An ordered list is a construct whose list items may contain paragraph-level elements. An ordered list should only be used if the order is important, or if the list labels will be used as referents in the text. The enumerated list items are presented in the same order as they appear in the source files. We use ordered lists to distinguish among a set of related paragraphs whose enumeration may not be as obvious if presented in running text. Ordered lists are not a commonly used form of display list in this work.

An ordered list:

1. Number one.
2. Number two.
3. Number three.

### Example 27. Ordered List: DocBook5

```
<orderedlist>
  <listitem>
    <para>Number one. </para>
  </listitem>
  <listitem>
    <para>Number two. </para>
  </listitem>
  <listitem>
    <para>Number three. </para>
  </listitem>
</orderedlist>
```

### Example 28. Ordered List: XHTML5

```
<p>An ordered list:</p>
<div class="orderedlist" epub:type="list">
  <ol class="orderedlist" type="1">
    <li class="listitem" epub:type="list-item">
      <p>Number one. </p>
    </li>
    <li class="listitem" epub:type="list-item">
      <p>Number two. </p>
    </li>
    <li class="listitem" epub:type="list-item">
      <p>Number three. </p>
    </li>
  </ol>
</div>
```

## Glossary List

A glossary list is a set of terms and their definitions. A glossary list is a relatively formal type of list because the definition of terms places requirements upon its structure and format. A glossary list is a set of matched pairs; a term and a definition. We assign identifiers to glossary entry, terms, and definition, to enable cross-references and transclusion, as we will discuss further later. There are localized glossary lists, like the one that follows, and there is a master glossary. Keeping them in sync is a bit complex.

What follows is an example of a glossary list occurring within the context of a chapter; it defines three terms and assigns identifiers to the terms, and definitions.

Lexical tagging	Tagging content with the purpose of identifying lexical components of the text. The wordasword, phrase, and symbol elements are examples of lexical tagging.
Semantic tagging	Tagging content with the purpose of identifying some aspect of its semantics. The personname, hardware, and URI elements are examples of semantic tagging.
Typographic tagging	Tagging content with the purpose of affecting the typographic styling. The emphasis element is an example of typographic tagging.

### Example 29. Glossary List example

```
<glosslist>
<glossentry>
  <glossterm xml:id="term_lexical_tagging">Lexical tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_lexical_tagging" role="definition">Tagging
      content with the purpose of identifying lexical components of
      the text. The wordasword, phrase, and symbol elements are
      examples of lexical tagging. </phrase></para>
    </glossdef>
  </glossentry>
<glossentry>
  <glossterm xml:id="term_semantic_tagging">Semantic tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_semantic_tagging" role="definition">Tagging
      content with the purpose of identifying some aspect of its
      semantics. The personname, hardware, and URI elements are
      examples of semantic tagging. </phrase></para>
    </glossdef>
  </glossentry>
<glossentry>
  <glossterm xml:id="term_typographic_tagging">Typographic tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_typographic_tagging" role="definition">Tagging
      content with the purpose of affecting the typographic styling.
      The emphasis element is an example of typographic tagging.
    </phrase></para>
    </glossdef>
  </glossentry>
</glosslist>
```

### Example 30. Glossary List: XHTML5

```
<div class="glosslist">

  <dl epub:type="glossary">
    <dt epub:type="glossterm" class="glossterm"
      id="term_lexical_tagging"
    ><dfn>Lexical tagging</dfn></dt>
    <dd class="glossdef" epub:type="glossdef">
      <p><span id="def_lexical_tagging" class="definition" role="definition">Tagging
        content with the purpose of identifying lexical
        components of the text. The wordasword, phrase, and
```



```
        symbol elements are examples of lexical tagging.</span></p>
    </dd>
</dl>

<dl epub:type="glossary">
  <dt epub:type="glossterm" class="glossterm"
    id="term_semantic_tagging"
    ><dfn>Semantic tagging</dfn></dt>
  <dd class="glossdef" epub:type="glossdef">
    <p><span id="def_semantic_tagging" class="definition" role="definition">Tagging
      content with the purpose of identifying some aspect of
      its semantics. The personname, hardware, and URI
      elements are examples of semantic tagging.</span></p>
  </dd>
</dl>

<dl epub:type="glossary">
  <dt epub:type="glossterm" class="glossterm"
    id="term_typographic_tagging"><dfn>Typographic
    tagging</dfn></dt>
  <dd class="glossdef" epub:type="glossdef">
    <p><span id="def_typographic_tagging" class="definition" role="definition">Tagging
      content with the purpose of affecting the typographic
      styling. The emphasis element is an example of
      typographic tagging. </span></p>
  </dd>
</dl>
</div>
```

## Variable List

A variable list is similar to a glossary list, but it is less formal. Its structure allows each entry to have one or more terms, followed by its list item, which may contain one or more paragraph-level constructs. We tend to use variable lists in situations where an enumeration of terms and related descriptions is required, rather than formal definitions.

A variable list:

First term, Second term, Third term	One description
Fourth term	Another description
Fifth term, Sixth term	Yet another description

### Example 31. Variable List: DocBook5

```
<variablelist>
  <varlistentry>
    <term>First term</term>
    <term>Second term</term>
    <term>Third term</term>
    <listitem>
      <para>One description</para>
    </listitem>
  </varlistentry>
```

```
<varlistentry>
  <term>Fourth term</term>
  <listitem>
    <para>Another description</para>
  </listitem>
</varlistentry>
<varlistentry>
  <term>Fifth term</term>
  <term>Sixth term</term>
  <listitem>
    <para>Yet another description</para>
  </listitem>
</varlistentry>
</variablelist>
```

### Example 32. Variable list: XHTML5

```
<p>A variable list: </p>
<div class="variablelist" epub:type="list">
  <dl class="variablelist">
    <dt><span class="term">First term<br /></span>      <!-- Why are <br> inside of <span>? -->
      <span class="term">Second term<br /></span>
      <span class="term">Third term</span> </dt>
    <dd><p>One description</p></dd>

    <dt><span class="term">Fourth term</span></dt>
    <dd><p>Another description</p></dd>

    <dt><span class="term">Fifth term<br /></span>
      <span class="term">Sixth term</span> </dt>
    <dd><p>Yet another description</p></dd>
  </dl>
</div>
```

## Task and Procedure List

The task and procedure list presents a formal structure for relating a procedure as a set of steps, optionally within the context of a task. We employ procedure lists in a couple of chapters.

### Task List

#### Task Summary

The task at hand, as it were, is to follow along with the Boy Scout song [<http://www.scoutsongs.com/lyrics/hokeypokey.html>].

#### Task Prerequisites

Before you get started, you should be with a group of friends or fellow revelers. Musical accompaniment is optional.

#### Procedure 1. Procedure List

1. You put your right hand in,
2. You put your right hand out,

3. You put your right hand in,
4. And you shake it all about
5. You do the hokey pokey and you turn yourself around.
6. Repeat from step 1, substituting “left hand,” “right foot,” “left foot,” “head,” “butt,” and “whole self.”

### Task-related information

That's what it's all about.

### Example 33. Task/Procedure List: DocBook5

```
<task role="instructive">
  <title>Task List</title>
  <tasksummary>
    <title>Task Summary</title>
    <para>The task at hand, as it were, is to follow along with the
      <link xmlns:xlink="http://www.w3.org/1999/xlink"
        xlink:href="http://www.scoutsongs.com/lyrics/hokeypokey.html"
      >Boy Scout song</link>.</para>
  </tasksummary>
  <taskprerequisites>
    <title>Task Pre-Requisisites</title>
    <para>Before you get started, you should be
      with a group of friends or fellow revelers.
      Musical accompaniment is optional.</para>
  </taskprerequisites>

  <procedure role="instructive">
    <title>A Procedure List</title>
    <step><para>You put your right hand in,</para></step>
    <step><para>You put your right hand out,</para></step>
    <step><para>You put your right hand in,</para></step>
    <step><para>And you shake it all about</para></step>
    <step><para>You do the hokey pokey and you turn yourself around.</para></step>
    <step><para>Repeat from step 1, substituting
      <quote>left hand,</quote>
      <quote>right foot,</quote>
      <quote>left foot,</quote>
      <quote>head,</quote>
      <quote>butt,</quote> and
      <quote>whole self.</quote>
    </para>
  </step>
</procedure>

  <taskrelated>
    <title>Task-related information</title>
    <para>That's what it's all about.</para>
  </taskrelated>
</task>
```

### Example 34. Procedure List: XHTML5

```
<div class="task" title="Task List">
```

---

```
<div class="task-title">Task List</div>
<div class="tasksummary">
  <div class="tasksummary-title">Task Summary</div>
  <div class="tasksummary-contents">
    <p>The task at hand, as it were, is to follow along with the
    <span class="link"><a class="link" target="_top"
      href="http://www.scoutsongs.com/lyrics/hokeypokey.html"
    >Boy Scout song</a></span>.</p>
  </div>
</div>
<div class="taskprerequisites">
  <div class="taskprerequisites-title">Task Prerequisites</div>
  <div class="taskprerequisites-contents">
    <p>Before you get started, you should be with a group
      of friends or fellow revelers. Musical accompaniment is optional.</p>
  </div>
</div>
<div class="procedure" title="Procedure 12.1. Procedure List">
  <div class="procedure-title">Procedure 12.1. Procedure List</div>
  <ol class="procedure" type="1">
    <li class="step" title="Step 1">
      <p>You put your right hand in,</p></li>
    <li class="step" title="Step 2">
      <p>You put your right hand out,</p></li>
    <li class="step" title="Step 3">
      <p>You put your right hand in,</p></li>
    <li class="step" title="Step 4">
      <p>And you shake it all about</p></li>
    <li class="step" title="Step 5">
      <p>You do the hokey pokey and you turn yourself around.</p></li>
    <li class="step" title="Step 6">
      <p>Repeat from step 1, substituting
        <span class="quote">“left hand,”</span>
        <span class="quote">“right foot,”</span>
        <span class="quote">“left foot,”</span>
        <span class="quote">“head,”</span>
        <span class="quote">“butt,”</span>
        and <span class="quote">“whole self.”</span></p></li>
  </ol>
</div>
<div class="taskrelated">
  <div class="taskrelated-title">Task-related information</div>
  <div class="taskrelated-contents">
    <p>That's what it's all about.</p>
  </div>
</div>
</div>
```

## Tables, Figures, Examples

### Informal table

An informal table

1	2	3
4	5	6

7	8	9
---	---	---

## Formal Table

A formal CALS table:

**Table 1. Table Title**

1	2	3
4	5	6
7	8	9

## Figure

Figures are used to present core graphic content. Figures are numbered and titled. Primary and secondary subject matter descriptions are provided through ALT text and caption. Credit for the graphic or image is provided along with the caption.

### Example 35. Figure: DocBook5

```
<figure >
  <title>Figure with image, descriptive text, and caption</title>
  <mediaobject>
    <imageobject>
      <imagedata fileref="figs/Figure-1.1.jpg" format="JPG"/>
    </imageobject>
    <textobject>
      <phrase role="ALT descriptive">A conceptual representation
        of an organizing system. [...]</phrase>
    </textobject>
    <caption>
      <para><phrase role="caption">An organizing system is
        [...]</phrase> </para>
      <para>A figure is labeled with a numbered heading. [...] </para>
    </caption>
  </mediaobject>
</figure>
```

### Example 36. Figure: XHTML5

```
<figure id="..."
  title="Figure with image, descriptive text, and caption"
  class="figure" >
  <div class="figure-title">Figure 12.1.
    Figure with image, descriptive text, and caption</div>
  <div class="figure-contents">
    <div class="mediaobject">
    <div class="caption">
      <p class="CORE"><span class="caption">An organizing system is [...]</span></p>
      <p>A figure is labeled with a numbered heading [...]</p>
    </div>
  </div>
</div>
```

</figure>

## Informal figure

An informal figure follows. An informal figure has all of the characteristics of a figure except for a numbered title. Other than this example, this work does not include any informal figures. It is included here for completeness.

Descriptive text is intended as a text alternative to the image for use by screen readers and other assistive technology, following guidelines established by WAI. The descriptive text reflects the primary subject matter.

### Example 37. Informal Figure: DocBook5

The markup to achieve this follows:

```
<informalfigure>
  <mediaobject>
    <imageobject>
      <imagedata fileref="Pictures/1.2.1-Tickets.JPG" format="JPG" width="120" />
    </imageobject>
    <textobject>
      <phrase role="ALT descriptive">A collage of concert tickets. [...] </phrase>
    </textobject>
    <caption>
      <para>A ticket is a set of [...] </para>
      <para><phrase role="credit">(Photo by [...].)</phrase></para>
    </caption>
  </mediaobject>
</informalfigure>
```

### Example 38. Informal figure: XHTML5

```
<div class="informalfigure Markup ">
  <div class="mediaobject">
    
  </div>
  <div class="caption">
    <p>A ticket is a set of [...]</p>
    <p><span class="credit">(Photo by [...].)</span></p>
  </div>
</div>
```

### Example 39. Informal Figure: XHTML5 & HTMLbook

```
<figure>
  <figcaption>Adorable cat</figcaption>
  
</figure>
```

## Example

Examples are titled and numbered. The example is a container for a group of one or more paragraphs and code segments. Some examples are designated by discipline, user level, and architecture for inclusion/exclusion.

**Example 40. Example: DocBook5**

A self-referential example, using literallayout to identify text that is to be rendered as-is.

```
<example arch="XHTML5"><title>Example</title>
  <para>A self-referential example,
    using literallayout to identify text that is to be rendered as-is. </para>
  <literallayout>
    &lt;example>&lt;title>Example&lt;/title>
    &lt;para>A self-referential example&lt;/para>
    &lt;literallayout>
      [...]
    &lt;/literallayout>
  &lt;/example>
</example>
```

**Example 41. Example: XHTML5**

```
<div id="..."
  title="..."
  class="example Markup"
  data-type="example"
  data-arch="XHTML5">
  <div class="example-title">Example 12.39.
    Example example: DocBook5</div>
  <div class="example Markup-contents">
    <p>A self-referential example, [...]</p>
    <div class="literallayout Markup">
      <p><br />
        &lt;example>&lt;title>Example&lt;/title><br />
        &lt;para>A self-referential example, <br />
          [...] &lt;/para><br />
        &lt;literallayout><br />
          [...]
        &lt;/literallayout><br />
      &lt;/example></p>
    </div>
  </div>
</div>
```

Notice the use of &lt; to represent the left angle bracket < character so that it is not interpreted as an XML starttag marker.

We use the following @arch values to distinguish examples whose content is of a known type:

Architecture types	DB5@arch -- [XHTML5@data-arch]
	CSS
	DocBook5
	JavaScript
	JSON
	XHTML5
	XML

**Example 42. Example of an example**

An example containing a paragraph and literal layout

echo "Hello World!"

```
<example arch="DocBook5"><title>Example</title>
  <para>An example containing a paragraph and a program listing.</para>
  <programlisting>echo "Hello World!"</programlisting>
</example>
```

## Informal example

Informal examples lack a numbered title. The informal example is a container for a group of one or more sentences and code segments. Some informal examples are designated by discipline, user level, and architecture for inclusion/exclusion.

The following is an informal example...

An informal example with opening paragraph and a program listing.

```
echo "Hello World!"
```

### Example 43. Informal: DocBook5

```
<para>The following is an informal example...</para>
<informalexample>
  <para>An informal example with opening paragraph and a program listing.</para>
  <literallayout>echo "Hello World!"</literallayout>
</informalexample>
```

### Example 44. Informal: XHTML5

```
<p>The following is an informal example...</p>
<div class="informalexample" data-type="example">
  <p>An informal example with opening paragraph and literal layout.</p>
  <pre class="programlisting" data-type="programlisting">echo "Hello World!"</pre>
</div>
```

## Hypertext

In most academic works, whether in print or electronic form, there are four classic forms of reference: the cross-reference, the citation, the footnote, and the glossary term. Hypertext capability enhances the useability of an academic work by facilitating structural and relational navigation.

Hypertext presents itself with varying capability in the ePub readers we have tried. Notably, the Lucifox [<http://lucidor.org/lucifox/>] add-on Firefox offers more sophisticated hypertext cues than other readers; hovering the mouse pointer over internal cross-references yields a pop-up text box that offers a glimpse of the link target. For example, when hovering over a footnote marker, the content of the footnote appears, and when hovering over a cross-reference to a chapter, section, figure, or sidebar, a text box shows the beginning of the respective content. We have toyed with the idea of adding scripted solutions in other readers, but we are ever hopeful that ePub readers will evolve toward more sophisticated hypertext capabilities.

Hypertext cross-references should offer cues upon selection, such as a tool tip with the full title of the target, its type, and an elided version of the target content, graphics excluded. This feature is currently under development. Parenthetical cross references are discoverable by inspection; they are most often enclosed within parentheses; they contain a cross-reference; they may be presented with diminished opacity, remedied by hovering over the text.

## Cross-references

Cross-references are a common feature of academic and technical works, where short-form connections to other components of the work must remain synchronized, even while the referents to those components, including their labels and titles, may be subject to editorial change. In print



contexts, the changing referent problem is complicated by potentially changing page numbers during the copy-fitting phase of the book building process. In hypertext contexts, the referents can be presented as hypertext links, without any practical need for page numbers.

Cross-references are effected in the markup with the `<xref>` element, whose `@linkend` value reflects the identifier of any chapter, section, table, figure, example, or sidebar. References to footnotes are effected by the `<footnoteref>` element. References to bibliographic entries are effected by the `<biblioref>` element. Cross-references rely upon identifiers assigned to every chapter, section, figure, example, sidebar, footnote, and bibliographic entry. For chapters and sections, we have chosen to employ identifiers with type prefixes and numbering that reflects the location of the object within the book hierarchy. Figures, tables, and example identifiers reflect the containing chapter number and the sequential position of the object within the chapter. Sidebar identifiers reflect the containing section number and a keyword assigned by one of the editors. Footnote identifiers reflect the type with an “endnote-” prefix followed by a three-digit number that reflects its position in the sequence of footnotes through the book.

Chapter references

#### Example 45. Chapter reference: DocBook5

```
<para>Reference:
    <xref linkend="chapter-1"/></para>
<para>Reference (short):
    <xref linkend="chapter-1" xrefstyle="short"/></para>
```

#### Example 46. Chapter reference: XHTML5

```
<p>Reference:
    <a href="ch1.xhtml" title="Chapter 1. Foundations for Organizing Systems"
      class="xref"
      rel="chapter" rev="xref"
      epub:type="xref"
      data-type="xref"
    >Chapter 1, “<em>Foundations for Organizing Systems</em>”</a></p>
<p>Reference (short):
    <a href="ch1.xhtml" title="Chapter 1. Foundations for Organizing Systems"
      class="xref"
      rel="chapter" rev="xref"
      epub:type="xref"
      data-type="xref"
    >Chapter 1</a></p>
```

Section references

#### Example 47. Section reference: DocBook5

```
<para>Reference to a Section:
    <xref linkend="PRODNOTES.6"/></para>
<para>Reference (short) to a Section:
    <xref linkend="PRODNOTES.6" xrefstyle="short"/></para>
<para>Reference to a Section:
    <xref linkend="PRODNOTES.6.1"/></para>
<para>Reference (short) to a Section:
    <xref linkend="PRODNOTES.6.1" xrefstyle="short"/></para>
```

#### Example 48. Section reference: XHTML5

```
<p>Reference to a Section:
    <a href="ch12s06.xhtml#PRODNOTES.6.1"
      title="6.1. Cross-references"
      class="xref"
      rel="section" rev="xref"
      epub:type="xref"
    ></a>
```

```
data-type="xref"
>§12.6.1, “Cross-references”</a></p>
<p>Reference (short) to a Section:
<a href="ch12s06.xhtml#PRODNOTES.6.1"
  title="6.1. Cross-references"
  class="xref"
  rel="section" rev="xref"
  epub:type="xref"
  data-type="xref"
>§12.6.1</a></p>
```

Figure references

**Example 49. Figure reference: DocBook5 and XHTML5**

```
<para>Reference to a figure:
  <xref linkend="chapter-1-figure-1.1"/></para>

<p>Reference to a figure:
  <a href="ch01s02.xhtml#chapter-1-figure-1.1"
    title="Figure 1.1. An Organizing System."
    class="xref"
    rel="figure" rev="xref"
    epub:type="xref"
    data-type="xref"
  >Figure 1.1, “An Organizing System.”</a></p>
```

Example references

**Example 50. Example reference: DocBook5 and XHTML5**

```
<para>Reference to an example:
  <xref linkend="chapter-8-example-5"/></para>

<p>Reference to an example:
  <a href="ch08s03.xhtml#chapter-8-example-5"
    title="Example 8.5. Basic ways of writing part of a book description."
    class="xref"
    rel="example" rev="xref"
    epub:type="xref"
    data-type="xref"
  >Example 8.5, “Basic ways of writing part of a book description.”</a></p>
```

Table references

**Example 51. Table reference: DocBook5 and XHTML5**

```
<xref linkend="table-ascii"/>

<p>Reference to a table:
  <a href="ch08s03.xhtml#table-ascii"
    title="ASCII"
    class="xref"
    rel="table" rev="xref"
    epub:type="xref"
    data-type="xref"
  >Table 8.1. “ASCII”</a></p>
```

Sidebar references

**Example 52. Sidebar reference: DocBook5 and XHTML5**

```
<xref linkend="sidebar-PN.6-parenthetical-xrefs"/>
```

Footnote references

```
<p>Reference to a sidebar:
  <a href="ch12s06.xhtml#sidebar-PN.6-parenthetical-xrefs"
    title="Parenthetical Cross-references"
    class="xref "
    rel="sidebar" rev="xref"
    epub:type="xref"
    data-type="xref"
  >Parenthetical Cross-references</a>.</p>
```

**Example 53. Footnote reference: DocBook5 and XHTML5**

```
<para>A footnoteref<footnoteref linkend="endnote-PN-1"/></para>

<p>A footnoteref:<span class="footnotesuperscript"
  ><a href="ch12.xhtml#ftn.endnote-PN-1"
    title="Endnote XXX [...]"
    rel="footnote" rev="footnoteref"
    class="footnoteref LIS Law"
    epub:type="noteref"
  >12-1[Production]</a></span></p>
```

(See also the section called “Footnotes”.)

Bibliographic references

**Example 54. Bibliographic reference: DocBook5 and XHTML5**

```
<biblioref linkend="Nelson1981"/>
<biblioref linkend="Shakespeare1623"/>

<p>A biblioref:
  [<a href="bi01.xhtml#Nelson1981"
    title="[Nelson1981]"
    class="biblioref"
    rel="biblioentry" rev="biblioref"
    epub:type="xref"
    data-type="xref">Nelson1981</a>] or
  [<a href="bi01.xhtml#Shakespeare1623"
    title="[Shakespeare1623]"
    class="biblioref"
    rel="biblioentry" rev="biblioref"
    epub:type="xref"
    data-type="xref"
  >Shakespeare1623</a>]</p>
```

Glossary references

**Example 55. Glossary reference: DocBook5 and XHTML5**

```
<para>Glossary term references:
  <firstterm linkend="gloss_glossary-term">First mention</firstterm> or
  <glossterm linkend="gloss_glossary-term">glossary term</glossterm>
</para>

<p>Glossary term references:
  <a class="firstterm" epub:type="glossterm"
    rel="glossentry" rev="firstterm "
    href="go01.xhtml#gloss_glossary-term"
  ><em class="firstterm">First mention</em></a> or
```

```
<a class="glossterm" epub:type="glossterm"
    rel="glossentry" rev="glossterm "
    href="go01.xhtml#gloss_glossary-term"
><em class="glossterm" epub:type="glossterm">glossary term</em></a>
</p>
```

## Citations

### Example 56. Citation: DocBook5

```
<para>Citititle: <citititle linkend="Shakespeare1623">Macbeth</citititle></para>
<para>Citation: <citation linkend="Shakespeare1623">(Shakespeare 1623)</citation></para>
```

### Example 57. Citation: XHTML5

```
<p>Citititle: <em class="citititle"
  ><a href="bi01.xhtml#Shakespeare1623"
    title="The Tragedie of Macbeth"
    class="citititle"
    rel="biblioentry" rev="citititle"
    epub:type="xref"
    data-type="xref">Macbeth</a></em></p>
<p>Citation: <span class="citation"
  ><a href="bi01.xhtml#Shakespeare1623"
    title="The Tragedie of Macbeth"
    class="citation"
    rel="biblioentry" rev="citation"
    epub:type="xref"
    data-type="xref"
  >(Shakespeare 1623)</a></span></p>
```

Here is how we encoded an earlier paragraph:

```
<para>In order express some of the characters used in this book,
we require a Unicode font as in our reference to the classic Japanese film
<citititle linkend="Kurosawa1957"
  ><foreignphrase xml:lang="jp">Kumonosu-jo</foreignphrase
  > (<symbol>蜘蛛巣城</symbol>) (Throne of Blood)</citititle>
<citation linkend="Kurosawa1957">(Kurosawa 1957)</citation>.
</para>
```

What we want to consider in this encoding are the details toward the end, where part of the cited title is identified as a foreign phrase of XML language type “jp” and both are linked to the bibliography entry whose XML identifier is “Kurosawa1957”. In most print or most ebook contexts, we expect the cited title to present itself in a slanted typeface. In an ebook context, we might further expect the citation to become a hypertext link to the appropriate entry in the bibliography, or a pop-up information box containing a hypertext cue, or even the entry itself. In an interactive ebook context, we might expect scripts to search for local availability of the cited resource, within an academic or institutional library, for example. In a learning tools context, we might expect scripts to discover that Shakespeare’s *Macbeth* is cited in close proximity to [Kurosawa] in the text.

### Abbreviations, Acronyms, and Cite Ref Entry

Abbreviations and acronyms like USA and NASA. Abbreviation is an early form of editorial hypertext, where we rely upon the memory to mentally equate the abbreviation with the fully expanded form. Editorially, when we first mention an undefined abbreviation or acronym, it should parenthetically follow the fully expanded form, or defining term. It is up to the author/editor to decide whether an abbreviation is commonly understood. Subsequent or other mentions of an abbreviation or acronym can be linked back to first use, defining term, or master glossary entry as appropriate.

Cite reference entry: National Association of Professional Organizers(NAPO) is presented in normal face.

We use the `<citerefentry>` element to collect the fully expanded defining term and its associated abbreviation. The `<citerefentry>` element is a vestige of the UNIX operating system and the so-called “man page references” that abounded in technical documentation. The UNIX commands, file types, libraries, network protocols, etc., were all categorized according to a widely-adopted volume numbering system, which was later supplanted by a combination of numeric and alpha type designators. Because we are not using UNIX man pages in our book project, I decided to re-purpose this handy markup to facilitate my editorial goals.

We re-purposed `<citerefentry>` as an editorial tool to marshal first mentions and defining terms for any formal name with an associated abbreviation or acronym. A `<citerefentry>` contains a `referentrytitle` and a `manvolnum`, which we associate with the defining term and the abbreviation, respectively. The identifier for a first mention `citerefentry` has an identifier prefix “first\_” and the `referentrytitle` has identifier prefix “ref\_” These identifiers enable cross-references to and from the bibliography and glossary, as I will explain later.

## Footnotes

Footnotes in this work may be presented in one of the following modalities: relocated as end notes; placed at the end of every chapter in print; and, at the end of every section in an ePub.

- In print, the footnotes in each chapter are gathered to the end of the chapter and presented in the order of appearance, distinguished typographically by reduced type size or otherwise.
- In the PDF, each footnote will be presented following the paragraph in which it occurs, distinguished typographically by indent and reduced type size or otherwise.
- In ebook editions designed for use on general purpose ePub3/XHTML/CSS readers, including Adobe Digital Editions and Bluefire, each footnote will be presented following the paragraph in which it occurs, distinguished typographically by indent and reduced type size or otherwise. ADE does not provide JavaScript and its CSS interactions are limited.
- In ebook editions designed for use on ePub3/XHTML/CSS+JavaScript readers, including Azardi, Calibre, and Radium, each footnote will be hidden until a mouse action upon the footnote reference, after which it will be presented following the paragraph in which it occurs, distinguished typographically by indent and reduced type size or otherwise. Calibre and the Radium extension for Chrome behave quite well in the face of JavaScript and CSS challenges. **Wanted Feature:** JQuery script that fetches footnote and presents in an alert box; not sure about how/whether links would work.
- In ebook editions designed for use specifically on iBooks with its pop-up viewports for references to footnotes, the content of the footnotes is presented in a small window that appears above the main context view.
- In the ebook edition designed for use specifically on Lucifox with its pop-up tooltip hypertext cues, the footnotes in each section are gathered to the end of the section and presented in the order of

appearance, distinguished typographically by reduced type size or otherwise. Lucifox's hypertext cues provide a preview of the footnote when hovering over the footnote reference. Lucifox exhibits strange behavior around JavaScript and CSS interactions that affect the DOM, or content visibility.

- In ebook editions intended for use on Alkido, Kindle, Kobo, Lekst Readers, Publiwide, STDU, and Sumatra, the behavior of footnotes is currently undetermined.

### Footnote pop-ups in iBooks on Mac

In iBooks on a Mac, When one clicks on a footnote marker, a pop-up box presents the content of the footnote. So far, so good; this is a desirable feature for a book of this kind.

However, we ran into a bit of a snag with links within footnote. When one clicks on an in-book link, like a citation, rather than retrieve the linked entry for presentation in that same view port, the view port goes blank. When one clicks instead on an external link, the resource is retrieved and presented in the view port, which is awkward because the view port cannot be re-sized to accommodate a full-sized web site.

Now, we have to decide: Which is the more palatable dysfunction? Without the pop-up, selection of the note reference is a potentially costly traversal, and the reader can gain no foreknowledge of the link target. With the pop-up, readers initially achieve instant gratification, but then suffer cognitive dissonance due to unexpected link behavior within its view port.

An alternative would be to implement our own hypertext cue tooltips as mentioned earlier.

This is a paragraph with a footnote.<sup>3</sup>

### Example 58. Footnote: DocBook5

```
<para>This is a paragraph with
    a footnote.<footnote xml:id="endnote-PN-3">
    <para>This is my footnote. </para>
</footnote>
</para>
```

### Example 59. Footnote marker in content: XHTML5

In the XHTML5 representation of the paragraph content, a footnote marker with a hypertext link to the actual footnote is included in the position at which it was found in the DocBook5 source document. The content of the actual footnote is saved for presentation, which may be at the end of the chapter, section, or even the end of the paragraph in which it is contained. We have provided for each of these footnote presentations in different build scenarios.

```
<p class="Markup">This is a paragraph with
a footnote.<span class="footnotesuperscript"
    ><a id="ftn.endnote-PN-2"
        href="#endnote-PN-2"
        class="footnote Markup"
        rel="footnote" "rev="footnoteref"
        epub:type="noteref"
    >12-2[Mark]</a></span>
</p>
[...]
```

Every footnote has an identifier, a label, and one or more audience values.

---

<sup>3</sup>This is my footnote.

Notice that the footnote markup abuts the sentence-ending punctuation. One must be vigilant about placement of footnote markup to avoid unfortunate line breaks that place the footnote marker on a new line, apart from the sentence to which it is supposed to be attached.

### Example 60. Section Footnotes: XHTML5

The XHTML5 result is a bit more complex. The footnote marker is left at the end of each of the affected sentences, and the footnote is repositioned to the end of the section.

```
<div class="footnotes" epub:type="footnotes">
  <br /><hr class="footnote-hr" />
  <div id="endnote-PN-2"
    class="footnote Markup"
    epub:type="footnote"
    data-audience="Markup">
    <p class="Markup"><a href="#ftn.endnote-PN-2"
      rel="footnoteref" "rev="footnote backlink"
      class="backlink para Markup"
    ><span class="footnotesuperscript">[12-2]
    </span></a>This is my footnote. </p>
  </div>
```

### Example 61. Paragraph Footnotes: XHTML5

The XHTML5 result leaves a footnote marker is left at the end of each of the affected sentences, and the footnote is repositioned to the end of the paragraph. If there is more than one footnote within a given paragraph, they are collected to the end of the paragraph and presented in the original order of occurrence.

Choosing selectors to manipulate footnote presentation is critical. Other scripts will be toggling the visibility of paragraphs, while direct manipulation of a given footnote will require some precision. When a user selects a `footnoteref`, we want to get its `#href` value and use it in the selector to show the contents of the footnote. The relevant selectors are `[#href]>p`, `div.inline.footnotes`, `div[epub:type='footnotes']<div[epub:type='footnote']`, `[id][data-audience]<p[data-audience]`, and `a[epub:type='noteref']`, and `a.footnote`, and `[href]`.

```
<div class="inline footnotes" epub:type="footnotes">
  <div id="endnote-PN-2"
    class="footnote Markup"
    epub:type="footnote"
    data-audience="Markup">
    <p class="Markup" data-audience="Markup"
      ><a href="#ftn.endnote-PN-2" class="backlink para Markup"
        rel="footnoteref" "rev="footnote backlink"
      ><span class="footnotesuperscript">[12-2]
    </span></a>This is my footnote. </p>
  </div>
```

### Example 62. Paranotes: JQuery

The paragraph endnotes script first tests to determine that it is allowed to run in the current reader, announces its findings to the console, then it tests whether there are any paragraph endnotes in this document before proceeding, again announcing that it is running. (Assuming that `core.js` has enabled/disabled features.) Next, we hide all paragraph endnotes and resize the window to ensure that the document flow is accurate. We then create two click functions. First is the hypertext link that represents the footnote call marker, typically found at the end of a sentence or paragraph, so that when the user activates the call marker, the associated footnote exposes its contents. Second is the footnote label

marker, typically found at the beginning of the first paragraph of a footnote, so that when the user activates it, the contents of the footnote will fade and then slide up. We hide the contents of the footnote rather than the footnote itself for a technical reason: hiding hypertext link targets can cause some browsers and readers to become confused and get jumpy.

```
$(document).ready(function() {
  var paranotes = (window.features['paranotes']);
  var inlines = $("div.inline.footnotes").length;
  console.log ("ParaNotes feature: " + paranotes + " on " + window.reader);
  if (($("div.inline.footnotes").length) &&
      (window.features['paranotes'] == "enabled")) {

    console.log ("ParaNotes feature: Preparing document and scripts");
    $("div.inline.footnotes>div.footnote").children().hide();
    $(window).resize();

    $("a.footnote[href]").click(function(){
      var xxx = $(this).attr('href');
      // only one visible endnote at a time.
      if ($(xxx).children(":hidden").length) {
        // Hide any already visible footnote contents
        $("div.inline.footnotes>div.footnote").children().slideUp(900);
        // open slowly with dim text, gradually darken, then quickly
        $(xxx).children().animate({opacity: '.25'}, 'fast');
        $(xxx).children().animate({height: 'toggle'}, 'slow');
        $(xxx).children().animate({opacity: '1'}, 'fast');
      } else {
        $(xxx).children().animate({opacity: '.66'}, 'slow');
        $("div.inline.footnotes>div.footnote").children().slideUp(900);
      }
    });

    $("a[href^='#ftn.endnote']").click(function() {
      // User clicks on superscripted backlink before content of footnote.
      // Hide content of this note and all others.
      $("div.inline.footnotes>div.footnote").children().slideUp(900);
    });
  }
});
```



### Footnote Metadata

The HEAD of each XHTML5 result document contains a set of LINK and META elements that relate to footnotes contained within that document. This markup will be used by scripts to present a user interface to selectively manage the visibility of footnotes by type, and individually.

### Example 63. Link elements in head: XHTML5

The following example comes from `ch01.xhtml`. There are three endnotes in that file; all three relate to LIS, and the first also relates to Law.

```
<link href="#endnote-001" role="LIS Law"
      title="Endnote 001 [LIS Law]"
      rel="noteref"
      epub:type="noteref" />

<link href="#endnote-002" role="LIS"
      title="Endnote 002 [LIS]"
      rel="noteref"
      epub:type="noteref" />

<link href="#endnote-003" role="LIS"
      title="Endnote 003 [LIS]"
      rel="noteref"
      epub:type="noteref" />

<meta name="notecount.LIS" content="3" />
<meta name="notecount.Law" content="1" />
<meta name="notecount" content="3" />
```

## Glossary terms and references

A glossary term is a word or phrase that is formally defined in the glossary of the work. A glossary term can be any word or phrase whose meaning, needs to be clearly defined in the context of the work. Distinctly defining the terms of discourse in use in a work is a commonly recognized editorial and typographic practice. As authors, editors, and compositors, we strive to highlight and identify glossary terms that appear in text as a service to our readers.

We distinguish among three general classes of glossary terms: *first mention*, **defining term**, and other mentions. A first mention of a glossary terms should be distinguished text, such as an italic typeface. We suggest italic because that is the traditional typographic device to use. Within a definition, the defining term should be distinguished, typically with a bold typeface, although some disciplines prefer the use of underscore. Highlighting and linking other mentions of a glossary term is optional; other mentions within a *work* are usually presented in an italic typeface.

### Example 64. Glossary Terms: DocBook5

```
<para><phrase xml:id="def_glossary-term"
           role="definition" role="definition"
>We distinguish among three general classes of glossary terms:
  <firstterm xml:id="first_glossary_term"
            linkend="gloss_glossary-term">first mention</firstterm>,
  <glossterm xml:id="term_glossary-term"
```

```
>defining term</glossterm>,
and
<glossterm linkend="gloss_glossary-term"
>other mentions</glossterm>.</phrase>
```

### Example 65. Glossary Terms: XHTML5

```
<p><span id="def_glossary-term" class="definition" role="definition"
>We distinguish among three general classes of glossary terms:

<a id="first_glossary_term"
href="go01.xhtml#gloss_glossary-term"
class="firstterm"
rel="glossentry" rev="firstterm"
epub:type="glossterm glossref"
><em class="firstterm">first mention</em></a>,

<em id="term_glossary-term"
class="glossterm"
rel="glossentry" rev="glossterm"
epub:type="glossterm"
>defining term</em>,

and

<a href="go01.xhtml#gloss_glossary-term"
class="glossterm"
rel="glossentry" rev="glossref"
epub:type="glossref"
><em class="glossterm" epub:type="glossterm"
>other mentions</em></a>.
</span>

[...]
```

### Example 66. Glossary Terms: DocBook5

The first mention and defining glossary terms must have distinct identifiers. The first mention may link to the defining mention or to a glossary entry. Other mentions may link to a first mention, a defining mention, or the glossary, depending on the editorial and rhetorical intent of the authors and editors.

A `<firstterm xml:id="first_widget" linkend="gloss_widget">widget</firstterm>` can be used to...

A `<glossterm xml:id="term_widget">widget</glossterm>` is a...

A `<glossterm linkend="gloss_widget">widget</glossterm>`, as we recall, is a...

Glossary term mentions relate to their definitions through links, which may occur in running text or in localized glossary lists. Such links may relate to the book's main glossary or to localized glossary lists. In this work, I linked glossary terms anywhere I thought that the gentle reader might be served by a reminder that the definition of an important term is handily available.

In the book's main glossary, and in various chapters, there are formal glossary lists, composed of a glossary term and a glossary definition. The presentation of these glossary terms is as list markers beside their corresponding glossary definitions.

In print, the multi-column presentation of the main glossary had a quirky algorithm behind it that was calculating column allocations incorrectly. As a consequence, the right column of the glossary proper was indented to the left, with respect to the page margin, for no apparent reason.

Present in *italic* face, except when it is a defining term and then present in **bold** face.

## Links

DocBook5 offers us internal and external links through its `<link>` element, and email links through its `<email>` element. Links are presented as distinguished text. Hovering over a link should provide cues. Link selection invokes traversal.

### Example 67. Links: DocBook5

See <http://en.wikipedia.org/wiki/Hamlet>.

An external link is created by setting the `@xlink:href` value to any valid URI. When using DocBook's external links, the `xlink` namespace must be declared.

```
<para>See <link xmlns:xlink="http://www.w3.org/1999/xlink"
      xlink:href="http://en.wikipedia.org/wiki/Hamlet"
      ><uri>http://en.wikipedia.org/wiki/Hamlet</uri></link>.
</para>
```

### Example 68. Links: XHTML5

```
<p>See <span class="link"
      ><a class="link" href="http://en.wikipedia.org/wiki/Hamlet" target="_top"
      ><code class="uri" >http://en.wikipedia.org/wiki/Hamlet</code></a></span>.
</p>
```

An internal link is created by setting the `@linkend` value to any identifier in the book. We have worked hard to avoid using the `link` element for internal linking, giving preference to DocBook's cross-referencing and citation facilities.

## Local Transclusion

We use DocBook's transclusion capabilities to maintain as much original text as possible in the chapters, and we use `xref` in a special way to effectively copy phrases of text into the quizzes and the glossary. We can do the same with the defining mention of an abbreviation or acronym, or the content of a paragraph, so long as those containers have identifiers. Anything that has been identified can be transcluded, but only into a similar location in the content hierarchy. The implications are a bit tricky to explain, so I will try to give enough examples of how we used DocBook5 transclusion, mostly in the glossary and quizzes, but in a few other circumstances as well.

### Example 69. Transclusion: DocBook5

Transclusion of local content within a text is a feature of DocBook5 that must be reified as an XHTML5 document instance by transformation. We express the target of a transclusion by using the `xref` element with both `@linkend` and `@endterm` set to the same phrase-level element, optionally setting `@role="transclusion"`. The first form, without `@role`, copies the content of the referenced element, and layers a hypertext link from the target to the source. The second form, with `@role="transclusion"`, copies the content, but does not make it a link, thereby allowing links within to continue to come through.

```
<glosslist>
...
<glossentry xml:id="gloss_lexical_tagging">
  <glossterm><xref linkend="term_lexical_tagging" endterm="term_lexical_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_lexical_tagging" endterm="def_lexical_tagging" role="transclusion"/></para>
  </glossdef>
</glossentry>
```

```
...
<glossentry xml:id="gloss_semantic_tagging">
  <glossterm><xref linkend="term_semantic_tagging" endterm="term_semantic_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_semantic_tagging" endterm="def_semantic_tagging" role="transclusion"/></para>
  </glossdef>
</glossentry>
...
<glossentry xml:id="gloss_typographic_tagging">
  <glossterm><xref linkend="term_typographic_tagging" endterm="term_typographic_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_typographic_tagging" endterm="def_typographic_tagging" role="transclusion"/></para>
  </glossdef>
</glossentry>
...
</glosslist>
```

## Semantic Text

We will now consider a semantic perspective on the sentences, phrases, and words, within the prose. When we are reading the text of the book, we encounter abbreviations, acronyms, and terms that are new to us. We see people's names and job titles, organizations, citations and bibliographic entries, authors, editors, publishers, locations and dates, definitions and statements, quoted text, symbols, file names, and hypertext links abound. (Oh, my!) Sometimes the intentions expressed in the underlying editorial patterns reveal themselves to the gentle reader through text that is highlighted in some way (e.g., “emphasized” with *slanted* or **heavy** typefaces, or **both**). At other times, there are no typographic cues to let the reader know that a given word, phrase or sentence is distinguished in any way, yet their interpretation can be often inferred.

The key point is that we have identified some semantic components within the text of this work, and we have attempted to employ them in such a way that the result will enhance comprehension and offer utility. We had hoped to develop the ability to refine searches by semantic types, and to control a few formatting parameters on semantic types, to underscore definitions or dim parenthetical cross-references, for example.

## Why?

Our central purpose in organizing is “bringing like things together and differentiating among them.”

In order to express semantic distinctions in the text, we must start with suitable markup in our sources, distinguish target media types with stylistic nuance in our style sheets, and imbue those elements with functionality through the inherent capabilities of the platform where practical, and through our own code as needs arise.

We have, for example, attempted to provide a subtle typographic distinction for definitions, principles, and statements in this work. Some stylesheets also offer text enlarging and other effects when hovering over a word, phrase, or sentence.

To embellish the content of the book, we can offer reader control over presentational characteristics, such as pull-quotes from key passages in the text. Alternatively, we can offer the reader fore-shortened content, by filtering out related but ancillary content.

Luckily, DocBook5 offers many utilitarian elements in its native vocabulary. Most of the time, we employed DocBook5 elements and attributes in the ways that they were originally intended. At other times, we took license, not by modifying DocBook5 syntax in any way, but by overlaying our own meaning on the use of various attributes, and their potential value sets. For example, we use `@audience` and `@userlevel` with a tightly controlled vocabulary to produce customized editions. We use `@role` much more carelessly. Please note that we have not yet taken full advantage of all of the markup that

we have employed in the book, but we leave that as an exercise for the reader and for future editions of the work.

As we explain elsewhere, we have changed the meaning of DocBook's `<citerefentry>` to accommodate a new notion of a reference entry, as a citation of a formal name and its corresponding short form, which is most often an abbreviation or an acronym. For example, Extensible Markup Language(XML). I treated them like the first mention of a glossary term.

## Lexical and Typographic Elements

Eventually, after you get past all the semantic tagging, you get down to the lexical and typographic stuff, and we might as well throw in those pesky trade marks and registered symbols too. Let's face it, sometimes you simply need to get down to the metal and insist on a specific type of emphasis, for editorial purposes, of course.

### DocBook5 elements used for editorial, lexical, and typographic annotation

emphasis

Emphasis: `@role= bold`, *italic*, and *bold italic* all work on all targets. We observe that **strong** emphasis may present in italic on some readers and bold on others, as a result, we prefer not to use it. However, underline and ~~striketrough~~ are unreliable among the readers we have surveyed.

#### Example 70. Emphasis: DocBook5 and XHTML5

```
<emphasis role="bold">bold</emphasis>
<emphasis>italic</emphasis>
<emphasis role="bold italic">bold italic</emphasis>
<emphasis role="strong">strong</emphasis>
<emphasis role="underline">underline</emphasis>
<emphasis role="striketrough">striketrough</emphasis>

<strong class="bold">bold</strong>
<em class="emphasis">italic</em>
<em class="bold italic">bold italic</em>
<strong class="strong">strong</strong>
<em class="underline">underline</em>
<em class="striketrough">striketrough</em>
```

We have used emphasis extensively. We make no apologies. Attempts to force authors and editors to use semantic elements to achieve typographic distinctions are futile. Sometimes the content creator wants a text passage to be distinguished typographically for a reason that was not conceived by the markup designer. Artistic license grants content creators that right. The `b`, `i`, `u`, and `s` elements were more convenient than using `<em class="...">` but we do what we have to do.

foreignphrase

Foreign phrase: *C'est la vie*. As a matter of editorial style, foreign phrases are presented in *italic* face. In this work, the markup editor made a point of tracking down every foreign word or phrase. There has been some editorial contention over the question of whether *ad hoc* is still considered a foreign phrase. `@xml:lang="fr"`

#### Example 71. Foreign phrase: DocBook5 and XHTML5

```
<foreignphrase xml:lang="fr">C'est la vie.</foreignphrase>
```

literal

```
<span class="foreignphrase"><em class="foreignphrase">C'est la vie.</em></span>
```

Literal text: `literal text`. Present in monospace face; kerning is optional.

**Example 72. Literal: DocBook5 and XHTML5**

```
<literal>literal text</literal>
```

```
<code class="literal">literal text</code>
```

phrase

A phrase: A text phrase. (See also the section called “Editorial Semantics”.)

**Example 73. Phrase: DocBook5 and XHTML5**

```
<phrase>A text phrase</phrase>
```

```
<span class="phrase">A text phrase</span>
```

quote

A quoted “word or expression.” Supply surrounding quotes, and design for contrasting interior quotes.<sup>4</sup>

**Example 74. Quote: DocBook5 and XHTML5**

A quoted `<quote>word or expression.</quote>`

A quoted `<span class="quote">“word or expression.”</span>`

superscript

Superscript  $x^2$ . Superscript, size .6 em or 60%

**Example 75. Superscript: DocBook5 and XHTML5**

```
x<superscript>2</superscript>
```

```
x<sup>2</sup>
```

subscript

Subscript  $B_{12}$ . Subscript, size .6 em or 60%

**Example 76. Subscript: DocBook5 and XHTML5**

```
B<subscript>12</subscript>
```

```
B<sub>12</sub>
```

symbol

A symbol: 城 ⇔ We don't use a lot of symbols—but we have tried to identify them as such.

**Example 77. Symbol: DocBook5 and XHTML5**

```
<symbol xreflabel="城">城</symbol>
```

```
<symbol>&#8660;</symbol>
```

```
<span class="symbol">城</span>
```

---

<sup>4</sup>One must be diligent about U.S. style punctuation rules, the preference of this work's editor and the publishers. The tendency of this markup editor is to leave all punctuation outside of the quote marks unless the punctuation is being quoted, as in a question mark. It is a question of typographic style versus logical punctuation. However, it does affect our ability to rely on the suitability of text for transclusion. If we quote a “term,” and then want to refer to that term again by transclusion, we will absorb its punctuation. Luckily, we didn't need to do that because we edited around that problem. In any case, you may notice that I have adopted very loose punctuation rules in this chapter.

	<code>&lt;span class="symbol"&gt;↔&lt;/span&gt;</code>
wordasword	A word, as a <i>word</i> . <sup>5</sup> Present in a contrasting face, such as italic, sans-serif, or with letter spacing.
<b>Example 78. Wordasword: DocBook5 and XHTML5</b>	
	A word, as a <code>&lt;wordasword&gt;word&lt;/wordasword&gt;</code>
	<code>&lt;em class="wordasword"&gt;word&lt;/em&gt;</code>

## Editorial Semantics

In this work, the DocBook5 phrase element is often used to contain and identify key text passages, sometimes as brief as an abbreviation, a cross-reference, or part of a sentence; sometimes a phrase may contain as much as several sentences, when they form definitions and statements. Since we lack grammatical semantics in our markup languages of choice, we are coercing the `<phrase>` element into serving as a generic sub-paragraph container, which we sub-class for various purposes.

Phrases with `@role` values are used to identify editorial and parenthetical content; definitions, statements, and principles; and, other discourse types; among others.<sup>6</sup>

### Elements of Exposition

definition	A <i>definition</i> consists of a text phrase, as short as a few words or as long as several sentences, that serves as the semantic expression of a term. (Include <code>@role="definition"</code> in XHTML5 for ARIA.)
principle	A <i>principle</i> consists of a text phrase, as short as a few words or as long as several sentences, that serves as the semantic expression of a principle. Notice that the previous sentence is categorized as a definition, while the following sentence is categorized as a principle. Following a definition with an example improves cognition.
statement	A <i>statement</i> consists of a text phrase, as short as a single sentence or as long as several sentences, that serve as key topic sentences in the work. By extracting the statements in this work, one would discover an interesting, if disjoint, distillation of its content.

### Editorial Phrases and Paragraphs

editorial	Text content that discusses or annotates the book and could be safely ignored without losing any of the meaning of the work.
-----------	--

---

<sup>5</sup>There are many instances in this work where a word is quoted to distinguish it as a novel word. Editorially, we are remiss and should have used *wordasword* more, rather than relying on quotes to capture the semantics of “word as word.” We have used quoted text instead of *wordasword* simply because the authors had already quoted their text in Microsoft Word and it would have consumed too much effort to attempt to identify all occurrences of quoted text that should be re-tagged, re-tagging them, and then changing the CSS formatting of *wordasword* so that it was presented in quotes. The juice was just not worth the squeeze, so we are stuck with “word as word” from MS Word..

<sup>6</sup>There are almost three hundred phrases identified, in the DocBook5 markup, by `@role="definition"`, over two hundred identified by `@role="principle"`, and over seven hundred phrases identified by `@role="statement"`. Most definitions and many statements have identifiers to enable content transclusion into the Glossary and the Key Points sections.

The DocBook5 phrase element and its role values is transformed into an XHTML5 span element with equivalent `@class` values. Thus, the CSS selector for these semantic phrases is derived simply; the selector for a definition is: `span.definition`

We have begun to explore the use of comparative, contrast, cause, effect, problem, solution, and tradeoff as other subtypes of expository text.

There are over two thousand `<phrase>` elements in this work; over three hundred of those have other `@role` values for quiz-, index-, and glossary-making purposes.

Much of the front matter, the acknowledgments, and various editorial notes at the beginning of the bibliography, glossary, and index, for example. An experienced reader might choose to hide them to hasten the flow of the text. (65)

parenthetical

There are hundreds of parenthetical cross-references throughout this work. (~845) One reader might choose to highlight parenthetical cross-references to encourage discovery, while another reader might choose to hide them to hasten the flow of the text.<sup>7</sup>

### Note

We have chosen not to use markup to identify registered product names (e.g., Microsoft Word®), service names (Google Maps(SM)), trademarks (CocaCola™), and copyrights (Bugs Bunny©). We cannot depend on downstream conversion tools and readers to display the marks in a predictable and expected way. Given that we could not rely on proper presentation, we decided to drop them on the basis that this is an academic work rather than a commercial work. Naturally, the editors and authors acknowledge the ownership of any products, services, and marks mentioned in this work. We trust that we have not exceeded the bounds of fair or nominative use.

## Bibliography and Citations

Building a bibliography is a demanding task. A scholarly bibliography must be complete and accurate. Sure, ya could throw together a few citations and call it a day. But that's not the cowboy way.<sup>8</sup>

Bob maintained his bibliography in a database, from which extracted an XML representation and then converted that to our DocBook5 representation. The contents of the bibliography originated in a Mendeley program from which it was exported to End Note XML format. From there it was processed by an XSLT script to transform it into a DocBook5 representation, while injecting identifiers for each entry and each citation. When that process was complete, the manual editing job began.

Each entry in the Bibliography was assigned an identifier so that multiple citations, possibly in different forms or languages, can target the same entry. Each entry in the bibliography also has a link back to the first citation. Every citation has a link to its corresponding entry, and an identifier. We have thought about offering links back to every citation, but we have not yet designed an interface for that selection menu. Since the conversion was completed (December 2012) the contents of the Bibliography have been hand edited to accommodate hypertext linking to and from citations, and new entries added to reflect casual mentions of works in the surface text of the chapters.

There are two general forms of citation in this work: a cited title, like *Moby Dick*, and a formal citation of author and year, like [(Melville 1851)].

### Example 79. Citation: DocBook5

This is how they are encoded:

---

<sup>7</sup>Navigational cues are a useful feature of any work and can be especially helpful to a reader who is new to the material. (Parenthetical content can be distracting when trying to focus upon the underlying text.) Material that is dense with navigational cues can seem more difficult to read, as the landscape of the page is cluttered with hypertext links, and the flow of content is interrupted by parenthetical references.

<sup>8</sup>“The Cowboy Way” was a catch phrase used by Eliot Kimber in a series of postings to comp.text.sgml in the 1980s, wherein he would offer elegant, intellectually satisfying, and, somehow, morally superior solutions to information organization problems using SGML.

“The cowboy way as I used it came from the musical group Riders in the Sky [[http://en.wikipedia.org/wiki/Riders\\_in\\_the\\_Sky\\_\(band\)](http://en.wikipedia.org/wiki/Riders_in_the_Sky_(band))], who had a running bit where the character, a good-guy cowboy in the Lone Ranger mold, would say, in the face of some unpleasant action to be taken, ‘It might be the easy way, but it wouldn’t be the cowboy way!’ the implication being that the cowboy way was the right way. I started using that in the context of using SGML, e.g., ‘SGML, it’s not the easy way, but it’s the Cowboy Way.’ I stopped using the phrase when it was pointed out to me by European and British colleagues that “cowboy” had a much more negative connotation outside the U.S., generally meaning to do something in a reckless or unconsidered fashion (e.g., George Bush was a ‘cowboy’ by rushing into war in Iraq).”

—Eliot Kimber (2014-10-28)



<para>There are two general forms of citation in this work: a cited title,  
like <citetitle linkend="Melville1851">Moby Dick</citetitle>,  
and a formal citation of author and year,  
like <citation linkend="Melville1851">(Melville 1851)</citation>.  
This is how they are encoded:  
</para>

```
<biblioentry xml:id="Melville1851">
  <authorgroup>
    <author><personname><firstname>Herman</firstname><surname>Melville</surname></personname></author>
  </authorgroup>
  <citetitle>Moby Dick</citetitle>
  <subtitle>or The Whale</subtitle>
  <pubdate>1851</pubdate>
  <address><city>London</city></address>
  <publisher>
    <publishername>Richard Bentley</publishername>
  </publisher>
</biblioentry>
```

### Example 80. Citation and Biblioentry: XHTML5

<p>There are two general forms of citation in this work: a cited title,  
like <em class="citetitle"  
 ><a href="bi01.xhtml#Melville1851"  
 title="Moby Dick"  
 class="citetitle"  
 rel="biblioentry" "rev="citetitle"  
 epub:type="biblioref"  
 >Moby Dick</a></em>,  
and a formal citation of author and year,  
like <span class="citation"  
 ><a href="bi01.xhtml#Melville1851"  
 title="Moby Dick"  
 class="citation"  
 rel="biblioentry" "rev="citetitle"  
 epub:type="biblioref"  
 >(Melville 1851)</a></span>. </p>

```
<div id="Melville1851"
  title="Moby Dick"
  class="biblioentry"
  epub:type="biblioentry">
  <p><a class="backlink" "rel="citation" rev="biblioentry backlink"
    href="ch01.xhtml#cite_Melville1851">[Melville1851] </a>
    <span class="authorgroup">
      <span class="firstname">Herman</span>
      <span class="surname">Melville</span>.
    </span>
    <span class="title"><em>Moby Dick</em>. </span>
    <span class="subtitle">or The Whale. </span>
    <span class="pubdate">1851. </span>
    <span class="address">London. </span>
    <span class="publisher">
      <span class="publishername">Richard Bentley.</span>
    </span>
  </p>
</div>
```

The principal citation for any bibliography entry, usually the first citation, has to be assigned an identifier that uses the entry identifier with a prefix, such as “cite\_Melville1851” in the following example.

```
<citation xml:id="cite_Melville1851" linkend="Melville1851">(Melville 1851)</citation>
```

In choosing identifiers for bibliography entries, we opted to use the first author's surname and the year of publication. We could have chosen to elaborate the identifiers by using a type prefix, as we did with the principal citations, but we decided not to use a prefix of any kind for these identifiers. We did run into one problem though: several authors' names included characters outside of the accepted range of identifier characters. We had to compensate by manually editing those few identifiers.

### Example 81. Disciplinary Bibliography Entry: DocBook5

When a bibliography entry is only mentioned in one disciplinary endnote, and is not mentioned in the surface text, we have chosen to assign that entry to its specific discipline or disciplines so that it only appears in editions in which those endnotes will appear. Here, we have an entry that will only appear in editions in which computing, information architecture or the web are selected:

```
<biblioentry xml:id="Morville2006" audience="Computing IA Web">
  <authorgroup>
    <author><personname><firstname>Peter</firstname><surname>Morville</surname></personname></author>
    <author><personname><firstname>Louis</firstname><surname>Rosenfeld</surname></personname></author>
  </authorgroup>
  <pubdate>2006</pubdate>
  <title>Information Architecture for the World Wide Web</citetitle>
  <address>Sebastopol, CA</address>
  <publisher>
    <publishername>O'Reilly</publishername>
  </publisher>
</biblioentry>
```

## Glossary

Building a glossary is a tricky task. It involves first selecting the terms that should be included; then, adding markup for a glossary entry; then, adding appropriate markup to mentions of the term, distinguishing first mentions from defining mentions; then, marking up a term's definition or definitions; then, assigning identifiers to all of the parts; and, finally, adding a transclusion link from the glossary to the place in the text where the term is defined, or vice-versa.

### Example 82. Glossary terms in text: DocBook5

A glossary term should be mentioned within the text of the work and be listed in the master glossary. A glossary entry that is not mentioned in the text is editorially incomplete. A term may be defined within the text, within a glossary list in a section, or only in the master glossary. The first mention, defining mention, and other mentions of a term should be distinguished. (See also the section called “Glossary terms and references”.)

A <firstterm xml:id="first\_widget" linkend="gloss\_widget">widget</firstterm> can be used to...

A <glossterm xml:id="term\_widget">widget</glossterm> is a...

A <glossterm linkend="gloss\_widget">widget</glossterm>, as we recall, is a... that can be used to...

```
<abbrev linkend="gloss_ABC">ABC</abbrev>
```

```
<abbrev linkend="first_ABC"><xref linkend="term_ABC" endterm="term_ABC"/></abbrev>
```

### Example 83. Glossary definitions in text: DocBook5

A glossary definition occurring within the text may or may not include a mention of its term, but its identifier should match the prescribed pattern.

```
<para><phrase xml:id="def_widget" role="definition"
  >A <glossterm xml:id="term_widget">widget</glossterm> is a...</phrase>
</para>

<firstterm xml:id="first_ABC">
  <citereftentry xml:id="ref_ABC">
    <citereftitle xml:id="def_ABC">A Big Company</citereftitle>
    <manvolnum xml:id="term_ABC">ABC</manvolnum>
  </citereftentry>
</firstterm>
```

### Example 84. Glossary Terms defined in a list: DocBook5

A glossary list is a convenient editorial device that facilitates recognition of important terms and their definitions. These terms and definitions will be transcluded into the master glossary as we shall see in the next example.

```
<glosslist>
<glossentry>
  <glossterm xml:id="term_lexical_tagging">Lexical tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_lexical_tagging" role="definition">Tagging
      content with the purpose of identifying lexical components of
      the text. The wordasword, phrase, and symbol elements are
      examples of lexical tagging. </phrase></para>
    </glossdef>
  </glossentry>
<glossentry>
  <glossterm xml:id="term_semantic_tagging">Semantic tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_semantic_tagging" role="definition">Tagging
      content with the purpose of identifying some aspect of its
      semantics. The personname, hardware, and URI elements are
      examples of semantic tagging. </phrase></para>
    </glossdef>
  </glossentry>
<glossentry>
  <glossterm xml:id="term_typographic_tagging">Typographic tagging</glossterm>
  <glossdef>
    <para><phrase xml:id="def_typographic_tagging" role="definition">Tagging
      content with the purpose of affecting the typographic styling.
      The emphasis element is an example of typographic tagging.
      </phrase></para>
    </glossdef>
  </glossentry>
</glosslist>
```

### Example 85. Master Glossary entry: DocBook5

The master glossary list entries are listed alphabetically, each with an identifier that begins with “gloss\_”. The entries consist of a pair of terms and definitions, the content of which may be included therein or transcluded by reference. It is considered good editorial form to include a reference back to the source of the transclusion, and possibly to the first mention if that is deemed appropriate; this enables the reader to examine and consider the definition in the context in which it is originally presented.

```
<glosslist>
<glossentry xml:id="gloss_ABC">
```

```
<glossterm><xref linkend="term_ABC" endterm="term_ABC"/></glossterm>
<glossdef>
  <para><xref linkend="ref_ABC" endterm="ref_ABC" role="transclusion"/></para>
  <para role="parenthetical">(From <xref linkend="first_ABC"/>)</para>
</glossdef>
</glossentry>
...
<glossentry xml:id="gloss_lexical_tagging">
  <glossterm><xref linkend="term_lexical_tagging" endterm="term_lexical_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_lexical_tagging" endterm="def_lexical_tagging" role="transclusion"/></para>
    <para role="parenthetical">(From <xref linkend="def_lexical_tagging"/>)</para>
  </glossdef>
</glossentry>
...
<glossentry xml:id="gloss_semantic_tagging">
  <glossterm><xref linkend="term_semantic_tagging" endterm="term_semantic_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_semantic_tagging" endterm="def_semantic_tagging" role="transclusion"/></para>
    <para role="parenthetical">(From <xref linkend="def_semantic_tagging"/>)</para>
  </glossdef>
</glossentry>
...
<glossentry xml:id="gloss_typographic_tagging">
  <glossterm><xref linkend="term_typographic_tagging" endterm="term_typographic_tagging"/></glossterm>
  <glossdef>
    <para><xref linkend="def_typographic_tagging" endterm="def_typographic_tagging" role="transclusion"/></para>
    <para role="parenthetical">(From <xref linkend="def_typographic_tagging"/>)</para>
  </glossdef>
</glossentry>
...
<glossentry xml:id="gloss_widget">
  <glossterm><xref linkend="term_widget" endterm="term_widget"/></glossterm>
  <glossdef>
    <para><xref linkend="def_widget" endterm="widget" role="transclusion"/></para>
    <para role="parenthetical">(From <xref linkend="def_widget"/>)</para>
  </glossdef>
</glossentry>
</glosslist>
```

## Example 86. Glossary: XHTML5

```
<body>
  <header></header>
  <section id="Glossary" title="Glossary"
    class="glossary"
    epub:type="glossary"
    data-type="glossary"
  >
    <div class="titlepage"><div><div><h1 class="title">Glossary</h1></div></div></div>
```

[...]

```
<dl id="gloss_lexical_tagging" title="Lexical tagging"
  epub:type="glossary" data-type="glossary" >
  <dt class="glossterm"
    epub:type="glossterm" data-type="glossterm"
    ><dfn><a class="backlink phrase" rel="glossterm" rev="glossterm"
```

```
        href="ch12s04.xhtml#term_lexical_tagging"
    >Lexical tagging</a></dfn></dt>
<dd class="glossdef"
    epub:type="glossdef" data-type="glossdef" >
    <p>Tagging content with the purpose of
        identifying lexical components of the text.
        The wordasword, phrase, and symbol elements
        are examples of lexical tagging. </p>
</dd>
</dl>

<dl id="gloss_semantic_tagging" title="Semantic tagging"
    epub:type="glossary"
    data-type="glossary" >

    <dt class="glossterm" epub:type="glossterm" data-type="glossterm"
        ><dfn><a class="backlink phrase" rel="glossterm" rev="glossterm"
            href="ch12s04.xhtml#term_semantic_tagging"
            >Semantic tagging</a></dfn>
    </dt>
    <dd class="glossdef" epub:type="glossdef" data-type="glossdef"

        ><p>Tagging content with the purpose of
            identifying some aspect of its semantics.
            The personname, hardware, and URI elements
            are examples of semantic tagging. </p>
    </dd>
</dl>

<dl id="gloss_typographic_tagging" title="Typographic tagging"
    epub:type="glossary"
    data-type="glossary" >
    <dt class="glossterm" epub:type="glossterm" data-type="glossterm"
        ><dfn
    ><a class="backlink phrase" rel="glossterm" rev="glossterm"
        href="ch12s04.xhtml#term_typographic_tagging"
        >Typographic tagging</a></dfn>
    </dt>
    <dd class="glossdef" epub:type="glossdef" data-type="glossdef"
        ><p>Tagging content with the purpose of
            affecting the typographic styling.
            The emphasis element is an example of
            typographic tagging. </p></dd>
</dl>
```

## Index

There are many different kinds of index. Ours is an index of primary, and optionally secondary, and tertiary terms along with a referent in the form of a locator, or list of locators. The referent generally takes the form of a page number in print media; in electronic media, referents may present hypertext-enabled information intended to guide the reader to a particular chapter, section, paragraph, table, sidebar, figure, example, or a definition in glossary entry. In early drafts of the print edition, we had an index of abbreviations, acronyms, and initialisms that was once characterized as “a kind of secret decoder ring.” We later integrated that index with the glossary.

You can refer to the index of this work. The index takes the form of a list which is constructed during the book building process. As each index term is encountered, the values are recorded along with identifiers that are left behind in the XHTML5 output. When the book building process reaches the

index, it starts to sort all of the collected entries, according to their primary, secondary and tertiary values. When sorted, a complex XHTML5 list is produced to represent this hierarchical structure.

It sounds good in theory. As they say, “In theory, there is no difference between theory and practice. But, in practice, there is.”<sup>9</sup>

Creating an index with DocBook5 and its downstream tools is a bit challenging, both in terms of the precision with which the markup and spaces must be managed, and also in terms of identifying what, exactly, is being indexed. First of all, it is only safe to insert DocBook5<indexterm> markup in a paragraph or in an <info> element.

### Indexing Footnotes

In the first edition, both print and e-book, we stretched indexing convention by indexing the footnotes; that's generally not done. Because the disciplinary content of this work contains much of the intellectual content, we thought it fitting. Anyway, long story short; when we upgraded from DocBook4.5 to DocBook5, we lost the ability to have index terms in footnotes. We got our old behavior back in DocBook5.1.

## Index Terms in Paragraph Content

One must exercise such extreme care in placing index terms within a paragraph content, lest unwanted spaces appear at the beginning of a paragraph, that we have exercised editorial rule against them. Basically, if it is worth indexing, it has an identity, so all index terms now go into info elements, unless there is a strong editorial or production reason otherwise.

### Example 87. Index Entry White-space Considerations

This is ok:

```
<para><indexterm><primary>term</primary></indexterm>When we discuss
<glossterm linkend="gloss_term">term</glossterm>
[...]
```

This is not ok:

```
<para>
<indexterm><primary>term</primary></indexterm>When we discuss
<glossterm linkend="gloss_term">term</glossterm>
[...]
```

Neither is this:

```
<para><indexterm><primary>term</primary></indexterm>
When we discuss
<glossterm linkend="gloss_term">term</glossterm>
[...]
```

## Index Terms in Info Elements

When indexterm are inserted within an <info> element, we always use @zone to point to its referent. The granularity with which one chooses the referent is typically an identifiable structural element, such as a section, paragraph, figure, and example; or a phrase-level semantic element, such as citation or personname.

---

<sup>9</sup>According to Wikipedia: Attributed to Jan L. A. van de Snepscheut [[http://en.wikiquote.org/wiki/Jan\\_L.\\_A.\\_van\\_de\\_Snepscheut](http://en.wikiquote.org/wiki/Jan_L._A._van_de_Snepscheut)] in: Doug Rosenberg and Matt Stephens (2007) *Use Case Driven Object Modeling with UML Theory and Practice* p. xxvii; Quote is also cited without attribution in Doug Rosenberg and Kendall Scott (2001) *Applying Use Case Driven Object Modeling With UML*, p. 1

### Example 88. Indexing Structural Elements

```
<para xml:id="para_reprehenderit"><info><termset>
  <indexterm zone="para_reprehenderit">
    <primary>reprehenderit</primary>
  </indexterm>
</info>Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do eiusmod tempor ut labore et dolore magna aliqua.
<phrase role="definition">Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu fugiat nulla pariatur. </phrase>
Exceper sint occaecat cupidatat endente non proident, id est laborum.
```

When we include an index term within a paragraph with the intent of indexing a sub-component of the paragraph, such as a definition or a person's name, we can use `@zone` to specify identifier of the target element.

### Example 89. Indexing Phrases

```
<para><info><termset>
  <indexterm zone="def_reprehenderit">
    <primary>reprehenderit</primary>
  </indexterm>
</info>Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor ut labore et dolore magna aliqua.
<phrase xml:id="def_reprehenderit" role="definition">Duis aute irure dolor in reprehenderit
in voluptate velit esse cillum dolore eu fugiat nulla pariatur. </phrase>
Exceper sint occaecat cupidatat endente non proident, id est laborum.
```

### Example 90. Index entry: DocBook5

```
<info><termset>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
  </indexterm>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
    <secondary>ePub.type</secondary>
  </indexterm>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
    <secondary>ePub.type</secondary>
    <tertiary>index</tertiary>
  </indexterm>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
    <secondary>ePub.type</secondary>
    <tertiary>index-editor-note</tertiary>
  </indexterm>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
    <secondary>ePub.type</secondary>
    <tertiary>index-entry, -list</tertiary>
  </indexterm>
  <indexterm linkend="PRODNOTES.10.4">
    <primary>index output</primary>
    <secondary>ePub.type</secondary>
    <tertiary>index-group</tertiary>
```

```
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-headnotes</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-legend</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-locator, -list, -range</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-term, -categories, -category</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-xref-preferred</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>ePub.type</secondary>
  <tertiary>index-xref-related</tertiary>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>preferred</secondary>
  <see>reference</see>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>related</secondary>
  <seealso>reference</seealso>
</indexterm>
<indexterm linkend="PRODNOTES.10.4">
  <primary>index output</primary>
  <secondary>secondary</secondary>
  <tertiary>tertiary</tertiary>
</indexterm>
</itermsset>
</info>
```

## Index Processing

Did I mention that creating an index was challenging? As an author/editor, I want to be able to distinguish a particular word, phrase, sentence, paragraph, or passage to convey meaning behind the prose, to underscore its essence, as when we identify a glossary term, proper name, abbreviation, acronym, citation, email address, or URI.

Now, one might think that the engine that builds these books could infer an indexentry from a glossary term, or a definition, or from many other structural and semantic elements, but the index machinery



is old and crufty, it seems that nobody is up to the challenge of improving it, and we are glad that it works as well as it does.

Features we would like:

- `indexterm @zone` inferred from parent element, where possible, such as chapter, section, para, etc.
- `@significance="preferred"` pass through to XHTML5 to highlight defining terms
- imputation of index terms from semantic elements, such as names, abbreviations, and acronyms.
- better the section called “XHTML5 Output” with `@epub:type`

## XHTML5 Output

The generated XHTML5 output is a complex list structure that is not easily disassembled, which prevented us from having dictionary style page headers in the print index.

### Editors' Note

This XHTML5 structure limits what we can do with its presentation and comprehension. We would prefer a tighter list format, with expandable secondary and tertiary entries, and simpler referent lists. Also want to conform with ePub index structure and attributes. Hard to fix in the transform from DocBook5 to XHTML5; may have to post-process before packaging the XHTML5 into the ePub.

### Example 91. Idealization of XHTML5 Result

Refer to the EPUB Indexes specification [IDX] for definitions of these terms and content structuring requirements.

```
<section id="Index" title="Index"
  class="index"
  epub:type="index"
  data-type="index">

  <header epub:type="index-headnotes">
    <p>The following abbreviations are used in this index.</p>
    <dl epub:type="index-legend">
      <dt>Ch.</dt><dd>Chapter</dd>
      <dt>§</dt><dd>Section</dd>
      <dt>¶</dt><dd>Paragraph</dd>
      <dt>Sb.</dt><dd>Sidebar</dd>
      <dt>Def.</dt><dd>Definition</dd>
    </dl>
    <p>The following formatting conventions are used in this index-</p>
    <dl epub:type="index-legend">
      <dt>bold text</dt><dd>term defined</dd>
      <dt>superscript</dt><dd>indicates endnote number and discipline</dd>
    </dl>
  </header>

  <div class="indexdiv"
    epub:type="index-group"
    title="A">
    <h3>A</h3>
    [...]
  </div>
  [...]
</div class="indexdiv" epub:type="index-group">
```

```
<h3>I</h3>
<dl epub:type="index-entry-list">
  <dt epub:type="index-entry"
    ><span class="primary"
      epub-type="index-term">i...</span>:
    <span epub:type="index-locator-list">
      <a class="indexterm" epub:type="index-locator" href="...">[...]</a>;
      <a class="indexterm" epub:type="index-locator" href="...">[...]</a>
    </span></dt>
  <dd></dd>
</dt>[...]</dt>
<dd></dd>

<dt epub-type="index-entry"
  ><span class="primary"
    epub-type="index-term">index output</span>:
  <span epub:type="index-locator-list">
    <a class="indexterm" epub:type="index-locator"
      href="ch12s10.xhtml#PRODNOTES.10.4"
      >§12.10.4, “XHTML5 Output”</a>;
    <a class="indexterm" epub:type="index-locator"
      href="ch12s10.xhtml#d0e53653"
      >in §12.10.4, “XHTML5 Output”</a></span></dt>
<dd>
  <dl class="index-secondary">
    <dt><span class="secondary"
      epub-type="index-term">CSS styling: </span>
      <a class="indexterm" epub:type="index-locator"
        href="ch12s10.xhtml#d0e53736"
        >in §12.10.4, “XHTML5 Output”</a></dt>
    <dd></dd>
    <dt><span class="secondary" epub-type="index-term">ePub:type: </span>
      <a class="indexterm" epub:type="index-locator"
        href="ch12s10.xhtml#PRODNOTES.10.4"
        >§12.10.4, “XHTML5 Output”</a></dt>
    <dd>
      <dl class="index-tertiary">
        <dt><span class="tertiary"
          epub-type="index-term">index: </span>
          <a class="indexterm" epub:type="index-locator"
            href="ch12s10.xhtml#PRODNOTES.10.4"
            >§12.10.4, “XHTML5 Output”</a></dt>
        <dd></dd>
        <dt><span class="tertiary"
          epub-type="index-term">index-editor-note: </span>
          <a class="indexterm" epub:type="index-locator"
            href="ch12s10.xhtml#PRODNOTES.10.4"
            >§12.10.4, “XHTML5 Output”</a></dt>
        <dd></dd>
        <dt><dt><span class="tertiary"
          epub-type="index-term">index-entry, -list: </span>
          <a class="indexterm" epub:type="index-locator"
            href="ch12s10.xhtml#PRODNOTES.10.4"
            >§12.10.4, “XHTML5 Output”</a></dt>
        <dd></dd>
        <dt><dt><span class="tertiary"
          epub-type="index-term">index-group: </span>
          <a class="indexterm" epub:type="index-locator"
```

```

        href="ch12s10.xhtml#PRODNOTES.10.4"
        >§12.10.4, "XHTML5 Output"</a></dt>
    <dd></dd>
    [...]
</dl>>
</dd>
<dt epub:type="index-xref-preferred">preferred (see reference)</dt>
<dd></dd>
<dt>related:
    <a class="indexterm" epub:type="index-locator"
        href="ch12s10.xhtml#PRODNOTES.10.4"
        >§12.10.4, "XHTML5 Output"</a></dt>
    <dd>
        <dl class="index-tertiary">
            <dt epub:type="index-xref-related">(see also reference)</dt>
            <dd></dd>
        </dl></dd>
    <dt>secondary</dt>
    <dd>
        <dl class="index-tertiary">
            <dt><span class="tertiary" epub-type="index-term">tertiary: </span>
                <a class="indexterm" epub:type="index-locator"
                    href="ch12s10.xhtml#PRODNOTES.10.4"
                    >§12.10.4, "XHTML5 Output"</a></dt>
                <dd></dd>
            </dl>
        </dd>
    </dl>
</dd>
</div class="indexdiv"
    epub:type="index-group"
    title="[A-Z]"><h3>[A-Z]</h3> <!-- Fill in @title with [A-Z] -->
    <dl epub:type="index-entry-list"> <!-- Distinguish entry, term, locator -->
        <dt epub-type="index-entry"> <!-- Put punctuation on outside side of span and a -->
            <span class="primary"
                epub-type="index-term"
                title="TERM">TERM</span>: <!-- Fill in @title for primary and secondary -->
            <span epub-type="index-locator-list"> <!-- Distinguish locators by @rel/rev and @class -->
                <a class="indexterm section"
                    rel="section" rev="index-locator"
                    epub:type="index-locator"
                    >[...]</a> ; <!-- Punctuation -->
                <a class="indexterm sidebar"
                    rel="sidebar" rev="index-locator"
                    epub:type="index-locator"
                    >[...]</a> ;
                <a class="indexterm definition"
                    rel="definition" rev="index-locator"
                    epub:type="index-locator"
                    >[...]</a>
            </span></dt>
        <dd>
            <dl class="index-secondary">
                <dt><span class="secondary"
                    epub-type="index-term"
                    title="secondary term">secondary term</span>:
                    <a class="indexterm" epub:type="index-locator">[...]</a></dt>

```

```
<dd>
  <dl class="index-tertiary">
    <dt><span class="tertiary"
      epub-type="index-term">tertiary term</span>:
      <a class="indexterm"
        epub:type="index-locator"
        >[...]</a></dt>
    <dd></dd>
  </dl>
</dl>
</div>
```

### **Warning**

We need to differentiate between the index term and the referents. We also need to incorporate `epub:type= index-group, index-entry-list, index-entry, index-term, index-locator, index-locator-list, index-locator-range, index-xref-related, index-term-category, index-term-categories`. Specs to follow.

## **Example 92. CSS Styles and Interactions**

```
section.index {
  margin-left: 1.5em ;
  line-height: 1.1 ;
}

div.indexdiv {
  padding: 0 ;
  widows: 9 ;
  orphans: 8 ;
}

section.index div.indexdiv>h3 {
  string-set: index-division content () ; }

section.index div.indexdiv dl {
  line-height: 1 ;
  margin-top: 0 ;
  margin-bottom: 0 ;
  widows: 5 ;
  orphans: 3 ;
}

section.index div.indexdiv dl dt {
  text-indent: -1.5em ;
  margin-top: 0 ;
  margin-bottom: 0 ;
  padding: 0 ;
  widows: 2 ;
  orphans: 2 ;
}

section.index div.indexdiv dl dt a.indexterm {
}

section.index div.indexdiv dd:empty { line-height: 0 ; display: none ; }
```

```
section.index div.indexdiv dd { line-height: 1 }

section.index>div.indexdiv>dl {
  string-set: index-division normal ;
  string-set: index-primary normal ;
  string-set: index-secondary normal ;
  string-set: index-tertiary normal ;
}
section.index>div.indexdiv>dl>dt>span.primary {
  string-set: index-primary content () ;
  string-set: index-secondary normal ;
  string-set: index-tertiary normal ;
}
section.index>div.indexdiv>dl>dd>dl>dt>span.secondary {
  string-set: index-secondary content () ;
  string-set: index-tertiary normal ;
}
section.index>div.indexdiv>dl>dd>dl>dd>dl>dt>span.tertiary {
  string-set: index-tertiary content () ;
}
```

## Accessibility

In preparing the electronic books for publication, we were fortunate that our transformation engine was already equipped to incorporate some markup for Accessible Rich Internet Applications(ARIA) [<http://www.w3.org/TR/wai-aria/roles>].

One of the ways that we have enabled accessibility is by including the semantic labeling in the XHTML5 markup. In particular, we include HTMLbook:data-type attributes and ePub:type attributes with eduPub extensions. We also label many of the same elements with DocBook5 labels. We referred to the ePub3 Accessibility Guidelines [<http://www.idpf.org/accessibility/guidelines/>] for ePub:type values and examples to guide our transformations from DocBook5.

## Book Navigation

### Example 93. Book Navigation: XHTML5

The Chapters and all major sections of the book will have a navigation menu in its header, except for the Cover, Table of Contents, Titles, Credits, Copyrights, Dedication, Abstract, and Colophon. In the Chapters, all top-level sections will have a navigation menu in its footer.

```
<div class='nav'>
<nav class='menu'>
  <ul>
    <li><a href='#'>Front</a>
    <ul>
      <li><a rel='cover' title='Cover' href='cover.xhtml'>Cover</a></li>
      <li><a rel='title' title='Titles' href='index.xhtml'>Titles</a></li>
      <li><a rel='dedication' title='Dedication' href='index.xhtml#Dedication'>Dedication</a></li>
      <li><a rel='foreword' title='Foreword' href='pr01.xhtml'>Foreword</a></li>
      <li><a rel='preface' title='Preface' href='pr02.xhtml'>Preface</a></li>
      <li><a rel='abstracto' title='Abstract' href='pr03.xhtml'>Abstract</a></li>
    </ul>
  </li>
  <li><a href='#'>Chapters</a>
  <ul>
    <li><a rel='chapter' title='Chapter 1. ...' href='ch01.xhtml'>Chapter 1</a></li>
```

```
<li><a rel='chapter' title='Chapter 2. ...' href='ch02.xhtml'>Chapter 2</a></li>
<li><a rel='chapter' title='Chapter 3. ...' href='ch03.xhtml'>Chapter 3</a></li>
[ ... ]
<li><a rel='chapter' title='Chapter 11. ...' href='ch11.xhtml'>Chapter 11</a></li>
</ul>
</li>
<li><a href='#'>Back</a>
<ul>
<li><a rel='acknowledgments' title='Acknowledgments' href='pr03.xhtml'>Acknowledgments</a></li>
<li><a rel='bibliography' title='Bibliography' href='bi01.xhtml'>Bibliography</a></li>
<li><a rel='glossary' title='Glossary' href='go01.xhtml'>Glossary</a></li>
<li><a rel='index' title='Index' href='ix01.xhtml'>Index</a></li>
<li><a rel='colophon' title='Colophon' href='co01.xhtml'>Colophon</a></li>
</ul>
</li>
<li style='float:right;'><a href='#'>Quick Links</a>
<ul>
<li><a rel='this chapter' title='Chapter N. ...' href='ch01.xhtml'>This Chapter</a></li>
<li><a rel='this section' title='...' href='ch01s02.xhtml'>This Section</a></li>
<li><a rel='previous' title='...' href='ch01s01.xhtml'>Previous Section</a></li>
<li><a rel='next' title='...' href='ch01s03.xhtml'>Next Section</a></li>
</ul>
</li>
</ul>
</nav>
</div>
```

## Bibliography Navigation

### Example 94. Bibliography Navigation: XHTML5

The Bibliography will have a navigation menu at the head of each alphabetic section, where the corresponding `@href="bi01.xhtml#Bibliography_[A-Z]"` is replaced by `@id="Bibliography_[A-Z]"`. In cases where an alphabetic section is missing, its `@id` will be set on the subsequent section such that there will be two or more `@id` set in that nav list. If a given section contains less than four entries, it may be combined with subsequent sections, up to a total of ten entries, such that, for example, X, Y, and Z sections might have a single navigation menu.

```
<nav class="biblionav">
<ul>
<li><a id="Bibliography_A">A</a></li>
<li><a href="bi01.xhtml#Bibliography_B">B</a></li>
<li><a href="bi01.xhtml#Bibliography_C">C</a></li>
[ ... ]
<li><a href="bi01.xhtml#Bibliography_Z">Z</a></li>
</ul>
</nav>
```

## Glossary Navigation

### Example 95. Glossary Navigation: XHTML5

The Glossary will have a navigation menu at the head of each alphabetic section, where the corresponding `@href="bi01.xhtml#Glossary_[A-Z]"` is replaced by `@id="Glossary_[A-Z]"`. In cases where an alphabetic section is missing, its `@id` will be set on the subsequent section such that there will be two or more `@id` set in that nav list. If a given section contains less than four entries, it may be combined with subsequent sections, up to a total of ten entries, such that, for example, X, Y, and Z sections might have a single navigation menu.

```
<nav class="glossnav">
<ul >
  <li><a id="Glossary_A">A</a></li>
  <li><a href="go01.xhtml#Glossary_B">B</a></li>
  <li><a href="go01.xhtml#Glossary_C">C</a></li>
  [ ... ]
  <li><a href="go01.xhtml#Glossary_Z">Z</a></li>
</ul>
</nav>
```

## Index Navigation

### Example 96. Index Navigation: XHTML5

The Index will have a navigation menu at the head of each alphabetic section, where the corresponding `@href="bi01.xhtml#Index_[A-Z]"` is replaced by `@id="Index_[A-Z]"`. In cases where an alphabetic section is missing, its `@id` will be set on the subsequent section such that there will be two or more `@id` set in that nav list. If a given section contains less than four entries, it may be combined with subsequent sections, up to a total of ten entries, such that, for example, X, Y, and Z sections might have a single navigation menu.

```
<nav class="indexnav">
<ul >
  <li><a id="Index_A">A</a></li>
  <li><a href="ix01.xhtml#Index_B">B</a></li>
  <li><a href="ix01.xhtml#Index_C">C</a></li>
  [ ... ]
  <li><a href="ix01.xhtml#Index_Z">Z</a></li>
</ul>
</nav>
```

## eBook Readers

### The ePub3 Readers

First tier ePub3 readers

The best ePub3 readers present a flowed text media presentation of a book with features such as interactive selection of displayed content, context-sensitive tables of contents, navigation cues, alternative text for images, and an interactive quiz. Each reader offers the reader some control over some over content formatting characteristics, including font selection, background color, and so on. We prefer to read the book in a continuous single column view, however these readers also offer multi-column paged views. Some provide more comprehensive page navigation cues than others.

- Adobe Digital Editions V4 (**ios osx windows**)  
epubReadingSystem.name = "RMSDK" Has its own navigation viewport. Has two reading modes: single and double page. tdo-paranotes.js does not work in single page mode.
- Calibre V2.25 (**linux osx windows**) Has its own viewport feature for footnote links encoded with `epub:type="noteref"`. Has its own navigation viewport.
- iBooks (**ios osx**) Features pop-up footnotes. `epubReadingSystem.name = "iBooks"` Has its own viewport

feature for footnote links encoded with `epub:type="noteref"`. Poor navigation through native TOC menu. Responds to `tdo-nav`. Has two reading modes: single and double page. `tdo-paranotes.js` does work in both page modes.

- Radium (Chrome) (**android ios linux osx windows**) `epubReadingSystem.name` = "Radium". Has its own navigation viewport.

#### Second tier ePub3 Readers

A flowed text media presentation of a book with table of contents, navigation cues, hypertext references, alternative text for images, and inhibited CSS interactivity. Each offers the reader some control over some over content formatting characteristics, including font selection, background color, and so on. Both provide previous/next navigation cues.

- Aldiko (**android**) Publishing network. Adobe Reader Mobile SDK white-label reader.
- Azardi (**android ios linux osx windows**) Poor navigation through native TOC menu. Responds to `tdo-nav`. `tdo-paranotes.js` does not work in flow mode, but seems to work in one and two page modes..
- Kitabu V1.08 (**osx**)
- Lucifox (Firefox+XUL) (**android ios linux osx windows**) Has its own hypertext viewport feature. Responds badly to XHTML5 form elements such as `button`, `select`, and `option`; hence, we cannot reliably deploy a polyvalent edition on this platform. (Document lives inside of `window.viewpub` )
- EPUBReader (Firefox) (**android ios linux osx windows**) Features book formatting controls. Responds badly to XHTML5 form elements such as `button`, `select`, and `option`; hence, we cannot reliably deploy a polyvalent edition on this platform. (Document lives inside of `window.epubreader` )

#### Note

By editorial decision, `tdo-toggle` and `tdo-viz` will not run in these readers.

#### ePub3 Reader with Formless XHTML5 + CSS

A flowed text media presentation of a book with features such as tables of contents, hypertext references, alternative text for images, and inhibited CSS interactivity. The ePub3 readers in this class are targeted at specialized network markets of book consumers.

Formless XHTML5 and lack of JavaScript prevents content changes and interactivity. Responds in unexpected ways to the use of XHTML5 form elements such as `button`, `select`, and `option`; hence, we cannot deploy the polyvalent edition on this platform. For some reason, ADE does not handle links in nav menus well; hence, we cannot deploy the alphabetic link lists in `biblionav`, `glossarynav`, and `indexnav`.

- Bluefire (**android, ios**) This application is intended for book borrowing and Adobe DRM-protected content. White-label ebook reader. User control over font face, size, line spacing,



margins, and justification. text and background colors, and packaged themes. Limited navigation interactions. Lacks JavaScript. Having trouble with biblionav, glossarynav, and indexnav. Adobe Reader Mobile SDK white-label reader.

- FBReader (**android linux osx windows**) Serves as intermediary to a distributed book storage network, built on Google Drive. Epub2 only. Lacks JavaScript. Very poor CSS.
- Lektz Readers (android ios jvm osx windows) Publishing network.

#### PDF Reader

A paged media presentation of a book with features, such as table of contents and hypertext references. Paged media improves cognition and memory. Hypertext page references facilitate discovery and selection. We rely upon the presence of `@condition="PDF"` to signal DocBook5 content that is intended for inclusion/exclusion profiling.

- Adobe Reader

#### Physical Artifacts

Physical artifacts may be produced in printed and Braille book forms from the ePub and PDF manifestations. Interactions with the physical book involve discovery, selection, and reading. Navigation cues, and references from the tables of contents, glossary, and index support discovery. We rely upon the presence of `@condition="print"` or `@condition="braille"` to signal DocBook5 content that is intended for inclusion/exclusion profiling and `@data-condition` in the XHTML5 to signal content that is intended for conditional rendering.

#### ePub3-derivative ebook Readers

A paged media presentation of a book with features, such as table of contents and hypertext references. Interactions are limited to those offered by the reader itself, such as font choices and other stylistic parameters. A lack of JavaScript and CSS interactivity makes these the least versatile platforms for content consumption.

- Kindle (kindle) Transformation required; using Calibre. Separate edition.
- Kobo (kobo) Transformation required; using Calibre. Separate edition.

## Implementing Interactions

"XML gives Java something to do."

—Jon Bosak -- XML, Java, and the future of the Web [<http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.htm>]

Documents encoded with hierarchically structured markup, whether DocBook5 or XHTML5, are ideally suited for discovery, selection, and interaction by JavaScript, and its convenient cousin JQuery. This book offers several features that rely upon XHTML5, CSS, and DOM scripting capabilities. It's tricky stuff, to be sure. A missing space or semicolon could lead to hours of debugging. Unexpected incompatibilities among browsers and ePub3 readers

## The Scripts

All of the scripts rely upon the presence of JavaScript and JQuery, `tdo-viz` relies upon `d3` and some build-generated JSON data files, `tdo-nav` relies upon build-generated data in the XHTML5 files, and `tdo-misc` accesses files within the archive. The `tdo-paranotes` script relies upon custom markup and tagging within each XHTML5 file.

`core.js`

The `core.js` script collects and calculates information from its environment. These variables can and should be used by other scripts to determine whether to exercise any given feature, based on device or reader capability, screen dimensions, column flow and layout characteristics, or quasi-editorial aesthetic judgements. Variables currently include `window.reader`, `OS`, `browser`, `chapterFileName`, `deviceType`, `screenWidth`, `screenHeight`, `viewportWidth`, `viewportHeight`, `windowWidth`, `windowHeight`, `columnWidth`, `columnHeight`, `numberOfColumns`. We also set the value of `window.features['viz|nav|quiz|...'] = "[enabled|disabled]"` and `window.features['misc']` with a list of selectors to specify over which anchors the script will operate. The list of selectors varies depending on the reader/platform combination.

So, for example:

```
if (window.reader== "ibooks") { [code] }
if (window.features['nav'] == "enabled") { [code] }
if (window.features['viz'] == "enabled") { [code] }
```

`tdo-paranotes.js`

This script should only be deployed in readers that are **known** to execute correctly. Currently that only includes iBooks on Mac in two page mode.

The `tdo-paranotes.js` script (see Example 62, “Paranotes: JQuery”) enables show/hide of individual endnotes placed at the end of the paragraph from which it is referenced. The script must first test whether or not paragraph endnotes are being used in the document, which is a build option, then it must ask whether the OS/reader combination supports the script in its current layout/pagination mode. Initial tests have shown that iBooks, for example, perform well in all modes. The Safari plug-ins, Lucifox and EPUBReader, seem to have problems with show/hide.

`tdo-nav.js`

The `tdo-nav.js` script collects data from the XHTML files to build a small menu that appears at the start and end of many content file, although not all. It is not quite complete, and it should only be deployed on specific readers. In many readers it exhibits extremely undesirable behaviour. This script requires some further work on its data gathering, and would benefit from an aesthetic re-design along with corresponding CSS and scripting work.

`tdo.toggle.js`, `tdo-config.js`, `tdo.viz.js`

These scripts should only be deployed in readers that are **known** to execute correctly..

The disciplinary selection script, `tdo-toggle.js` creates a selection list menu for hiding/showing content based on the `@data-audience` value on paragraphs, asides, and list items in the XHTML5 markup.

The corresponding visualization script, `tdo-viz.js` use data files containing JSON data related to tagged endnotes: `tdo-viz-book-breakdown.js` and `tdo-viz-chapters-breakdown.js` and `tdo-viz-disciplines-breakdown.js`. It also provides an interface through which the user may select to hid/show disciplinary content, and record the user settings within the book and during for future

sessions. However, local storage has proven to be elusive on different readers, and currently only works reliable in iBooks.

The related script `tdo-toggle-config.js` populates a set of `window.toggle` variables, including categories and groups, which correspond to disciplines and edition groupings. In addition, `window.toggle` lists types of book component, with detailed breakdown of the members of a part, front, back, chapters, chapter, section, display feature, paragraphs, paragraph, as well as different types of semantic phrases.

`tdo-quiz.js` The `tdo-quiz.js` relies upon content provided during the build process. We have good feedback with respect to this script's ability to run correctly in JS/JQ-enabled readers, and it falls back gracefully in less capable readers. Although we do set `window.features['quiz']`, we don't actually pay attention to it.

`tdo-misc.js` The `tdo-misc.js` script was designed to provide a pop-up hypertext viewport when hovering over a content reference link. We have continued to encounter various issues related to file access on different platforms. This feature is not ready for prime time, but it will be added to the "*Experimental Polyvalent Edition*". This script uses `windows.features['misc']` to get a list of anchor selectors.

```
anchorSelectors = window.features['misc'];
```

### Example 97. epubReadingSystem object

We access `epubReadingSystem` information in `core.js`

<http://epubzone.org/news/epub-3-and-interactivity>

```
document.onload = function () {  
  if (navigator && navigator.hasOwnProperty('epubReadingSystem')) {  
    runEbookScripts(); // This is the only entry point to script documents  
  }  
};  
  
<script type="text/javascript">  
alert("Name: " + navigator.epubReadingSystem.name +  
      " / version: " + navigator.epubReadingSystem.version +  
      " / layoutStyle " + navigator.epubReadingSystem.layoutStyle);  
</script>
```

Detect features Through its `hasFeature()` method, `epubReadingSystem` allows content authors to query the exact characteristics of the reading system's scripting support. The following features are required to return true or false by any system that implements `epubReadingSystem`:

dom-manipulation	Spine-level scripts are allowed to make structural changes to the DOM.
layout-changes	Spine-level scripts may modify attributes and CSS styles that affect content layout.
touch-events	The device supports touch events and the reading system passes touch events to the content.
mouse-events	The device supports mouse events and the reading system passes mouse events to the content.
spine-scripting	Spine-level scripting is supported.

## The Quiz

The first edition print and epub books all included a section at the end of chapters 2 through 9 with key points from the chapters. When we upgraded to epub3, we decided to turn those key points into an interactive quiz, based upon DocBook's question and answer list markup, when it appears, stand-alone, within the last section of a chapter. (Attempting to place a quiz in any other part of the book will yield undefined results.)

### Example 98. Quiz: DocBook5

```
<qandaset role="quiz">
  <qandaentry>
    <question><para>What is a book?</para></question>
    <answer>
      <para><xref linkend="Answer-Book" endterm="Answer-Book" role="transclusion"/></para>
      <para role="parenthetical">(See <xref linkend="Answer-Book"/>)</para></answer>
    </qandaentry>
    <qandaentry>
      <question><para>Question 2?</para></question>
      <answer><para>Answer 2</para></answer>
    </qandaentry>
    <qandaentry>
      <question><para>Question 3?</para></question>
      <answer><para>Answer 3</para></answer>
    </qandaentry>
    <qandaentry>
      <question><para>Question 4?</para></question>
      <answer><para>Answer 4</para></answer>
    </qandaentry>
    <qandaentry>
      <question><para>Question 5?</para></question>
      <answer><para>Answer 5</para></answer>
    </qandaentry>
    <qandaentry>
      <question><para>Question 6?</para></question>
      <answer><para>Answer 6</para></answer>
    </qandaentry>
  </qandaset>
```

### Example 99. Quiz: XHTML5 and JavaScript

The transformation result includes a header with user-interface elements, followed by section that contains another user-interface element and a JavaScript that initializes itself and then loads the named div element that follows. The first div element is used by the JavaScript to find the collection of quiz-data, sets of quiz-item with question and answer members.

```
<body data-type="book" >
  <header>
    <div id="quiz">
      <h2 class="title">Self-Review</h2>
      <div class="question-section"><h3>Question</h3>
        <div class="question"></div>
      </div>
      <p><input type="button" class="show" value="Show Answer" />
        <input type="button" class="next" value="Next Question" />
        <input type="button" class="previous" value="Previous Question" />
        <input type="button" class="quiz-controller" value="Back to Keypoints" />
      </p>
    </div>
  </header>
</body>
```

```
</p>
<div class="answer-section">
  <h3>Answer</h3><div class="answer"></div>
</div>
</div>
</header>

<section id="PRODNOTES.14"
  title="13.14. Key Points in Chapter Twelve"
  class="section keypointonlyDIV"
  epub:type="subchapter">
  <div class="titlepage"><div><div>
    <h2 class="title" style="clear: both">12.14. Key Points in Chapter Twelve</h2>
  </div></div></div>

  <button name="Quiz"
    type="button"
    class="quiz-controller"
    >Switch to Quiz Mode</button>

<script type="text/javascript">
var quiz = new Quiz();
window.addEventListener("load",function() {
  quiz.init({ quiz: { id: "quiz"}, section: { id: "PRODNOTES.14"} });
  quiz.loadData($(document.body).find(".quiz-data")[0]);
});
</script>
```

### Example 100. Quiz example continued: XHTML5

```
<div class="quiz-data">

<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question">
    <p>What is a book?</p>
  </div>
  <div class="answer" epub:type="answer">
    <p>A book is a collection of conceptual elements [...] </p>
    <p class="parenthetical">(See
      <a class="xref" href="ch12.xhtml#Answer-Book"
      >§12.1.1, “Major Structural Components of a Book”</a></p>
  </div>
</div>

<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question"><p>Question 2?</p></div>
  <div class="answer" epub:type="answer"><p>Answer 2</p></div>
</div>

<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question"><p>Question 3?</p></div>
  <div class="answer" epub:type="answer"><p>Answer 3</p></div>
</div>

<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question"><p>Question 4?</p></div>
  <div class="answer" epub:type="answer"><p>Answer 4</p></div>
</div>
```

```
<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question"><p>Question 5?</p></div>
  <div class="answer" epub:type="answer"><p>Answer 5</p></div>
</div>
<div class="quiz-item" epub:type="qna">
  <div class="question" epub:type="question"><p>Question 6?</p></div>
  <div class="answer" epub:type="answer"><p>Answer 6</p></div>
</div>
</div>
```

## TDO Viewport

TDO Viewport provides readers with an opportunity to preview the target of various hypertext links within the book within a pop-up non-modal viewport, including: footnote call marker links to footnotes; cross-references to chapters, sections, sidebars, tables, figures, and examples; citation and cited title links into the bibliographic entries; and, first term and gloss term references into the glossary. Put simply, when one hovers the mouse pointer over an enabled link, a pop-up viewport appears showing all or part of the content of the target. The viewport is designed to size itself up to a maximum threshold based upon the current window or column dimensions. The viewports are typically tall enough to accommodate many smaller objects, such as footnotes, bibliographic and glossary entries, and so on. For larger objects, such as sections and sidebars, any content that does not fit is simply from view inside of the viewport. Naturally, activating the link with a click will cause link traversal as usual. Making this work requires two pieces of trickery: the first being making and hiding copies of link targets, and the second being presentation of a viewport at the right place on the screen in over a dozen different readers.

One might think that getting a copy of the identified resource would be just as easy as performing an HTTP Request. After all, we have the `@href` value. But it turns out that we cannot depend on all readers allowing even local file access, but where we can the performance impact is so severe that we ruled it out. Next, we considered placing copies of referenced objects in the XHTML5 document footer, using elements whose CSS rules included `:hidden`, but we were stymied by simple readers like FBReader, which do not obey CSS hidden. So, now we have to consider gathering the referents during the build process, stripping out any internal IDs and obscuring any `@hrefs`, and perhaps eliding the text for brevity, and packing them into a JSON data package that the scripts can access at will.

### Hypertext Preview Viewport

The first problem is how or from where we get duplicates of cross-referenced content. The second problem is detecting that a user is expressing interest in a typed link by hovering over it. The third problem is figuring out the current display context and presenting a well-positioned/sized floating sidebar-like device, with the object content in it, and if the object content is too long to fit in the sidebar, it just flows away inside the box and does not flow out or spill over.

Opening files is a problem on some platforms. We would prefer to fetch the content at the moment that the hover occurs, but the readers/browsers don't seem to like it when we do that.

JavaScript-enable or not JS-enabled? In non-JS readers, we cannot simply add copies of content to end of file because it will show up in non-JS readers, in spite of CSS :hidden rules. So, if we want to work in JS and non-JS readers, we must insert the duplicates by script. Either way, a lot of content is being duplicated, often more than once.

There are several approaches that we could take.

- The current working solution is to replicate what Benjamin Finn has done with his version of tdo-misc. He post-processes the ePub3 archive files with a PHP script (because he knows PHP and not XSLT so much). For every unique cross-reference, he gets an elided copy of the target resource, stripping out @id to avoid name collision, and obfuscating or eliminating other attributes. His PHP script grabs up to five paragraphs of content from the target, closing incomplete elements as necessary. It also discards "Navigating this Chapter" and the new Statistics sidebars. This process bulks up a file. The index is the most extreme example of hypertext linking in the book; the file grows from 500kb to 2.5MB. But the effect is phenomenal. Using an index with the aid an hypertext preview viewport changes everything.
- Another idea is to collect a package of excerpts into an XHTML file. We have seen that accessing the bibliography and glossary to fetch entries is not too much of a burden on most readers. Maybe that is because some readers are caching those files during a single session with the book. So if we collected all of the elided cross-reference outtakes into a single file, then we could access that file (along with the bibliography and glossary) to fetch them. However, that could prove to be a drain on most machines.
- Alternatively, we could gather elided copies of every unique object that is referenced into a large JSON structure that contains each of the excerpts, and a function that takes an ID and returns an excerpt. The scripts could then access any object, by its TDO-wide ID. (URLs? We don't need no steenkin' URLs; we got XML IDs.)

The exact rules by which we elide content, strip attributes and white space, and will affect the size of the package of copies. Some attributes, like @class, @href, @data-audience need to be kept so that elements maintains their appearance. @rel/rev, @epub:type and other data-\* attributes can be safely stripped for brevity. (Murray/Jirka should enumerate a list of safe-to-strip attributes.) We'll have to decide what to do about footnotes within the target elements—strip them out or leave them in? Good reasons on either side.

Figuring out the location of the link is being hovered is relatively easy. Figuring out whether the page layout is one or two columns turns out to be quite a bit harder, and different readers answer CSS questions differently. So, it seems that Ben has figured out how to do wide sidebars that are centered on the screen, and position above or below the equator in opposition to the position of the link being hovered. I have a script that positions in four quadrants in two column mode, and above/below in two column mode, but only on iBooks. We need to find a universal algorithm, or define a suite of algorithms that we can use in specific detectable circumstances.

We recognize that this feature may only work in our core readers: iBooks, ADE, Calibre, Radium. Next on the list are Azardi and ePubReader. After that, we cannot expect to run this feature on less than a laptop or without a mouse. We need better mouse detection in core.js. The ePubReadingSystem variables might also prove useful.

## TDO Toggle

While we were testing our multi-disciplinary text, we used a script to be able to toggle one or more disciplines on or off. It was a handy tool for editing, but it wasn't something that we wanted to include in the current editions because it still exhibits some bugs in different readers, and because we can easily create static editions with any given set of discipline-specific endnotes and paragraphs.

## Build Process

The initial DocBook5 profiling stage of the build process relies upon values specified in the build configuration files and those specified on attributes within the source files. In this phase, any content whose `@audience` and `@userlevel` values do not match the profile established in the configuration file are stripped out, leaving a single document file. Vendor-specific content could be profiled to provide custom content for specific schools or publishers.

### Example 101. Configuration for Professional Edition Epub3 with JavaScript

This configuration is intended to produce an ePub3 archive that is ready to use in a JavaScript-enabled reading environment. We place footnotes at the end of sections. **Wanted Features:** A script will enable hypertext cues for footnotes to ease the burden of traversing a link; we will provide a pop-up with elided text of the target book content.

```
<!-- The Professional Edition -->
<group xml:id="Professional"
  name="Professional Edition"
  os="jvm"
  features="all"
  endnotes="section"
  source="TDO-PRO.xml"
  epub="TDO-PRO.epub"
  pdf="TDO-PRO.pdf"
  stylesheets="core.css          quiz.css tdo-nav.css tdo-toggle.css tdo-viz.css "
  javascripts="jquery-1.10.2.min.js d3.js quiz.js  tdo-nav.js  tdo-toggle.js tdo-toggle-config.js tdo-viz.js "
  ref=" professional
  core
  lis archives museums
  computing ia web
  law bus
  cogsci ling phil"/>
```

### Example 102. Configuration for Professional Edition Epub3 without JavaScript

This configuration is intended to produce an ePub3 archive that is ready to transform into an archive that is suitable for use on Kindle and Kobo devices or software readers. **Wanted Features:** Since these platforms lack JavaScript, we do not include interactive features. We place footnotes after the paragraph in which they are first referenced.

```
<group xml:id="Professional-LCD"
  name="Professional Edition"
  os="Kindle Kobo"
  features="none"
  endnotes="paragraph"
  source="TDO-PRO.xml"
  epub="TDO-PRO-LCD.epub"
  stylesheets="core.css TDO-PRO.css"
  javascripts=""
```



```
ref=" professional
core lis archives museums law
bus computing ia web
cogsci ling phil"/>
```

### Example 103. Configuration for Professional Edition Epub3 pre-PDF

This configuration is intended to produce an ePub3 archive that is ready to transform into a PDF manifestation of the work. We place footnotes at the end of the chapter in which they are contained.

```
<group xml:id="Professional-PDF"
  name="Professional Edition"
  os="PDFreader"
  features="none"
  endnotes="chapter"
  source="TDO-PRO.xml"
  epub="TDO-PRO-PDF.epub"
  stylesheets="core.css TDO-PRO.css"
  javascripts=""
  ref=" professional
core lis archives museums law
bus computing ia web
cogsci ling phil"/>
```

### Example 104. Configuration for an Academic Edition without toggle feature

```
<!-- Memory Institutions -->
<group xml:id="Memory"
  name="Academic Edition: Memory Institutions"
  os="jvm"
  features="nav quiz"
  endnotes="section"
  source="TDO-Memory.xml"
  epub="TDO-Memory.epub"
  pdf="TDO-Memory.pdf"
  stylesheets="core.css          quiz.css tdo-nav.css"
  javascripts="jquery-1.10.2.min.js quiz.js tdo-nav.js"
  ref=" graduate librarian archivist curator
      core
      lis museums archives"/>
```

### Example 105. Configuration for the Core Concepts Edition

```
<!-- The Undergraduate Edition -->
<group xml:id="Undergraduate"
  name="Core Concepts Edition"
  os="jvm"
  features="nav quiz"
  endnotes="none"
  source="TDO-LITE.xml"
  epub="TDO-LITE.epub"
  pdf="TDO-LITE.pdf"
  stylesheets="core.css          quiz.css tdo-nav.css"
  javascripts="jquery-1.10.2.min.js quiz.js tdo-nav.js"
  ref=" undergrad
      core"/>
```

**Profiling Attribute Values**

DocBook5 offers an array of attributes upon which content can be profiled.

**Example 106. @condition**

DocBook5 @condition attribute is set on elements to distinguish those intended for specific media types.

epub3  
mobi  
pdf  
print

**Example 107. @audience**

DocBook5 @audience attribute is set for every footnote in this work, according to its discipline, or multiple disciplines in some cases. Many paragraphs are also tagged according to discipline.

LIS  
Archives  
Museums  
Computing  
IA  
Web  
Linguistics  
CogSci  
Philosophy  
Business  
Law  
Markup\*  
Production\*  
Publishing\*

**Example 108. @userlevel**

DocBook5 @userlevel attribute is set for edition-specific metadata, such as the subtitle and ISBN, some preface material, and an assortment of academic sidebars. The user level represent the reader's academic level and/or vocational arena.

These types are used to distinguish content according to academic level.

Professional  
Professor  
Instructor  
Graduate  
Undergraduate  
Novice

DocBook5 @userlevel attribute is set to distinguish content according to vocation.

Polymath  
Librarian  
Archivist  
Curator  
Programmer  
Webbie  
Architect  
Linguist  
Philosopher  
Psychologist

DocBook5 @userlevel attribute is set to distinguish content for the editorial team.

Editor  
Marker  
Producer  
Publisher

