

Writing Documentation Using DocBook

Using DocBook at CERN

by Michel Goossens

Writing Documentation Using DocBook

by Michel Goossens

Copyright © 1997-2000, , 2001-2002 David Ruge, Mark Galassi, Eric Bischoff, Michel Goossens

This document is a tutorial on how to write documentation in DocBook, and generate HTML and PDF from the XML sources. The description of the DTD elements is general in nature, whereas some tools and the location of system files relate to the CERN XML set-up.

This material is based in part on "A crash-course in DocBook", which was itself created from a fusion of three documents:

- the "Introduction to DocBook" by Mark Galassi
- the "KDE crash-course to DocBook" by David Ruge
- parts of Eric Bischoff's tutorial about DocBook

I have freely adapted the above material to CERN, and have complemented it in many places. On the other hand, for reasons of maintainability, I have completely eliminated all references to SGML and DSSSL, since at CERN I only want to maintain the XSLT stylesheets to render DocBook source material.

This document can be freely redistributed according to the terms of the GNU General Public License.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Table of Contents

1. XML and DocBook	1
1.1. DocBook	1
1.1.1. DocBook's past and future	1
1.1.2. Who uses DocBook?	1
1.1.3. Why DocBook?	2
1.2. A few words about XSLT	3
1.3. Presentation of the DocBook XML Tools installed at CERN	3
1.4. My First DocBook File	4
1.5. Translation to HTML	5
1.6. What about those Style Sheets?	6
1.7. Translation to PDF	7
1.8. Anatomy of a DocBook Tag	8
1.9. The Structure of a DocBook File	9
1.9.1. Structure of a Book (or chapter)	9
1.9.2. Structure of an Article	9
1.9.3. Book elements step by step	9
1.10. The Document Type Declaration	10
1.10.1. Internal general entities	10
1.10.2. General external entities	12
2. Meta Information	15
3. Lists	17
3.1. The simplelist	17
3.2. The itemizedlist	17
3.3. The orderedlist	18
3.4. The variablelist	19
3.5. The segmentedlist	19
3.6. qandaset	20
3.7. Procedures	21
4. Tables	23
4.1. A simple table	23
4.2. An informal table	24
4.3. A more complex table with some math	25
5. Graphics	27
6. Links	29
7. Describing the Application's Interface	31
7.1. Examples	31
7.2. GUI Interface Elements	32
7.3. Command Line Elements	33
7.4. Elements for classes and methods	36
7.5. Describing an API	38
8. Bibliography elements	39
8.1. Handling	40
9. Miscellaneous	45
9.1. Labelling Tags	45
9.2. Formatting Tags	45
9.3. Warnings, Tips, and Notes	45
10. If you want to know more	47
A. Emacs PSGML mode tips	49
B. A complete example of a DocBook article	51
Glossary	61
C. Overview of all DocBook elements	65
Index	79

List of Figures

1.1. PDF rendering of a small DocBook test file	7
8.1. Citations and bibliography (PDF output)	43
8.2. Citations and bibliography (Displayed by the Mozilla browser)	43
A.1. Emacs and its psgml mode with the DocBook DTD	49
B.1. DocBook example with MathML translated into PDF	58
B.2. DocBook example with MathML translated into HTML and viewed with Mozilla (one file) .	59
B.3. DocBook example with MathML translated into HTML and viewed with Mozilla (chunked file)	
60	

List of Tables

4.1. Number of Visitors 24

4.2. Physical and mathematical constants 25

List of Examples

1.1. My first simple DocBook file	4
1.2. Running docbook2html (single output file)	5
1.3. HTML version generated from a DocBook file as displayed by lynx	5
1.4. Running docbook2html with the chunk option	5
1.5. A small DocBook file showing the use of attributes	6
1.6. Generating PDF with docbook2pdf	7
1.7. Chapters and sections	9
1.8. Entities used to share text	11
1.9. HTML version generated from a DocBook file as displayed by Lynx	11
1.10. Including external files using entities	12
2.1. DocBook metainformation	15
3.1. A simplelist	17
3.2. An itemizedlist	18
3.3. An orderedlist	18
3.4. A variablelist	19
3.5. A segmentedlist	20
3.6. A qandaset	20
3.7. A procedure list	21
4.1. A regular simple table	23
4.2. A simple informal table (no title)	24
4.3. A more complex table	25
5.1. An inline media object	27
5.2. A screenshot	27
6.1. Five kinds of links	29
7.1. An example	31
7.2. A BASIC Example	31
7.3. Displaying markup in a CDATA section	31
7.4. CDATA sections and markup	31
7.5. guimenu and shortcut	32
7.6. A command and its output	34
7.7. A commandsynopsis	35
7.8. Class synopsis	36
7.9. Describing a function in a C library API	38
8.1. Example of A bibliography and citations	41

Chapter 1. XML and DocBook

The aim of this chapter is to give the reader an introduction to XML and one of its best-known vocabularies, DocBook.

1.1. DocBook

The DocBook markup system is described in the printed book *DocBook - The Definitive Guide* [TDG1999]. There is an up-to-date local HTML copy at CERN [TDG2002]. For maximal efficiency, this tutorial must thus be used along with the above reference documentation. Indeed, by far not all of the (almost) 400 DocBook elements are covered. Hence, the complete list, as well as to all available attributes and entity sets, are explained in that reference document. This present tutorial mainly limits itself to the principles of the DocBook markup language. With the help of examples we introduce you to the use of the rich DocBook element tag set.

There are two DocBook-related discussion lists, one about Docbook itself [DBLIST] and one about its applications [DBAPPSLIST].

1.1.1. DocBook's past and future

DocBook is already more than ten years old. It began in 1991 as a joint project of HaL Computer Systems and O'Reilly. Its popularity grew, and in 1994, the Davenport Group became an officially chartered entity responsible for DocBook's maintenance. DocBook V1.2.2 was published at that time. In mid-1998, it became a Technical Committee (TC) of the Organization for the Advancement of Structured Information Standards (OASIS).

DocBook V3.1 was released in February 1999, and was, like its previous versions SGML only. In February 2001, when DocBook V4.1 and DocBook XML V4.1.2 became an OASIS Standard support SGML and XML is now available on an equal basis. More details are on the OASIS DocBook history Web page [<http://www.oasis-open.org/docbook/intro.shtml>]. The current V4.2 DTD versions for both XML and SGML were released in July 2002. Version 5 is foreseen to become available at the beginning of 2003.

The OASIS TC observes a very cautious policy regarding changes to the DTD. Backward-incompatible changes can only be introduced in major releases (4.0, 5.0, 6.0, and so on), and only if the change was described in comments in the DTD in the previous major release. In particular, the next version of DocBook, 5.0, will be XML compliant, and this will introduce a lot of changes.

1.1.2. Who uses DocBook?

DocBook has a very rich element set, is freely available, well documented, and comes with a good set of production tools, DocBook has become wide spread and has been adopted by many producers of software (open source as well as commercial), by students, and faculty for courses and theses, and by document managers everywhere. The following is an non-exhaustive list of who uses DocBook at present.

- Many books at O'Reilly, the publishing house, are marked up in DocBook (in particular [DocBook, the definitive Guide]).
- A lot of computer documentation, such as that for the following projects: xfree86, GNOME, KDE, FreeBSD, PHP, the Linux Documentation Project.
- Similarly, a lot of the documentation for commercial software and hardware vendors. For instance Sun, Red Hat, SuSE, HP, Cogent Real-Time Systems, Conectiva, Rational, Mandrakesoft, Caldera, Apple (Darwin docs), etc. use DocBook.

- Producing generated documentation from code comments (GNOME, Linux kernel, XSLT Standard Library).
- For training material, where from a single document, one can produce presentation slides, sample files, and printed handouts.
- Whitepapers, like system or formal specifications (e.g., the RELAX NG specification) and proposals.
- For maintaining websites, in particular those hosting FAQs.
- Producing presentation slides, courseware, theses and dissertations.
- Using a single docbook source with various stylesheets to document applications in various ways: Online manuals (PDF, HTML), Context-sensitive help (HTML, HTML Help), Man pages and formatted text (using also non-XSLT HTML to text conversion), Filtering conditionalized versions (using profiling).
- The complete development chain of a product, including a description of the product itself, the automated test tool documentation, the defect tracking database, related software (O/S, networking, Apache, SQL, etc.).
- Single sourcing to ensure consistency, by generating three targets from the same DocBook document: product API specifications (targeted at internal developers), API reference manual (targeted at our customers), API validation code (different programming languages).

1.1.3. Why DocBook?

The DocBook markup language, maintained by the OASIS consortium, is specifically suited for technical documentation. It provides a rich set of tags to describe the content of especially software documentation.

A number of key points that help understand what DocBook is follow.

Docbook is a markup language.

It is very similar to HTML in this respect. The tags give some structure to your document, and appear intermixed with the informational text.

This peculiar point makes it quite different from DTP (Desktop Publishing Tools) that spend most of their time "making the text look nice". In the case of DocBook and its associated XML tools, the rendering is done indirectly by using a transformation stage to generate the format for a given output medium.

DocBook was mainly developed for technical documentation.

DocBook is perfectly suited for car engine parts documentation. However, it is strongly biased towards computer programs documentation.

DocBook is maintained by an independent consortium.

The OASIS [<http://www.oasis-open.org>] consortium is in charge of maintaining and making this standard evolve through the DocBook Technical Committee. This is a guarantee of independence with respect to proprietary software and standards.

Major players of the industry like Boeing or IBM are members of OASIS. A more complete members list [<http://www.oasis-open.org/html/members.htm>] is at the OASIS site.

Technically, DocBook is a (SGML or) XML DTD.

This means that one can take profit of the many (SGML and) XML aware tools that are (often freely) available. While DocBook as an XML implementation is quite recent, it has a long history as a SGML application.

DocBook is not a presentation language.

DocBook carefully cares about *not* specifying how the final documentation looks like. This allows the writer to concentrate on the organization and meaning of the document being written. All the presentation issues are devolved to style sheets.

This ensures all your documents have a consistent appearance, whoever is the technical writer.

DocBook is customizable.

It is quite easy to customize the DTD to meet user need thanks to its modular organization. However, one must be aware that customization must respect SGML/XML conventions and that it can lead to incompatibilities.

If DocBook is used in conjunction with Norman Walsh's modular XSLT stylesheets, it is also possible to customize the way a DocBook file can be printed or put online.

DocBook is comprehensive.

The large number of tags defined in DocBook guarantees that it can accomodate a wide range of situations and of processing expectations.

This in turn makes it a bit difficult to learn, but one can manage writing documentation knowing only a limited set of tags and referring to the reference documentation when needed.

DocBook uses long and understable tags.

Example of such tags are `<itemizedlist>` or `<literallayout>`. This makes DocBook source text easier to read than an HTML source, for example. As a drawback, it can become quite tedious to type those long tags, but in that case specialized modes in editors, such as Emacs' psgml mode, can alleviate this burden. They also exist some complete DocBook authoring tools, but these are mostly not free.

1.2. A few words about XSLT

In Section 1.3 we introduce the `docbook2pdf` and `docbook2html` applications. They allow you, with the help of Norman Walsh' XSLT stylesheets, to transform DocBook XML sources into PDF and HTML, respectively. The XSLT language, that is used to transform the DocBook XML sources into HTML, XSLT-FO, etc. is described in the W3C recommendation *XSL Transformations [REC-XSLT]*. There is a nice XSLT tutorial [NICXSLTUT], as well as Mike Kay's book *XSLT Programmer's Reference* [KAY2001]. The style sheets themselves are also described [DBXSLTDOC].

1.3. Presentation of the DocBook XML Tools installed at CERN

This section describes how to work with DocBook on Unix-like systems, such as Linux and Solaris (although the tools can also be made to work easily on Nice 2000).

The DocBook XML tools consist of several packages that work together to edit DocBook XML files, to convert them into many other formats, and to perform other miscellaneous operations. The output formats include:

- HTML (and its XML variant XHTML)
- HELP
- JavaHelp

- FO (for generating PDF)
- HTMLHelp

At CERN the following two DocBook XML tools are installed.

- docbook2html - transforms a DocBook source into one or more HBOOK files.
- docbookpdf - transforms a DocBook source into a PDF file.

For Linux they are available in the `/afs/cern.ch/sw/XML/XMLBIN/bin/i386-linux` directory, which should be added into your PATH environment variable.

1.4. My First DocBook File

First, you need a DocBook file to convert. Using your favorite text editor you can type (or cut and paste) the following lines:

Example 1.1. My first simple DocBook file

```

1 <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN">
  <article>

    <articleinfo>
5 <title>XML rules the world!</title>
    </articleinfo>

    <section>
      <title>XML and DocBook do it together</title>
10
      <para>DocBook and XML are the golden twins
        of the 21st century !</para>

    </section>
15 </article>

```

Then save it, say, with the name `dbfile.xml`.

This file has to be well-formed. To check an XML document, you can use the following tool:

```
$rxp -s dbfile.xml
```

If you have catalog support the above source, in fact, can relate your XML source file to a Document Type Definition (DTD) for DocBook. A DTD defines the possible elements, their arguments, and their content models (which elements can contain which, how do elements relate to each other). If you have access to the relevant DTD, you can validate your file. At CERN, on Unix (with AFS) the DocBook DTD can also be referenced by adding the second line specifying its complete address, as follows:

```

<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
  "/afs/cern.ch/sw/XML/XMLBIN/share/www.oasis-open.org/docbook/xmltd-4.2/docbookx.dtd"
>

<article>
  .
  .
  .
</article>

```

To validate dbfile.xml, use the command:

```
$rxp -s -V dbfile.xml
```

If errors are shown, read through the error log and correct errors from the beginning of the list first. Often, an early error has knock-on effects further down the processing of the document.

Alternatively, when using emacs there exists an XML processing mode, psgml, see Appendix A “Emacs PSGML mode tips”.

1.5. Translation to HTML

Your file is now ready to be converted to other formats. To convert this file from DocBook format to HTML format, the command docbook2html can be used. This will run Norman Walsh’s XSLT to HTML stylesheets and Michael Kay’s saxon XSLT processor to generate the HTML files from the DocBook sources.

Example 1.2. Running docbook2html (single output file)

```
$docbook2html dbfile.xml
dbfile.xml abc.xsl dbfile.html
```

Using java 1.4 from AFS.

```
java version "1.4.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0-b92)
Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)
```

If you validated your XML file with an XML parser, then there should be no errors, and the procedure will generate the file dbfile.html, which can be viewed with an HTML browser, e.g., the line browser lynx shows the following.

Example 1.3. HTML version generated from a DocBook file as displayed by lynx

XML rules the world!

XML rules the world!

Table of Contents

XML and DocBook do it together

XML and DocBook do it together

DocBook and XML are the golden twins of the 21st century !

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.

Arrow keys: Up and Down to move. Right to follow a link; Left to go back.

H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

In general, with a larger document, and the chunk option, you will get several files, as shown when we run (an early version) of the present manual with the docbook2html procedure.

Example 1.4. Running docbook2html with the chunk option

```
$docbook2html -chunk dbatcern.xml
dbatcern.xml abc.xsl index.html
```

Using java 1.4 from AFS.

```
java version "1.4.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0-b92)
Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)
```

```
Writing ch01s02.html for sect1(why-docbook)
Writing ch01s03.html for sect1(your-world-view)
Writing ch01s04.html for sect1(markup-based-on-content)
Writing ch01.html for chapter(introduction)
Writing ch02s02.html for sect1(installing-the-tools)
Writing ch02s03.html for sect1(hello-world)
Writing ch02s04.html for sect1(style-sheets)
```

... Many more lines ...

```
Writing ch11s03.html for sect1(warnings)
Writing ch11.html for chapter(miscellaneous)
Writing ch12.html for chapter(next)
Writing apa.html for appendix(emacs)
Writing go01.html for glossary(glossary)
Writing index.html for book(dbatcern)
```

We can see that files are generated at the level of the section (`sect1` element), and in each case the value of the `id` attribute, if present, is displayed between parentheses.

1.6. What about those Style Sheets?

XML processors, and in particular XSLT or CSS stylesheets can tune the way they handle or display elements in clever and convenient ways. For instance, we can add a `lang` attribute on the `<book>` and an `id` attribute on the `<chapter>` tags in the previous example and get the following:

Example 1.5. A small DocBook file showing the use of attributes

```
1 <!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
   "/afs/cern.ch/sw/XML/XMLBIN/share/www.oasis-open.org/docbook/xmltd-4.2/docbookx.dtd"
>

5 <book lang="en-gb">
  <!-- This is a <huge>comment & -->

  <bookinfo>
    <title>XML rules the world!</title>
10 </bookinfo>

    <chapter id="Chap1">
      <title>XML rules the world!</title>

15 <para>Here we go with DocBook and XML.</para>

    </chapter>
  </book>
```

This example shows how to place comments (line 5) in an XML source file. A comment is all material between a leading `<!--` and a trailing `-->`. The textual material between these two markers cannot contain two consecutive hyphens, i.e., `--` is disallowed inside comments. A comment can span several lines. Comments are not interpreted by XML parsers, i.e. as seen in the example, you can put markup and other forbidden characters inside comments without problems.

To convert a DocBook file from DocBook format to another format, you need an XML processing tools and an XSL style sheet. The command `docbook2html` uses internally Mike Kay's saxon XSLT interpreter and Norman Wlsh' XSLT style sheets.

1.7. Translation to PDF

Instead of generating an HTML hyper-document it is sometimes necessary to produce a nicely printable copy. Therefore, starting from the same DocBook XML source document and using an alternative XSLT style sheet one can generate XSL-FO formatting objects, that can be translated into various media formats, in particular PDF with one of several XSL-FO to PDF translators.

At CERN a script, `docbook2pdf`, hides the details of this translation process from the user. By default the script takes one argument, the name of the XML source to be transformed into PDF, for example,

Example 1.6. Generating PDF with `docbook2pdf`

```
$docbookpdf dbfile.xml
docbook2pdf dbfile.xml
dbfile.xml abc.xsl dbfile.fo
```

Using java 1.4 from AFS.

```
java version "1.4.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0-b92)
Java HotSpot(TM) Client VM (build 1.4.0-b92, mixed mode)
```

Making portrait pages on A4 paper (210mmx297mm)

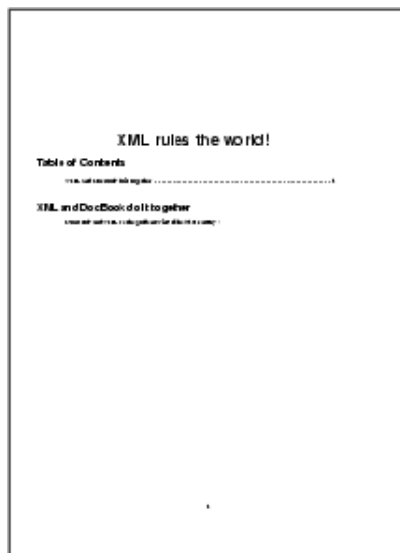
```
running pdftex/passivetex (twice): pdfxmltex dbfile.fo
```

```
This is pdfTeX, Version 3.14159-pre-1.0-unofficial-20010704 (Web2C 7.3.3.1)
(./dbfile.fo[afs/cern.ch/sw/TeXlive/tl7/texmf-var/pdftex/config/pdftex.cfg]
LaTeX2e <2001/06/01>
```

```
...
Output written on dbfile.pdf (1 page, 7710 bytes).
Transcript written on dbfile.log.
```

```
This is pdfTeX, Version 3.14159-pre-1.0-unofficial-20010704 (Web2C 7.3.3.1)
(./dbfile.fo[afs/cern.ch/sw/TeXlive/tl7/texmf-var/pdftex/config/pdftex.cfg]
...
Output written on dbfile.pdf (1 page, 7710 bytes).
Transcript written on dbfile.log.
```

In the above we see that first a XSLT translator (saxon) is run that generates XSL-FO elements, that are then rendered into PDF by `pdftex`, a variant of the TeX program that generates PDF. Thus we obtain a file `dbfile.pdf`, which can be seen in Figure 1.1.

Figure 1.1. PDF rendering of a small DocBook test file

The file `dbfile.pdf` can be viewed with `ghostview` or `acroread`, and can be sent by these applications to a PostScript printer.

What we obtained with the previous example is the default presentation as defined by the style sheets of Norman Walsh. In fact, Norman parameterized his style sheets in a convenient way, so that over one hundred parameters are available to finetune the format of the output page. At CERN you can specify a second filename to the `docbook2pdf` command, and when specified, it will be used as an XSLT customization style sheet, for instance, at CERN you could type:

```
$docbookpdf dbfile.xml dbforpassivetex.xsl
```

If you are interested in such customization, you can start from the style sheet `dbforpassivetex.xsl` that is available from the docbook DocBook Web page at CERN [<http://xml.cern.ch/docbook.html>]. A full discussion of all parameters can be found in the [reference guide] entry in the [DocBook XSLT stylesheets] section of the above Web page.

1.8. Anatomy of a DocBook Tag

XML, just like its predecessor SGML, is a generic markup language. It is a meta-language syntax to define a convenient vocabulary for the application domain being dealt with. In science, in particular, software, DocBook offers a large choice of elements that are useful for marking up and producing good and precise documents in computer science, etc.

However, DocBook being an instance of an XML language, it obeys the XML syntax, which introduces elements, attributes, and entities as main components. For example, the XML start tag `<chapter id="introduction">` has the element name `chapter` and the attribute `id`. The element influences the rendering of the text inside the element, while an element's attributes can further modify the behaviour or meta-information associated with the element. An element is closed with an XML end tag, for instance `</chapter>`. In the case of the `chapter` element the element contents, i.e., all text included within the starting and closing tags should be treated as a chapter, while the `id` attribute labels the chapter so it can be linked to or used as a file name when DocBook is translated to another format.

An element can also be *empty*, i.e., have no contents. For such an element we use a specific syntax, e.g. for a `xref` element, `<xref linkend="foo" />`.

Most DocBook tags contain a common set of attributes. The most often-used such common attributes are `lang`, which specifies the language of the data inside the element, `id` labels an element so that it can be referenced by other elements, and `role`, which allows one to subclass an element, to make it information-wise more specific.

As an XML language instance all DocBook elements must have a start and an end tag, or be empty (please forget your SGML or HTML habits, where end-tags are not always required). You should also make sure to use the proper case for element names, entities, and attributes. Detailed information about all element names and their attributes is in the DocBook reference guide.

1.9. The Structure of a DocBook File

DocBook document always has a top level element. In most cases it will be `book`, or `article`, but it can also be `set`, `part`, `chapter` and a few more.

The structural content of these two most-occurring cases for top (more correctly called root) elements of `book`, or `article` (or perhaps `chapter` when you are contributing to a book) are detailed next.

1.9.1. Structure of a Book (or chapter)

A book is structured in the following way:

```
book
  meta information
  chapter
    sect1
    sect1
  chapter
    sect1
  appendix
    sect1
  appendix
    sect1
  ...
  glossary
```

1.9.2. Structure of an Article

An article has a somewhat simpler structure than a book:

```
article
  meta information
  sect1
  sect1
    sect2
  sect1
  ...
```

1.9.3. Book elements step by step

Example 1.7. Chapters and sections

```
1 <!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
   "/afs/cern.ch/sw/XML/XMLBIN/share/www.oasis-open.org/docbook/xml/dtd-
4.2/docbookx.dtd"
   >
   <book id="hello-world" lang="en">
5
```

```

<bookinfo>
<title>Hello, world</title>
</bookinfo>

10 <chapter id="introduction">
    <title>Introduction</title>

    <para>This is the introduction. It has two sections</para>

15 <sect1 id="about-this-book">
    <title>About this book</title>

    <para>This is my first DocBook file.</para>

20 </sect1>

    <sect1 id="work-in-progress">
    <title>Warning</title>

25 <para>This is still under construction.</para>

    </sect1>

    </chapter>
30 </book>

```

Example 1.7 shows a skeleton of the structural tags. Lines 1 and 2 contain the document type declaration (via a public identifier on line 1, which needs a catalog to operate and via an explicit system path on line 2, which, of course, will only work at CERN when connected to AFS). This information is described more fully in Section 1.10.

Next comes the root element (line 3), which contains the complete document. The name of the root element must be identical to the element name following the DOCTYPE specifier on line 1. Note the use of the lang attribute inside the <book> start tag. It is good practice to always use a language attribute to clearly indicate the (main) language in which the document is written.

After the <book> tag comes the meta information for the document, which is enclosed in the <bookinfo> element (lines 5-7). This information is described in more detail in Chapter 2.

Then come the chapters of the book (lines 9 to 28), which can contain one or more section tags (<sect1> through <sect5>). It is important to associate a label to each of the structural elements of your document using the id attribute for <chapter>, <sect>, etc. This makes cross-references between structural elements possible and it allows the output processors to use the value of the id attribute to name the generated output files. In order to make the development and maintenance of your documents easier, it is advisable to assign meaningful values (e.g., not merely numbers) to these id attributes.

Structural elements, such as chapters and sections, must contain at least a <title> (lines 10, 15, 22), and a (possibly empty) <para> element (lines 12, 17, 24).

The content model for the various elements, i.e., where and how many times each one can occur at a given point in a document is defined by the DocBook DTD (or Schema).

Content in DocBook is contained within a <para> tag, which is very similar to the <p> tag in HTML and LinuxDoc except that it must always have a closing </para> tag. Each time there can be a line break in some text (like in a list item), it means that the text will have to be enclosed in <para> tags.

1.10. The Document Type Declaration

In non-trivial documents the document type declaration (the very first lines of a DocBook file) can be slightly more complicated than the one or two lines presented in the document skeleton previously.

1.10.1. Internal general entities

The following example shows how entities allow one to define abbreviations for text strings. They can also be used to declare a convenient name for a character that is unavailable on a standard keyboard.

Example 1.8. Entities used to share text

```

1 <?xml version="1.0" encoding="iso-8859-1" ?>
  <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
    <!ENTITY CERN "Conseil Européen pour la Recherche Nucléaire">
    <!ENTITY WWW "<application>World Wide Web</productname>">
5  <!ENTITY W3C "&WWW; Consortium">
  ]>

  <article id="The.Web" lang="en">
    <articleinfo>
10 <title>The origin of the &WWW;</title>
    </articleinfo>

    <section id="introduction">
      <title>General Introduction</title>
15
      <para>
        Tim Berners-Lee and collaborators invented the &WWW; at
        <acronym>CERN</acronym> (&CERN;) at the beginning of the
        nineteen nineties.
20 </para>

      <para>
        Today all further developments are coordinated by the &W3C;
        (MIT, INRIA, Keio).
25 </para>

    </section>
  </article>

```

Several important points should be noted in Example 1.8. Firstly, line 1 shows that we have to declare an explicit encoding when we want to use non-ASCII characters (more precisely, characters that are not encoded in Unicode's UTF8 or UTF16). In particular, on line 3, where we define the CERN entity, we make use of the é (accented e) which is part of the ISO-8859-1 encoding, as specified on line 1. On the other hand we could use a Unicode character reference and replace line 3 by the following statement.

```
<!ENTITY CERN "Conseil Europ&#xe9;en pour la Recherche Nucl&#xe9;aire">
```

In this case we do not have to declare an explicit encoding. Further, line 4 defines the WWW application, showing that one can use markup inside entity definitions, while the entity definition of W3C indicates that one can use entity references to previously declared entities (&WWW; in this case).

In the body of the text the defined entities are referenced on lines 10, 18, 19, and 23.

When you run the above file `dbw3cexa.xml` through the `docbook2html` procedure you obtain the file `dbw3cexa.html`, which is shown, as displayed by the Lynx line browser in Example 1.9.

Example 1.9. HTML version generated from a DocBook file as displayed by Lynx

```

                                The origin of the World Wide Web

The origin of the World Wide Web
-----

Table of Contents

```

General Introduction

General Introduction

Tim Berners-Lee and collaborators invented the World Wide Web at CERN (Conseil Européen de la Recherche Nucléaire) at the beginning of the nineteen nineties.

Today all further developments are coordinated by the World Wide Web Consortium (MIT, INRIA, Keio).

Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<-' to go back.
 Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
 H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list

The consistent use of entities has a number of advantages:

- you do not have to type several times a possibly quite long identical text string;
- it allows you to centralize changes to commonly-used phrases and terms in a single place, thus guaranteeing consistency;
- if the entity name is well chosen, it makes the source documents easier to read and maintain.

1.10.2. General external entities

For larger documents, it is convenient to keep chapters or other large text chunks in individual files that can be maintained separately. In such a case the various files that have to be included in the master document are declared in the document type declaration as general external entities.

For instance, the present tutorial document is split at the chapter and appendix level into separate files, that are included with a single main driver file, as follows (only part of the file is shown):

Example 1.10. Including external files using entities

```

1 <!DOCTYPE book SYSTEM
   "/afs/cern.ch/sw/XML/XMLBIN/share/www.oasis-open.org/docbook/xmltd-4.2/docbookx.dtd"

[
  <!ENTITY bookinfo SYSTEM "bookinfo.xml">
5  <!ENTITY introduction SYSTEM "introduction.xml">
   ....
  <!ENTITY emacs SYSTEM "emacs.xml">
  <!ENTITY examples SYSTEM "examples.xml">
  <!ENTITY glossary SYSTEM "glossary.xml">
10 ]>
   <book id="dbatcern" lang="en">
     &bookinfo;
     &introduction;
     ...
15 &emacs;
    &examples;
    &glossary;
   </book>

```

There exists a one-to-one correspondence between the entity definition using the `<!ENTITY ...>` declaration and the entity references, using the `&xxx;` syntax, where `xxx` is an entity that corresponds to a file that has to exist on the system as defined by the entity declaring `xxx`. A reference to a non-existing file will result in an error. For instance line 3 declares the `bookinfo` entity as an external reference to the file `bookinfo.xml`, that will be included on by the entity reference `&bookinfo;` on line 13.

Entity names have to be unique throughout the document, but of course entities can be references several times (e.g., `&WWW;` on lines 10 and 17 of Example 1.9). For internal entities multiple references could be useful in the case of a copyright or other company message that has to be repeated on every page or at the beginning of each chapter.

Chapter 2. Meta Information

The tags covered in this section provide meta information about the document, such as author, title, data of publication, etc.

- `abstract` - Document summary
- `articleinfo` - Metainformation for an article
- `author` - Author of a document
- `authorgroup` - Wrapper for author information
- `authorinitials` - Initials or other identifier for the author of a revision or comment
- `bookinfo` - Metainformation for a book
- `date` - Date of publication or revision of a document
- `firstname` - Given name
- `keyword` - Term describing the content of a document
- `keywordset` - Set of terms describing the content of a document
- `othername` - Name component that is not a firstname, surname, or lineage
- `releaseInfo` - Information about a particular version of a document
- `revhistory` - Revisions to a document
- `revision` - Entry in revhistory, describing some revision made to the text
- `revnumber` - Number of a revision
- `revremark` - Description of a revision
- `surname` - Family name
- `title` - Text of a heading or the title of a block-oriented element

Example 2.1. DocBook metainformation

```

<bookinfo>
<title>The TZirrdle Handbook</title>
<authorgroup>
<author>
<firstname>George</firstname>
<othername>N.</othername>
<surname>Ugnacious</surname>
</author>
</authorgroup>

<date>03/04/1999</date>
<releaseinfo>1.01.00</releaseinfo>

<abstract>
<para>
<application>twiddle</application> is an application specially
designed to do nothing you would ever want.
</para>
</abstract>

<keywordset>
<keyword>twiddle</keyword>
<keyword>sample application</keyword>
</keywordset>

</bookinfo>

```

The `<bookinfo>` tag contains all of the meta information for your document, as opposed as its contents that are in the various chapters. This allows you, for instance, to:

- conveniently retrieve a specific document among many others on your system;
- print some information on the front page, under the control of the style sheets;
- serve it to search engines if you put it online in HTML format.

Chapter 3. Lists

Elements related to DocBook lists are shown below. DocBook lists are very similar to their counterparts in HTML except that DocBook contains several more types of lists for specialized purposes.

- `answer` - An answer to a question posed in a `qandaset`
- `itemizedlist` - List in which each entry is marked with a bullet, dash, or other dingbat
- `listitem` - Wrapper for the elements of items in an `itemizedlist` or `orderedlist`
- `member` - Member of a `simplelist`
- `orderedlist` - List in which each entry is marked with a sequentially incremented label
- `procedure` - List of operations to be performed
- `qandaentry` - A question/answer set within a `qandaset`
- `qandaset` - A question-and-answer set
- `question` - A question in a `qandaset`
- `seg` - Component of a `segmentedlist`
- `seglistitem` - List item in a `segmentedlist`
- `segmentedlist` - List of sets of information
- `segtitle` - Title that pertains to one `seg` in each `seglistitem`
- `simplelist` - List of single words or short phrases
- `step` - Part of a procedure
- `substeps` - Wrapper for steps within steps
- `term` - Hanging term attached to a `listitem` within a `varlistentry` in a `variablelist`
- `variablelist` - List in which each entry is composed of sets of one or more terms with associated `listitems`
- `varlistentry` - Wrapper for term and its associated `listitem` in a `variablelist`

3.1. The simplelist

The easiest of all the lists to use is the simplelist. It is designed for lists of short phrases (like a grocery list) and only requires two tags for building the list as you can see in the example below. The `<member>` tag can only contain inline content, so a simplelist cannot contain other lists.

Example 3.1. A simplelist

```
<simplelist type="inline">
<member>Apples</member>
<member>Oranges</member>
<member>Bananas</member>
<member>Grapefruit</member>
<member>Black Beans</member>
</simplelist>
```

When converted, a simplelist will look something like this: Apples, Oranges, Bananas, Grapefruit, Black Beans

3.2. The itemizedlist

An itemizedlist is similar to the simplelist except that each entry contains a paragraph instead of just a short phrase, allowing you to put more varied content in your list. ItemizedLists can contain other lists.

Example 3.2. An itemizedlist

```
<itemizedlist>
<listitem><para>Apples - my favorite fruit.</para></listitem>
<listitem><para>Oranges - yummy, but sticky.</para></listitem>
<listitem><para>Bananas - they ripen too quickly!</para></listitem>
<listitem><para>Grapefruit - great when eaten in halves.</para></listitem>
<listitem><para>Black Beans - go well with rice.</para></listitem>
</itemizedlist>
```

The example will look something like this when converted:

- Apples - my favorite fruit.
- Oranges - yummy, but sticky.
- Bananas - they ripen too quickly!
- Grapefruit - great when eaten in halves.
- Black Beans - go well with rice.

3.3. The orderedlist

The orderedlist is like the itemizedlist except that each listitem is numbered or lettered. The numeration attribute specifies what kind of numbering will be used and can be one of the following values: arabic, upperalpha, loweralpha, upperroman, lowerroman. There are several other attributes that control the appearance of an orderedlist, see the DocBook Reference for details. orderedlists can contain other lists.

Example 3.3. An orderedlist

```
<orderedlist numeration="arabic">
<listitem><para>Wake up.</para></listitem>
<listitem><para>Eat Breakfast.</para></listitem>
<listitem><para>Take a shower.</para></listitem>
<listitem><para>Contemplate my navel.</para></listitem>
<listitem><para>Go to Sleep.</para></listitem>
</orderedlist>
```

The example will look something like this when converted:

1. Wake up.
2. Eat Breakfast.
3. Take a shower.
4. Contemplate my navel.
5. Go to Sleep.

3.4. The variablelist

The variablelist is similar to an HTML definition list. It is used when you have a list of terms and definitions. The variablelist consists of several tags: <varlistentry>, which is used to group related terms together, <term>, which contains the term, and <listitem>, which contains the description of the term.

Example 3.4. A variablelist

```
<variablelist>
<varlistentry>
<term>Black Beans</term>
<listitem><para>My favorite black bean recipe is black bean
soup, but they also go well with rice.</para></listitem>
</varlistentry>
<varlistentry>
<term>Apples</term>
<term>Bananas</term>
<listitem><para>You can eat them straight, but they also go
well in salads and in desserts.</para></listitem>
</varlistentry>
</variablelist>
```

When converted, the example variablelist will look something like this:

Black Beans

My favorite black bean recipe is black bean soup, but they also go well with rice.

Apples, Bananas

You can eat them straight, but they also go well in salads and in desserts.

3.5. The segmentedlist

segmentedlists are used to list information in distinct fields like the contents of an address book. The name of each field is put inside of a `<segtitle>` tag. Then, use the `<seglistitem>` tag to start and end each set of data. The actual data is put in the `<seg>` tag.

Example 3.5. A segmentedlist

```
<segmentedlist>
<segtitle>Name</segtitle>
<segtitle>Occupation</segtitle>
<segtitle>Favorite Food</segtitle>
<seglistitem>
<seg>Tux</seg>
<seg>Linux mascot</seg>
<seg>Herring</seg>
</seglistitem>
<seglistitem>
<seg>Konqui</seg>
<seg>The KDE Dragon</seg>
<seg>Gnomes</seg>
</seglistitem>
</segmentedlist>
```

This silly example looks something like this when converted:

Name: Tux

Occupation: Linux mascot

Favorite Food: Herring

Name: Konqui

Occupation: The KDE Dragon

Favorite Food: Gnomes

3.6. qandaset

The qandaset is a specialized list designed specifically to deal with sets of questions and answers, like you would see in a FAQ. Each set of questions and answers are contained within a `<qandaentry>` tag. The `<question>` and `<answer>` tags contain the questions and answers respectively.

Example 3.6. A qandaset

```
<qandaset>
<qandaentry>
<question>
<para>What are little boys made of?</para>
</question>
<answer>
<para>Snips and snails and puppy dog tails.</para>
</answer>
</qandaentry>
<qandaentry>
<question>
<para>What are little girls made of?</para>
</question>
<answer>
<para>Sugar and spice and everything nice.</para>
```

```
</answer>
</qandaentry>
</qandaset>
```

The qandaset looks something like this when converted:

3.6.1. What are little boys made of?

Snips and snails and puppy dog tails.

3.6.2. What are little girls made of?

Sugar and spice and everything nice.

3.7. Procedures

Procedure lists are a specialized `orderedlist` used for listing step-by-step procedures like you would find in a recipe or Linux HowTo.

Example 3.7. A procedure list

```
<procedure>
<title>Waking Up</title>
<para>This is what you must do to awaken.</para>
<step performance="required">
<para>
Bring yourself to a hypnopompic state, either from an ongoing dream or by use of
your internal clock. You may feel unable to move, but you will no longer be
dreaming. </para>
<para>Now you are ready for real-world readjustment.</para>
<substeps>
<step performance="optional">
<para>Roll over.</para>
</step>
<step performance="required">
<para>Squint out of one eye.</para>
</step>
</substeps>
</step>
<step performance="required">
<para>Yawn and rise from your bed.
</para>
</step>
</procedure>
```

The above example would look something like this when converted:

Procedure 3.1. Waking Up

This is what you must do to awaken.

1. Bring yourself to a hypnopompic state, either from an ongoing dream or by use of your internal clock. You may feel unable to move, but you will no longer be dreaming.
Now you are ready for real-world readjustment.

- a. Roll over.
 - b. Squint out of one eye.
2. Yawn and rise from your bed.

Chapter 4. Tables

Tables are used to organize data into a columnar representation. Optional titles, headers, and footers can be added to for describing the data, as required. DocBook tables come in two varieties: the `table`, which requires a title, and the `informaltable`, which does not have a title. All the other characteristics of these two table types are the same. DocBook elements associated to table markup are detailed below.

- `entry` - Cell in a table
- `entrytbl` - Subtable appearing as a table cell
- `informaltable` - Untitled table
- `row` - Row in a `tbody`, `thead`, or `tfoot`
- `table` - Table in a document
- `tbody` - Wrapper for the rows of a table or `informaltable`
- `tfoot` - Footer row of a table
- `tgroup` - Wrapper for part of a table that contains an array along with its formatting information
- `thead` - Heading row of a table

A table consists of formatting information and data entries. There are quite a few attributes that can be adjusted to tweak the display of your data. The examples show only a few of the basic formatting attributes. For more detailed description is in the DocBook Reference documentation.

4.1. A simple table

Let us look at an example to see how a table is constructed.

Example 4.1. A regular simple table

```

1 <table>
  <title>Number of Visitors</title>
  <tgroup cols="3">
    <thead>
5      <row>
        <entry>Month</entry><entry>Week</entry><entry>Visitors</entry>
      </row>
    </thead>
    <tfoot>
10     <row>
        <entry>Total</entry><entry></entry><entry>1833</entry>
      </row>
    </tfoot>
    <tbody>
15     <row>
        <entry>March</entry><entry>1</entry><entry>634</entry>
      </row>
      <row>
        <entry>March</entry><entry>2</entry><entry>657</entry>
20     </row>
      <row>
        <entry>March</entry><entry>3</entry><entry>542</entry>
      </row>
    </tbody>
  </tgroup>
</table>

```

```

25  </tgroup>
    </table>

```

As shown in Example 4.1, tables begin with the `<table>` (or `<informaltable>`) tag. Next, a title has to be defined using the `title` tag when using a regular table (line 2). Finally, we have the `tgroup` element which contains all of the header, footer, and row information (lines 3 to 25). There can be more than one `tgroup` if formatting options have to change for the different sections of the table. The `tgroup` tag (line 3) has a number of optional formatting parameters, but the `cols` attribute, which specifies the number of columns, is required. The `thead` (lines 4 to 8), `tfoot` (lines 9 to 13), and `tbody` (lines 14 to 24) elements contain the data in your table. Data in the `thead` appears at the top of the table, `tbody` appears in the middle, and `tfoot` appears at the end of the table.

Data in a table is contained in rows and entries, using `row` (lines 5-7, 10-12, 15-17, etc.) and `entry` (lines 6, 11, 16, etc.) elements, respectively. A table can be embedded inside another with the `entrytbl` element.

The above example would look something like this when converted:

Table 4.1. Number of Visitors

Month	Week	Visitors
March	1	634
March	2	657
March	3	542

4.2. An informal table

The structure of an informal table is similar to that of a normal table, but it has no title. Therefore, we present in Example 4.2 how such a table can be marked up. In most cases this example is rich enough to let you construct most of your real-life table.

Example 4.2. A simple informal table (no title)

```

1  <informaltable frame="all">
    <tgroup cols="4">
        <thead>
            <row>
2      <entry><emphasis>Title column 1</emphasis></entry>
          <entry><emphasis>Title column 2</emphasis></entry>
          <entry><emphasis>Title column 3</emphasis></entry>
          <entry><emphasis>Title column 4</emphasis></entry>
            </row>
10  </thead>
        <tbody>
            <row>
          <entry>Row 1 column 1</entry><entry>Row 1 column 2</entry>
          <entry>Row 1 column 3</entry><entry>Row 1 column 4</entry>
15  </row>
            <row>
          <entry>Row 2 column 1</entry><entry>Row 2 column 2</entry>
          <entry>Row 2 column 3</entry><entry>Row 2 column 4</entry>
            </row>
20  <row>
          <entry>Row 3 column 1</entry><entry>Row 3 column 2</entry>
          <entry>Row 3 column 3</entry><entry>Row 3 column 4</entry>
            </row>
        </tbody>
    </tgroup>
25 </informaltable>

```

Title column 1	Title column 2	Title column 3	Title column 4
Row 1 column 1	Row 1 column 2	Row 1 column 3	Row 1 column 4
Row 2 column 1	Row 2 column 2	Row 2 column 3	Row 2 column 4
Row 3 column 1	Row 3 column 2	Row 3 column 3	Row 3 column 4

4.3. A more complex table with some math

The following table has a title and references some mathematical symbols, such as the \sum ; and \times ; entities. In fact quite a few entities are available, but, as seen in the definition of Euler's constant, the quality is not very high. For more complex mathematical construct MathML markup is to be preferred. For the definition of column alignment, the `align` attribute specifies that the entries in the columns should be aligned on the decimal point of floating point numbers.

Example 4.3. A more complex table

```

1 <table frame="all">
  <title>Physical and mathematical constants</title>
  <tgroup cols="3" align="char" charoff="50" char=".">
    <thead>
5      <row>
        <entry>Quantity</entry>
        <entry>Definition</entry>
        <entry>Value</entry>
      </row>
10    </thead>
    <tbody>
      <row>
        <entry>Planck constant</entry>
        <entry>h</entry>
15      <entry>6.62606876(82)<math>\times 10^{-34}</math> J s</entry>
      </row>
      <row>
        <entry>Boltzmann constant</entry>
        <entry>k</entry>
20      <entry>1.3806503(24)<math>\times 10^{-23}</math> J K<math>^{-1}</math></entry>
      </row>
      <row>
        <entry/>
25      <entry/>
        <entry>8.617342(15)<math>\times 10^{-5}</math> eV K<math>^{-1}</math></entry>
      </row>
      <row>
30      <entry>Euler constant</entry>
        <entry>e<math>\sum</math><sub>k</sub></entry>
        x<sub>k</sub>/k!</entry>
        <entry>2.718281828459(1)</entry>
      </row>
35    </tbody>
  </tgroup>
</table>

```

This is how that table would be rendered in the output mode you are viewing now:

Table 4.2. Physical and mathematical constants

Quantity	Definition	Value
----------	------------	-------

Planck constant	h	$6.62606876(82) \times 10^{-34} \text{ J s}$
Boltzmann constant	k	$1.3806503(24) \times 10^{-23} \text{ J K}^{-1}$
		$8.617342(15) \times 10^{-5} \text{ eV K}^{-1}$
Euler constant	$e = \sum_k x_k / k!$	$2.718281828459(1)$

Chapter 5. Graphics

Documentation software applications often demands screenshots, pictures of icons and buttons, and other graphical elements. DocBook has element to support screenshots, graphics, and inline graphics. Below is a list of tags related to graphical objects.

- `imagedata` - One of the formats encoding the image, such as EPS when printing and PNG when displaying online
- `inlinemediaobject` - A picture, a sound, a text, that can be encoded in several different formats at the same time, to be rendered in-line
- `mediaobject` - A picture, a sound, a text, that can be encoded in several different formats at the same time, not rendered in-line
- `screeninfo` - Information about how a screenshot was produced
- `screenshot` - Representation of what the user sees or might see on a computer screen

The following examples contain a picture as an inline graphic (Example 5.1) and as a screenshot (Example 5.2).

Example 5.1. An inline media object

```

1 <para>
  Here are a bunch of rectangles
  <inlinemediaobject>
    <imageobject> <imagedata fileref="rectangles.eps" format="EPS"/> </imageobject>
5 <imageobject> <imagedata fileref="rectangles.png" format="PNG"/> </imageobject>
  <textobject> <phrase>A bunch of rectangles</phrase> </textobject>
  </inlinemediaobject>
</para>

```

The content of the `inlinemediaobject` element (lines 4 to 6) is displayed alongside text. The `fileref` attribute of the `<imagedata>` tag (lines 4 and 5) contains the name of the graphic file, the `format` attribute contains the type of the graphic file (the appropriate format should be chosen by the rendering application), and the optional `align` attribute changes the alignment of the graphic. The example will look something like this when converted:



Here are a bunch of rectangles

Example 5.2. A screenshot

```

1 <screenshot>
  <screeninfo>Feynman diagrams</screeninfo>
  <mediaobject>
    <imageobject>
5    <imagedata fileref="feyn1batik.png" scale="50"

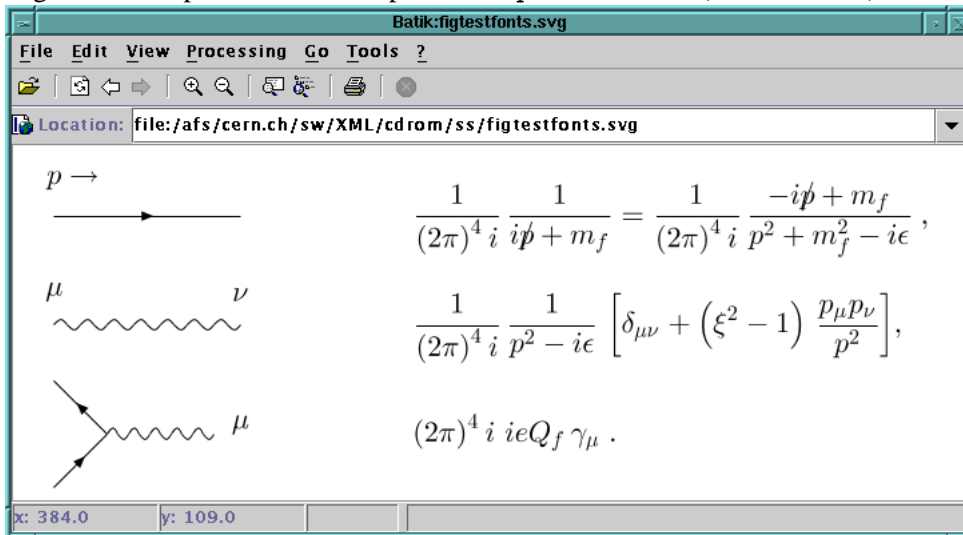
```

```

        align="center" format="PNG"/>
</imageobject>
<textobject>
  <phrase>We show three Feynman diagrams with their
10   corresponding mathematical expressions</phrase>
</textobject>
<caption>
<para>Feynman diagrams and their mathematical representation</para>
</caption>
15 </mediaobject>
</screenshot>

```

The `screenshot` element designates a screenshot with the `screeninfo` element (line 2) providing a textual description of the screen shot. The heart of the screenshot is the `mediaobject` element (lines 3 to 15) which points to a graphic file as explained in the first example. The `imageobject` element (lines 4 to 7) can contain one or more `imagedata` element (e.g. different formats for the same image, as in Example 5.1). In the present case (line 5 and 6) we only specify an PNG image, but we scale it to 50% (`scale` attribute) and center align it (`align` attribute). We also provide a textual description in the `textobject` element (lines 8 to 11) for applications that cannot handle the available graphics type(s). A global description with the help of the `caption` element (lines 12 to 14) ends the element content.



Feynman diagrams and their mathematical representation

Chapter 6. Links

Linking in DocBook is very similar to linking in HTML. The `<link>` and `<xref>` tags are used to link to any element with an `id` in a document, and can also be used to link to other local files as well. The `<ulink>` tag is used when you need to link to a URL. The `<email>` tag is a specialized form of the `ulink` tag used for email addresses. `<anchor>` is used to mark a spot in the text that you want to reference later with a Link. An overview of these most common link elements of DocBook follows.

- `anchor` - Spot in text
- `email` - Email address in an address
- `link` - Hypertext link
- `olink` - A link that addresses its target indirectly, through an entity
- `ulink` - A link that addresses its target by means of a URL, a Uniform Resource Locator
- `xref` - a hypertext link to an element, where the processing system has to generate appropriate cross-reference text

Example 6.1. Five kinds of links

```

1 <sect1 id="sec.link">
  <title id="sec.title.link">Docbook link types</title>
  <para id="par.link">
    This is a paragraph that will be linked to with a link tag.
2 If you want to know what's cool in HEP, point your browser
  at the <ulink url="http://www.cern.ch">CERN Web Page</ulink>.
  <anchor id="einstein.anchor"/>Einstein has been there.</para>
3 </para>
  The above paragraph is reference <link linkend="par.link">here</link>.
4
10 <xref linkend="sec.link"/> (<xref linkend="sec.link"
  endterm="sec.title.link"/>) is a reference to a section.
  My email address is <email>michel.goossens@cern.ch</email>
  </para>
</sect1>

```

We see that to address their link targets the `link` and `xref` elements use the `linkend` attribute (lines 9 and 10), while the `ulink` element uses the `url` attribute (line 6). The `link` and `xref` elements can also have a `endterm` attribute, which specifies that the content of the element with the value of its `id` attribute equal to the value of `endterm` must be used as the text of the cross-reference (line 11).

The above example would be displayed something like what is shown below.

```

1 1. Docbook link types

  This is a paragraph that will be linked to with a link tag.
  If you want to know what is cool in HEP, point your browser
2 at the CERN Web Page [http://www.cern.ch].
  Einstein has been there.

  The above paragraph is reference here. Section 1. (Docbook
  link types) is a reference to a section. My email address is
10 <michel.goossens@cern.ch>.

```

We see how on lines 8 and 9 the work "Section 1." was generated by the application. Between brackets we find the contents of the `title` element, since the value of its `id` attribute (`sec.title.link`) corresponds to that specified on the `endterm` attribute of the `xref` element. Thus, we can finetune what we want to display as cross-reference information by making optimal use of the `id` attributes of the structural document elements.

Chapter 7. Describing the Application's Interface

7.1. Examples

Often we want to include examples of source code, commands, or GUI actions in our documentation. DocBook has many tags to support these needs. Some of the relevant element are listed below.

- `example` - Example of a computer program or related information
- `informalexample` - Untitled example
- `literallayout` - Wrapper for lines set off from the main text that are not tagged as screens, examples, or `programlisting`, in which line breaks and leading white space are to be regarded as significant
- `programlisting` - Listing of all or part of a program
- `screen` - Text that a user sees or might see on a computer screen

Example 7.1. An example

```
<example>
<title>A BASIC Example</title>
<programlisting linenumbers='numbered'>
10 PRINT "HELLO WORLD"
20 GOTO 10
</programlisting>
</example>
```

In this first example, we have a listing of a simple BASIC program. The code contained in the `programlisting` element is displayed with the spacing and line breaks intact which is very useful for code examples and similar situations where you must preserve the literal formatting. The `literallayout` and `screen` elements work similarly, but they are used to indicate different types of content, in particular `screen` elements contain output that would appear on the screen, while `literallayout` elements are used for any other text that must be rendered with line breaks and tabs.

The example would look something like this when converted:

Example 7.2. A BASIC Example

```
1
2 10 PRINT "HELLO WORLD"
3 20 GOTO 10
4
```

One problem with the `screen`, `literallayout`, and `programlisting` elements is that all text is rendered literally, but DocBook tags are still interpreted as tags and not text. If you do not want tags and entity references interpreted, you can use a `<![CDATA []]>` section, which labels the text contained within the inner brackets as character data that should not be interpreted by the XML parser. Any text within the brackets will remain as-is after the conversion, so the example above will successfully reproduce its tags.

Example 7.3. Displaying markup in a CDATA section

```
<example>
<title>CDATA sections and markup</title>
<screen>
<![CDATA[
<para>This is a DocBook example.</para>
```

```
]]>
</screen>
</example>
```

This is what the markup example would look like when converted:

Example 7.4. CDATA sections and markup

```
<para>This is a DocBook example.</para>
```

7.2. GUI Interface Elements

- `accel` - keycap used with a meta key to activate a graphical user interface
- `action` - Function invoked in response to a user event
- `guibutton` - Text on a button in a graphical user interface
- `guiicon` - Graphic and, or, text appearing as a icon in a graphical user interface
- `guilabel` - Text in a graphical user interface
- `guimenu` - Name of a menu in a graphical user interface
- `guimenuitem` - Name of a terminal menu item in a graphical user interface
- `guisubmenu` - Name of a submenu in a graphical user interface
- `interface` - Element of a graphical user interface
- `interfacedefinition` - Full or short name of a formal specification of a graphical user interface
- `keycap` - Text printed on a physical key on a computer keyboard, not necessarily the same thing as a KeyCode
- `keycode` - Computer's numeric designation of a key on a computer keyboard
- `keycombo` - Combination of input actions
- `keysym` - Key symbol name, which is not necessarily the same thing as a Keycap
- `menuchoice` - Menu selection or series of such selections
- `mousebutton` - Conventional name of a mouse button

One could almost say that there are too many tags in DocBook for describing GUI elements. Most of the tags listed above can be used in a variety of contexts, but a few, such as `<keycap>`, must be used within other tags. The example and explanation below will not cover all of the tags listed above. This list is for your convenience since the DocBook Reference does not group tags by their function.

All of the GUI tags can be used within the context of a regular paragraph. So if I wanted to talk about the Trash icon or the Empty Trash button, I would just use the `<guiicon>` and `<guibutton>` tags like this: `<guiicon>Trash</guiicon>` icon, `<guibutton>Empty Trash</guibutton>` button. Note that all GUI tags may also contain inline graphics.

Below is a more complicated example of GUI tag usage.

Example 7.5. guimenu and shortcut

```

<variablelist>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>n</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>New</guimenuitem>
</menuchoice></term>
<listitem><para><action>Creates a new document</action></para></listitem>
</varlistentry>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>s</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>Save</guimenuitem>
</menuchoice></term>
<listitem><para><action>Saves the document</action></para></listitem>
</varlistentry>
<varlistentry>
<term><menuchoice>
<shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>q</keycap></keycombo>
</shortcut>
<guimenu>File</guimenu>
<guimenuitem>Quit</guimenuitem>
</menuchoice></term>
<listitem><para><action>Quits</action> application>Kapp</application></para></listitem>
</varlistentry>
</variablelist>

```

The most complicated part of this example is the `<Shortcut>` tag which labels keyboard shortcuts for menuitems. Shortcut contains either a `KeyCombo` or a single `KeyCap` that contains the key or group of keys the user would press to invoke that menuitem from the keyboard. It is important to use the `KeyCombo` and `KeyCap` tags within the `Shortcut` tag because it is incorrect to use character data (the `Ctrl-q` text for example) within a shortcut.

Other tags worth mentioning from the example are `menuchoice`, `action`, and `application`. `menuchoice` labels a menu choice and should contain the shortcut (if any) the name of the menu in `GUIMenu`, and the name of the menuitem in `guimenuitem`. `Action` simply labels a phrase that describes what the menuitem (or other interface element) does. `application` is a tag used to label the names of applications.

The example would look something like this when converted:

File->New (**Ctrl-n**)
Creates a new document

File->Save (**Ctrl-s**)
Saves the document

File->Quit (**Ctrl-q**)
QuitsKapp

7.3. Command Line Elements

The following tags are used to label elements of a command:

- `arg` - Argument in a `cmdsynopsis`
- `cmdsynopsis` - Synopsis for a command
- `command` - Executable program, or the entry a user makes to execute a command
- `computeroutput` - Data presented to the user by a computer
- `envvar` - Environmental variable
- `filename` - Name of a file, possibly including pathname
- `funcdef` - Name and return value of the function
- `funcprototype` - Prototype of the function
- `funcsynopsis` - Syntax summary for a function definition
- `funcsynopsisinfo` - General information on how to use the function
- `function` - Name of the function, routine, or method
- `literal` - Literal string, used in-line, that is part of data in a computer
- `option` - Option for a computer program command
- `paramdef` - Data type information and the name of the parameter this information applies to
- `parameter` - Part of an instruction to a computer
- `prompt` - Character indicating the start of an input field in a computer display.
- `replaceable` - Content that may be replaced in a synopsis or command line
- `symbol` - Name that is replaced by a value before processing
- `type` - Classification of a value
- `userinput` - Data entered by the user

There are two situations in which you want to describe a command: showing an example of a command typed on the command line and a detailed description of all of the arguments and options to a command like you would see in a man page.

DocBook supports both of these contexts with the `<command>` and `<cmdsynopsis>` tags.

Example 7.6. A command and its output

```
<screen>
<prompt>bash$</prompt> <command>twiddle <replaceable>myfile</replaceable>
</command>
twiddling myfile.....done!
</screen>
```

Would appear as:

```
bash$twiddle -c 1 myfile
twiddling myfile.....done!
```

The `<command>` tag can also be used within a paragraph to mark the name of a command. For example:

The `<command>twiddle</command>` command is used to twiddle files files. Twiddled files will be marked with the `.twid` extension, so if I `<command>twiddle</command> <replaceable>myfile</replaceable>` then it will become `<replaceable>myfile.twid</replaceable>`. Errors are written to the file `<filename>twiddle.err</filename>`.

The **twiddle** command is used to twiddle files files. Twiddled files will be marked with the `.twid` extension, so if I **twiddle***myfile* then it will become *myfile.twid*. Errors are written to the file `twiddle.err`.

The `<prompt>` tag is simply used to label the prompt in a command line. Replaceable labels text that should be replaced by the user. In the example, `myfile` is just an arbitrary name for a file since we don't know and don't care what the name of the file is, we just want to show how the command is used. If a filename in a command is known, use the `<filename>` tag instead.

Marking up a `cmds` synopsis is a bit more difficult. Here is an example from the DocBook Reference:

Example 7.7. A `cmds` synopsis

```
<cmdsynopsis>
  <!-- This is a synopsis for the command foo.
        The options -a and -x are optional and exclusive
        The option -c takes a cheese and is optional and repeatable
        The options -t and -k are referred to in another fragment
        The options -i, -j, and -k are required and exclusive
        The option -f takes a filename and is required
        The -t and -k options specify the kind of milk and mold in an
        optional and repeatable group
  -->
  <command>foo</command>
  <group>
    <arg>-a</arg>
    <arg>-x</arg>
  </group>
  <group>
    <arg rep="repeat">-c <replaceable>cheese</replaceable></arg>
    <synopfragmentref linkend="cheesetype">cheesetype</synopfragmentref>
  </group>
  <group choice="req">
    <arg>-i</arg>
    <arg>-j</arg>
    <arg>-k</arg>
  </group>
  <arg choice="req">-f <replaceable>filename</replaceable></arg>
  <synopfragment id="cheesetype">
    <group rep="repeat">
      <arg>-t <replaceable>milk</replaceable></arg>
      <arg>-k <replaceable>mold</replaceable></arg>
```

```

    </group>
  </synopfragment>
</cmdsynopsis>

```

Which looks like this:

```

foo [-a | -x] [-c cheese(1)] {-i | -j | -k} {-f filename}
(1) [-t milk | -k mold...]

```

A cmdsynopsis contains one command, groups of related args, independent args, and synopfragments. The <arg> labels arguments to the command. arg has two attributes: choice and rep. choice is used to indicate whether the tag is optional (the default), required (req), or to be displayed without any decoration (plain). The <group> tag is used to group together related args. synopfragment is the most complicated of the cmdsynopsis tags. It is used to provide a more detailed description of options for an argument. A synopfragment consists of two parts: the synopfragment, which contains the additional Args, and the synopfragmentref which points to the detailed description.

7.4. Elements for classes and methods

For describing classes and methods DocBook offers the following elements.

- **classname** - The name of a class, in the object-oriented programming sense
- **classsynopsis** - The syntax summary for a class definition
- **classsynopsisinfo** - Information supplementing the contents of a ClassSynopsis
- **exceptionname** - The name of an exception
- **fieldsynopsis** - The name of a field in a class definition
- **methodname** - The name of a method
- **methodparam** - Parameters to a method
- **methodsynopsis** - A syntax summary for a method
- **modifier** - Modifiers in a synopsis
- **ooclass** - A class in an object-oriented programming language
- **ooexception** - An exception in an object-oriented programming language
- **oointerface** - An interface in an object-oriented programming language

An example of a Java Class synopsis and its typeset result follows.

Example 7.8. Class synopsis

```

<classsynopsis language="java">
<ooclass>
<modifier>public</modifier>
<classname>TextFileWriter</classname>
</ooclass>
<ooclass>
<classname>HandlerBase</classname></ooclass>
<fieldsynopsis>
<modifier>private</modifier>
<type>Writer</type>
<varname>writer</varname>
</fieldsynopsis>
<fieldsynopsis>
<modifier>public</modifier>
<type>String</type>
<varname>writerName</varname>
<initializer>"MyWriter"</initializer>
</fieldsynopsis>
<methodsynopsis>
<modifier>static</modifier>
<modifier>public</modifier>
<void/>
<methodname>write</methodname>
<methodparam>
<type>ResultTreeFragment</type>
<parameter>frag</parameter>
</methodparam>
<methodparam>
<type>String</type>
<parameter>file</parameter>
</methodparam>
<exceptionname>SAXException</exceptionname>
</methodsynopsis>
</classsynopsis>

public TextFileWriter extends HandlerBase {

    private Writer writer ;

    public String writerName = "MyWriter";

    static public void write(ResultTreeFragment frag,
                           String file)
        throws SAXException;
}

```

Another example of a class definition, this time for IDL, follows.

```

<classsynopsis class="interface" language="idl">
<oointerface>
<interfacename>Element</interfacename></oointerface>
<oointerface>
<interfacename>Node</interfacename></oointerface>
<fieldsynopsis>
<modifier>readonly</modifier>
<modifier>attribute</modifier>
<type>DOMString</type>
<varname>tagName</varname>
</fieldsynopsis>
<methodsynopsis>
<type>DOMString</type>

```

```

<methodname>getAttribute</methodname>
<methodparam>
<modifier>in</modifier>
<type>DOMString</type>
<parameter>name</parameter>
</methodparam>
</methodsynopsis>
<methodsynopsis>
<void/>
<methodname>setAttribute</methodname>
<methodparam>
<modifier>in</modifier>
<type>DOMString</type>
<parameter>name</parameter>
</methodparam>
<methodparam>
<modifier>in</modifier>
<type>DOMString</type>
<parameter>value</parameter>
</methodparam>
<exceptionname>DOMException</exceptionname>
</methodsynopsis>
</classsynopsis>

```

```

interface implementsElement, Node {

    readonly attribute DOMString tagName ;

    DOMString getAttribute(in DOMString name);

    void setAttribute(in DOMString name, in DOMString value)
        raises(DOMException);
}

```

7.5. Describing an API

DocBook has a rather detailed way of marking up descriptions of function behaviour. The tag that introduces it is `<functsynopsis>`. Here is an example:

Example 7.9. Describing a function in a C library API

```

<functsynopsis>
<functsynopsisinfo>#include <stdlib.h></functsynopsisinfo>
<funcprototype>
<funcdef>double <function>atof</function></funcdef>
<paramdef>const char *<parameter>nptr</parameter></paramdef>
</funcprototype>
</functsynopsis>

```

Here is how it looks:

```

#include <stdlib.h>

double atof(nptr);
const char *nptr;

```


Chapter 8. Bibliography elements

Bibliographic information is specified inside the `bibliography` element. DocBook implements two styles of bibliography entries, and they have quite different processing expectations. On the one hand, you have *raw* information, that you specify generically inside a `biblioentry` element, which contain a database-like collection of named fields. On the other hand, `bibliomisc` entries *cooked*, i.e., fields occur in the order in which they are to be displayed and additional punctuation may be sprinkled between the fields.

In the following we will mainly emphasize the generic entries. In the text a `citation` element can be used to refer to a bibliographic entry.

- `abbrev` - An abbreviation, especially one followed by a period
- `abstract` - A summary
- `address` - A real-world address, generally a postal address
- `affiliation` - The institutional affiliation of an individual
- `artpagenums` - The page numbers of an article as published
- `author` - The name of an individual author
- `authorblurb` - A short description or note about an author
- `authorgroup` - Wrapper for author information with multiple authors or collaborators
- `authorinitials` - The initials or other short identifier for an author
- `bibliodiv` - A section of a Bibliography
- `biblioentry` - An entry in a Bibliography
- `bibliography` - A bibliography
- `bibliographyinfo` - Meta-information for a Bibliography
- `bibliomisc` - Untyped bibliographic information
- `bibliomixed` - An entry in a Bibliography
- `bibliomset` - A cooked container for related bibliographic information
- `biblioset` - A raw container for related bibliographic information
- `citation` - An inline bibliographic reference to another published work
- `citerefentry` - A citation to a reference page
- `citetitle` - The title of a cited work
- `city` - The name of a city in an address
- `collab` - Identifies a collaborator
- `collabname` - The name of a collaborator
- `confgroup` - A wrapper for document meta-information about a conference

- `contractnum` - The contract number of a document
- `contractsponsor` - The sponsor of a contract
- `contrib` - A summary of the contributions made to a document by a credited source
- `copyright` - Copyright information about a document
- `corppauthor` - A corporate author, as opposed to an individual
- `corpname` - The name of a corporation
- `country` - The name of a country
- `date` - The date of publication or revision of a document
- `edition` - The name or number of an edition of a document
- `editor` - The name of the editor of a document
- `firstname` - The first name of a person
- `honorific` - The title of a person
- `isbn` - The International Standard Book Number of a document
- `issn` - The International Standard Serial Number of a periodical
- `issuenum` - The number of an issue of a journal
- `orgname` - The name of an organization other than a corporation
- `othercredit` - A person or entity, other than an author or editor, credited in a document
- `othername` - A component of a persons name that is not a first name, surname, or lineage
- `pagenums` - The numbers of the pages in a book, for use in a bibliographic entry
- `printhistory` - The printing history of a document
- `pubdate` - The date of publication of a document
- `publisher` - The publisher of a document
- `publishername` - The name of the publisher of a document
- `pubsnumber` - A number assigned to a publication other than an ISBN or ISSN
- `releaseinfo` - Information about a particular release of a document
- `revhistory` - A history of the revisions to a document
- `seriesvolnums` - Numbers of the volumes in a series of books
- `subtitle` - The subtitle of a document
- `surname` - A family name; in western cultures the last name
- `title` - The text of the title of a section of a document or of a formal block-level element
- `titleabbrev` - The abbreviation of a Title
- `volumenum` - The volume number of a document in a set (as of books in a set or articles in a journal)

8.1. Handling

Example Example 8.1 is an XML DocBook source document that contains a short text with a few citations to books or Internet documents, that are defined in the bibliography.

Example 8.1. Example of A bibliography and citations

```

1 <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
  "/afs/cern.ch/sw/XML/cdrom/www.nwalsh.com/docbook/xml/docbookx.dtd">
  <article>
    <sect1>
      5 <title>DocBook and bibliographic references</title>

      <para>
        The DocBook markup system is described in the printed book
        <emphasis>DocBook - The Definitive Guide</emphasis> <citation><xref
10 linkend="bib.TDG99" endterm="bib.TDG99.abbrev"/></citation>. There is
        an up-to-date local HTML copy at CERN <citation><xref
        linkend="bib.TDGupd" endterm="bib.TDGupd.abbrev"/></citation>
        </para>

15 <para>
        The XSLT language, that is used to transform the DocBook XML sources
        into HTML, XSLT-F0, etc. is described in the W3C recommendation
        <emphasis>XSL Transformations <citation><xref linkend="bib.xsltrec"
        endterm="bib.xsltrec.abbrev"/></citation></emphasis>. There is a nice
20 XSLT tutorial <citation><xref linkend="bib.XSLTtut.Nic"
        endterm="bib.XSLTtut.Nic.abbrev"/></citation>, as well as Mike Kay's book
        <emphasis>XSLT Programmer's Reference</emphasis> <citation><xref
        linkend="bib.xsltkey2" endterm="bib.xsltkey2.abbrev"/></citation>.
        The style sheets themselves are also described <citation><xref
25 linkend="bib.docbook.xslt" endterm="bib.docbook.xslt.abbrev"/></citation>.
        </para>

        <para>
          The are two discussion lists, about Docbook <citation><xref
30 linkend="bib.docbook.list" endterm="bib.docbook.list.abbrev"/></citation>
          and its applications <citation><xref linkend="bib.docbook.apps.list"
          endterm="bib.docbook.apps.list.abbrev"/></citation>.
        </para>
      </sect1>

35 <bibliography id="sec.bibliography">
  <title>Bibliography</title>

  <biblioentry id="bib.xsltrec">
40 <abbrev id="bib.xsltrec.abbrev">REC-XSLT</abbrev>
    <editor><firstname>James</firstname><surname>Clark</surname></editor>
    <title><ulink url="http://www.w3.org/TR/xslt">XSL Transformations
      (XSLT) Version 1.0</ulink></title>
    <publishername>W3C Recommendation</publishername>
45 <pubdate>16 November 1999</pubdate>
  </biblioentry>

  <biblioentry id="bib.XSLTtut.Nic">
    <abbrev id="bib.XSLTtut.Nic.abbrev">NICXSLTUT</abbrev>
    <author><firstname>Miloslav</firstname><surname>Nic</surname></author>
50 <title><ulink url="http://www.zvon.org/xxl/XSLTutorial/Books/Book1/index.html">XSLT Tutorial</ulink></title>
  </biblioentry>

  <biblioentry id="bib.xsltkey2">
    <abbrev id="bib.xsltkey2.abbrev">KAY2001</abbrev>
    <author><firstname>Michael</firstname><surname>Kay</surname></author>
55 <title>XSLT 2nd Edition</title>
    <subtitle>Programmer's Reference</subtitle>

```

```

    <pubdate>2001</pubdate>
    <edition>2</edition>
    <isbn>ISBN: 1861005067</isbn>
60  <pagenums>940</pagenums>
</biblioentry>
<biblioentry id="bib.TDG99">
    <abbrev id="bib.TDG99.abbrev">TDG1999</abbrev>
    <authorgroup>
65    <author><firstname>Norman</firstname><surname>Walsh</surname></author>
        <author><firstname>Leonard</firstname><surname>Muellner</surname></author>
    </authorgroup>
    <title>DocBook</title>
    <subtitle>The Definitive Guide</subtitle>
70  <pubdate>1999</pubdate>
    <edition>1</edition>
    <isbn>ISBN: 156592-580-7</isbn>
    <pagenums>648</pagenums>
    <releaseinfo>
75    <ulink url="http://www.oreilly.com/catalog/docbook/index.html">Oreilly's catalog entry</ulink>.
        An online version <ulink url="http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html">is avail-
able</ulink>
    </releaseinfo>
</biblioentry>
<biblioentry id="bib.TDGupd">
80  <abbrev id="bib.TDGupd.abbrev">TDG2002</abbrev>
    <authorgroup>
        <author><firstname>Norman</firstname><surname>Walsh</surname></author>
        <author><firstname>Leonard</firstname><surname>Muellner</surname></author>
    </authorgroup>
85  <title>DocBook 4.2</title>
    <subtitle><ulink url="http://xml.cern.ch/www.docbook.org/tdg/en/html/docbook.html">Updated
        online Reference installed at CERN</ulink></subtitle>
</biblioentry>
<biblioentry id="bib.docbook.xslt">
90  <abbrev id="bib.docbook.xslt.abbrev">DBXSLTDOC</abbrev>
    <authorgroup>
        <author><firstname>Norman</firstname><surname>Walsh</surname></author>
        <author><firstname>Bob</firstname><surname>Stayton</surname></author>
    </authorgroup>
95  <title>DocBook XSL stylesheets</title>
    <releaseinfo>
        <ulink url="http://sourceforge.net/projects/docbook/">The sources n
        <literal>sourceforge</literal></ulink>. At CERN these
        stylesheets are installed locally, as well as the associated <ulink
100  url="http://xml.cern.ch/sourceforge.net/projects/docbook/docbook-xsl-1.49/doc/index.html"> documentation</ulink>
    </releaseinfo>
</biblioentry>
<biblioentry>
<biblioentry id="bib.docbook.list">
105  <abbrev id="bib.docbook.list.abbrev">DBLIST</abbrev>
    <title>DocBook discussion list <email>docbook@lists.oasis-open.org</email></title>
    <releaseinfo><ulink url="http://lists.oasis-open.org/archives/docbook/">List archive</ulink></releaseinfo>
</biblioentry>
<biblioentry id="bib.docbook.apps.list">
110  <abbrev id="bib.docbook.apps.list.abbrev">DBAPPSLIST</abbrev>
    <title>DocBook applications discussion list <email>docbook-apps@lists.oasis-open.org</email></title>
    <releaseinfo><ulink url="http://lists.oasis-open.org/archives/docbook-apps/">List archive</ulink></releaseinfo>
</biblioentry>
115 </bibliography>
</article>

```

The result of transforming the xml file of Example 8.1 into PDF (with docbook2pdf, see Figure 8.1) and HTML (with docbook2html, see Figure 8.2).

Figure 8.1. Citations and bibliography (PDF output)

DocBook and bibliographic references

The DocBook markup system is described in the printed book *DocBook - The Definitive Guide* [TDG1999]. There is an up-to-date local HTML copy at CERN [TDG2002].

The XSLT language, that is used to transform the DocBook XML sources into HTML, XSLT-FO, etc. is described in the W3C recommendation *XSL Transformations [REC-XSLT]*. There is a nice XSLT tutorial [NICXSLTUT], as well as Mike Kay's book *XSLT Programmer's Reference* [KAY2001]. The style sheets themselves are also described [DBNSLTDOC].

There are two discussion lists, about Docbook [DBLIST] and its applications [DBAPPSLIST].

Bibliography

[REC-XSLT] James Clark. *XSL Transformations (XSLT) Version 1.0* [<http://www.w3.org/TR/xslt>]. W3C Recommendation. 16 November 1999.

[NICXSLTUT] Miloslav Nic. *XSLT Tutorial* [<http://www.zvon.org/xsl/XSLTutorial/Books/Book1/index.html>].

[KAY2001] Michael Kay. *XSLT 2nd Edition. Programmer's Reference*. 2001. ISBN: 1861005067.

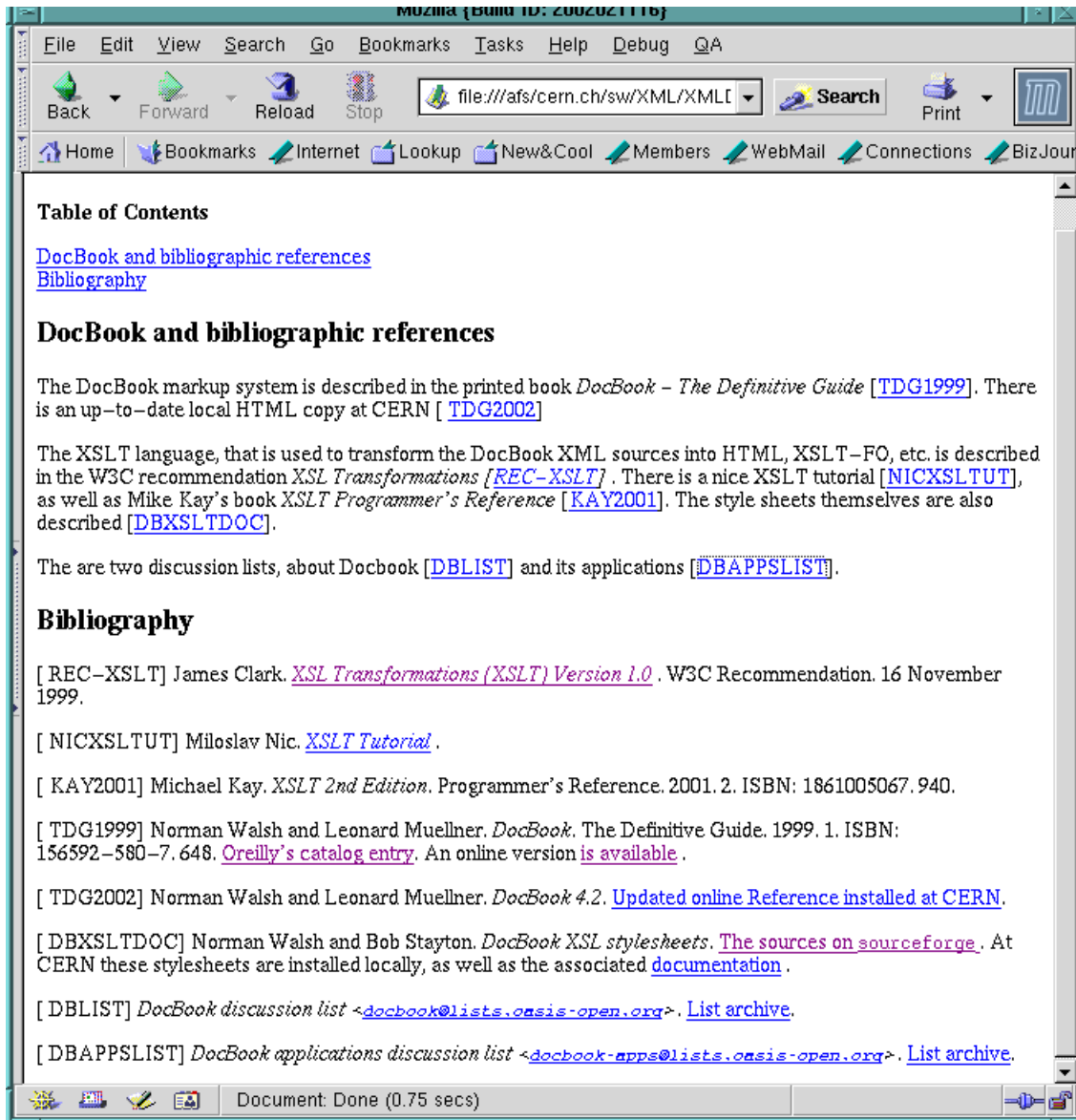
[TDG1999] Norman Walsh and Leonard Muellner. *DocBook. The Definitive Guide*. 1999. ISBN: 156592-580-7. O'Reilly's catalog entry [<http://www.oreilly.com/catalog/docbook/index.html>]. An online version is available [<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>].

[TDG2002] Norman Walsh and Leonard Muellner. *DocBook 4.2*. Updated online Reference installed at CERN [<http://xml.cern.ch/www.docbook.org/tdg/en/html/docbook.htm>].

[DBNSLTDOC] Norman Walsh and Bob Stayton. *DocBook XSL stylesheets*. The sources on [sourceforge](http://sourceforge.net/projects/docbook) [<http://sourceforge.net/projects/docbook>]. At CERN these stylesheets are installed locally, as well as the associated documentation [<http://xml.cern.ch/sourceforge.net/projects/docbook/docbook-xsl-1.49/doc/index.html>].

[DBLIST] *DocBook discussion list* <docbook@lists.oasis-open.org>. List archive [<http://lists.oasis-open.org/archives/docbook/>].

[DBAPPSLIST] *DocBook applications discussion list* <docbook-apps@lists.oasis-open.org>. List archive [<http://lists.oasis-open.org/archives/docbook-apps/>].

Figure 8.2. Citations and bibliography (Displayed by the Mozilla browser)

Chapter 9. Miscellaneous

9.1. Labelling Tags

The application tag labels the name of an application. The <markup> tag is used to label marked up text, such as HTML or TeX tags.

- `application` - Name of a software program
- `markup` - String of formatting Markup in text, which it is desired to represent literally

9.2. Formatting Tags

The following tags are fairly self-explanatory. *This is an emaphasized sentence with ^{superscripted} and subscripted text.*

- `emphasis` - Emphasized text
- `subscript` - Subscript
- `superscript` - Superscript

9.3. Warnings, Tips, and Notes

The following tags are used to set off paragraphs from the rest of the text. They all wrap around paragraphs.

- `caution` - Admonition set off from the text
- `important` - Admonition set off from the text
- `note` - Message to the user, set off from the text
- `tip` - Suggestion to the user, set off from the text
- `warning` - Admonition set off from the text

```
<warning><para>Danger, Will Robinson!</para></warning>
```

The example would look something like this:

Warning

Danger, Will Robinson!

Chapter 10. If you want to know more

As already explained in the introduction, various several useful parts of DocBook are not discussed or only briefly mentioned in this Tutorial. A reference of all element names and a one-line explanation is in Appendix C. However, for more detailed information you should consult the resources list that follows.

DocBook resources

[REC-XSLT] James Clark. *XSL Transformations (XSLT) Version 1.0* [<http://www.w3.org/TR/xslt>]. W3C Recommendation. 16 November 1999.

[NICXSLTUT] Miloslav Nic. *XSLT Tutorial* [<http://www.zvon.org/xxl/XSLTutorial/Books/Book1/index.html>].

[KAY2001] Michael Kay. *XSLT 2nd Edition*. Programmer's Reference. 2001. 2. ISBN: 1861005067. 940.

[TDG1999] Norman Walsh and Leonard Mueller. *DocBook*. The Definitive Guide. 1999. ISBN: 156592-580-7. Oreilly's catalog entry [<http://www.oreilly.com/catalog/docbook/index.html>]. Online version [<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>].

[TDG2002] Norman Walsh and Leonard Mueller. *DocBook 4.2*. Updated online Reference installed at CERN [<http://xml.cern.ch/www.docbook.org/tdg/en/html/docbook.html>].

[DBXSLTDOC] Norman Walsh and Bob Stayton. *DocBook XSL stylesheets (version 1.49, February 2002)*. The sources are on sourceforge [<http://sourceforge.net/projects/docbook/>]. At CERN these stylesheets are installed locally, as well as the associated documentation [<http://xml.cern.ch/sourceforge.net/projects/docbook/docbook-xsl-1.49/doc/index.html>].

[OASISTAB] Norman Walsh. *Organization for the Advancement of Structured Information Standards (OASIS) Technical Memorandum TR 9901:1999*. XML Exchange Table Model Document Type Definition. 1999. OASIS. HTML source [<http://www.oasis-open.org/html/tm9901.htm>].

[DBLIST] *DocBook discussion list* <docbook@lists.oasis-open.org>. List archive [<http://lists.oasis-open.org/archives/docbook/>].

[DBAPPSLIST] *DocBook applications discussion list* <docbook-apps@lists.oasis-open.org>. List archive [<http://lists.oasis-open.org/archives/docbook-apps/>].

Appendix A. Emacs PSGML mode tips

Several XML editors exist. A very flexible and free one is emacs with its psgml mode, which, amongst other things, offers emacs-style completion *on tags*. Also, it lets you check your file (if it can read the DTD), shows which elements are possible at a given point in the source file, etc.

The emacs editor is one of the most often used editors on Unix (and today is also popular on Microsoft Windows). It is highly customizable via Lisp code and can easily cope with the syntax of various languages. Lennart Staflin developed the basic SGML/XML support in the form of an emacs “macro mode” psgml [http://www.lysator.liu.se/projects/about_psgml.html]. For editing DTDs Tony Graham contributed his emacs macros tdttd [<http://www.menteith.com/tdtd/>].

All these modes present you with menus and commands for inserting tags, thus helping you to enter only contextually valid tags and allowing you to edit attribute values in a separate window with information about types and defaults. They also identify structural errors.

Figure A.1. Emacs and its psgml mode with the DocBook DTD

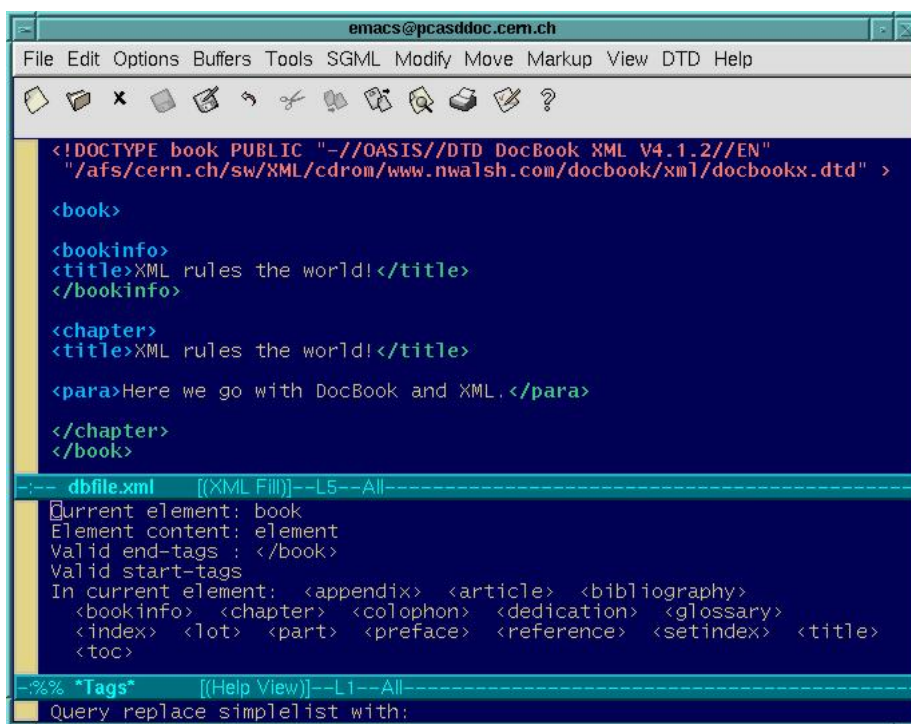


Figure A.1 shows an example of a small DocBook file where together with James Clark’s nsgmls SGML/XML parser and emacs’s psgml mode one can verify at each point in the document which element, attribute, etc. is allowed. Moreover, colours are used to distinguish the different XML language components, making editing significantly easier.

If you type "C-c C-e" (or go into the Markup window, and choose the Insert Element entry) emacs will prompt you for an element, and offer as completions all elements valid at that point.

Once it inserts the element, it inserts it with any required following elements along with a comment saying which ones you could put later on.

As an example, if you type the following into the a DocBook emacs:

C-c C-e variab<SPACE BAR><RETURN>

a `variablelist` and its expansion is inserted into the text buffer, as follows:

```
<variablelist>
  <varlistentry>
    <term></term>
```

```

<listitem>
  <!-- one of (epigraph authorblurb abstract highlights
  comment bridgehead anchor sidebar procedure msgset table
  figure example equation informatable informalexample
  informalequation graphicco graphic blockquote address
  simpara para formalpara funcsynopsis cmdsynopsis synopsis
  screenshot screenco screen programlistingco programlisting
  literallayout warning tip note important caution
  variablelist simplelist segmentedlist orderedlist
  itemizedlist glosslist calloutlist) -->
</listitem>
</varlistentry>
</variablelist>

```

As another example, when we type:

```
C-c C-e i<SPACE BAR>
```

we get the following completions:

Click mouse-2 on a completion to select it.
In this buffer, type RET to select the completion near point.

Possible completions are:

important	indexterm
informalequation	informalexample
informatable	itemizedlist

The current element is closed by typing "C-c /".

Appendix B. A complete example of a DocBook article

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
  <!DOCTYPE article SYSTEM
    "/afs/cern.ch/sw/XML/XMLBIN/share/www.oasis-open.org/docbook/xmltd-4.2/docbookx.dtd"
  [
5    <!ENTITY LaTeX "LaTeX">
    <!ENTITY TeX "TeX">
    <!ENTITY PTeX "PassiveTeX">
    <!ENTITY xmltex "<application>xmltex</application>">
  ]>
10 <article>
  <articleinfo>
    <title>A Docbook document featuring a few formulae</title>
    <author><firstname>Michel</firstname> <surname>Goossens</surname>
    </author>
15 <pubdate>Sunday, 17 March 2002</pubdate>
    <abstract>
      <para>
        This XML document is marked up according the the DocBook schema It
        shows a few elements of the DocBook vocabulary, as well as a couple of
20 examples of mathematical expressions where we used MathML markup.
      </para>
    </abstract>
  </articleinfo>

25 <sect1>
  <title>The Docbook model</title>
  <para>
    DocBook <citation><xref role="bib" linkend="bib.TDG99"
    endterm="bib.TDG99.abbrev"/></citation> proposes an XML model
30 <citation><xref role="bib" linkend="bib.xml"
    endterm="bib.xml.abbrev"/></citation> for marking up technical
    documents. It is particularly well-suited for software reference
    guides and computer equipment manuals.
  </para>

35 <para>
  Docbook contains hundreds of elements to markup up clearly and
  explicitly the different components of an electronic document (book,
  article, reference guide, etc.), not only displaying its hierarchical
40 structure but also indicating the semantical meaning of the various
  elements. The structure of the DTD is optimized to allow for
  customization, thus making it relatively straightforward to add or
  eliminate certain elements or attributes, to change the content model
  for certain structural groups, or to restrict the value that given
45 attributes can take.
  </para>

  <para>
    Norman Walsh developed a set of XSL stylesheets to transform XML
50 documents marked up using the DocBook DTD into HTML or XSL formatting
    objects. The latter can be interpreted by &PTeX; and &xmltex; to
    obtain PDF or PostScript output.
  </para>
  </sect1>

55 <sect1>
  <title>MathML and mathematics</title>
  <para>
    MathML <citation><xref role="bib" linkend="bib.mathml"
60 endterm="bib.mathml.abbrev"/></citation> is a W3C recommendation whose aim
    is to encode mathematical material for teaching and scientific

```

```

communication at all levels.
</para>

65 <para>
  MathML consists of two parts: presentation (notation) et contents
  (meaning). MathML permits the conversion of mathematical information
  into various representations and output formats, including graphical
  displays, speech synthesizers, computer algebra programs, other
70 mathematics typesetting languages, such as &TeX;, plain text,
  printers (PostScript), braille, etc.
  </para>

  <para>
75 MathML has support for efficiently browsing long and complex
  mathematical expressions and offers an extension mechanism. MathML
  code is human readable, easy to generate and process by software
  applications, and well suited for editing (even though MathML syntax
  might appear unnecessarily verbose and complex to the human reader).
80 </para>

  <para>
    The W3C MathML Working Group released the Second version of the the
    MathML Specification at the beginning of 2001. Two public initiatives
85 that allow the display of MathML code directly and that are under
    active development are W3C's <application>Amaya</application> and
    <application>Mozilla</application>. Commercial programs, such as IBM's
    <application>techexplorer</application> (a plugin for
    <application>Netscape</application> or Microsoft's
90 <application>Internet Explorer</application>) or Design Science
    <application>Webeq</application> (using the Java applet technology)
    can display MathML formulae in present day browsers. Several computer
    algebra programs, e.g., <application>Mathematica</application>, or
    editors using e.g., <application>mathtype</application>, offer a
95 user-friendly interface to enter, produce or read mathematics material
    marked up in MathML.
    </para>
  </sect1>

100 <sect1>
  <title>Presentation markup</title>

  <para>
    The <emphasis>presentation</emphasis> part of MathML describes the
105 spacial layout of the different elements of a mathematical expression.
    MathML presentation markup has about thirty elements, that form the
    basis of a mathematical syntax using classical visual layout model.
    Some fifty attributes provide precise control on the exact
    positioning of the various components of the math expression.
110 </para>

  <para>
    The list of presentation elements follows.
    <variablelist>
115 <varlistentry>
      <term>Token elements</term>
      <listitem><para>
        <sgmltag class="element">mi</sgmltag>,<sgmltag class="element">mn</sgmltag>,
        <sgmltag class="element">mo</sgmltag>,<sgmltag class="element">mtext</sgmltag>,
120 <sgmltag class="element">mspace</sgmltag>,<sgmltag class="element">ms</sgmltag>,
        <sgmltag class="element">mchar</sgmltag>,<sgmltag class="element">mglyph</sgmltag>
      </para></listitem>
    </varlistentry>
  </varlistentry>

```

Appendix B. A complete example of a DocBook article

```
125 <term>General layout schemata</term>
    <listitem><para>
        <sgmltag class="element">mrow</sgmltag>,<sgmltag class="element">mfrac</sgmltag>,
        <sgmltag class="element">msqrt</sgmltag>,<sgmltag class="element">mroot</sgmltag>,
        <sgmltag class="element">mstyle</sgmltag>,<sgmltag class="element">merror</sgmltag>,
130 <sgmltag class="element">mpadded</sgmltag>,<sgmltag class="element">mphantom</sgmltag>,
        <sgmltag class="element">mfenced</sgmltag>,<sgmltag class="element">menclase</sgmltag>
    </para></listitem>
    </varlistentry>
    <varlistentry>
135 <term>Script and limit schemata</term>
    <listitem><para>
        <sgmltag class="element">msub</sgmltag>,<sgmltag class="element">msup</sgmltag>,
        <sgmltag class="element">msubsup</sgmltag>,<sgmltag class="element">munder</sgmltag>,
        <sgmltag class="element">mover</sgmltag>,<sgmltag class="element">munderover</sgmltag>,
140 <sgmltag class="element">mmultiscripts</sgmltag>
    </para></listitem>
    </varlistentry>
    <varlistentry>
        <term>Tables and matrices</term>
145 <listitem><para>
        <sgmltag class="element">mtable</sgmltag>,<sgmltag class="element">mtr</sgmltag>,
        <sgmltag class="element">mttd</sgmltag>,<sgmltag class="element">maligngroup</sgmltag>,
        <sgmltag class="element">malignmark</sgmltag>
    </para></listitem>
150 </varlistentry>
    <varlistentry>
        <term>Enlivening expressions</term>
        <listitem><para>
            <sgmltag class="element">maction</sgmltag>
155 </para></listitem>
        </varlistentry>
    </variablelist>
    </para>

160 <sect2>
    <title>A MathML example</title>
    <para>
        A MathML formula can be typeset inline, as here
        <inlineequation>
165 <mml:math>
            <mml:mi>E</mml:mi><mml:mo>=</mml:mo><mml:mi>m</mml:mi><mml:msup><mml:mi>c</mml:mi>
            <mml:mn>2</mml:mn></mml:msup>
        </mml:math>
        </inlineequation>, Einstein's famous equation.
170 </para>

    <para>
        An mathematical equation can also be typeset in display mode using
        DocBook's <sgmltag class="element">informalequation</sgmltag> element,
175 as is shown in the following example containing a matrix:
    </para>

    <informalequation>
    <mml:math>
180 <mml:mrow>
        <mml:mi>A</mml:mi>
        <mml:mo>=</mml:mo>
        <mml:mfenced open="[" close="]">
            <mml:mtable><!-- table or matrix -->
185 <mml:mtr> <!-- row in a table -->
                <mml:td><mml:mi>x</mml:mi></mml:td><!-- table -->
                <mml:td><mml:mi>y</mml:mi></mml:td><!-- entry -->
```

```

        </mml:mtr>
        <mml:mtr>
190      <mml:td><mml:mi>z</mml:mi></mml:td>
        <mml:td><mml:mi>w</mml:mi></mml:td>
        </mml:mtr>
      </mml:mtable>
    </mml:mfenced>
195 </mml:mrow>
    <mml:mtext>.</mml:mtext>
    </mml:math>
  </informalequation>

200 <para>
  Note the two attributes <sgmltag class="attribute">open</sgmltag> and
  <sgmltag class="attribute">close</sgmltag> on the <sgmltag
  class="element">mfenced</sgmltag> element to specify the style of the
  braces to be used. The MathML Specification <citation><xref role="bib"
205 linkend="bib.mathml" endterm="bib.mathml.abbrev"/></citation> contains
  a detailed list of all possible attributes associated to each
  presentation element.
  </para>
</sect2>
210 </sect1>

  <sect1>
    <title>Content markup</title>

215 <para>
  The meaning of mathematical symbols (e.g., sums, products, integrals)
  exists independently of their rendering. Moreover the presentation
  markup of an expression is not necessarily sufficient to evaluate its
  value and use it in calculations. Therefore, MathML defines
220 <emphasis>content</emphasis> markup to explicitly encode the
  underlying mathematical structure of an expression.
  </para>

  <para>
225 It is impossible to cover all of mathematics, so MathML proposes only
  a relatively small number of commonplace mathematical constructs,
  chosen carefully to be sufficient in a large number of
  applications. In addition, it provides a <emphasis>extension
  mechanism</emphasis> for associating semantics with new notational
230 constructs, thus allowing these to be encoded even when they are not
  in MathML content element base collection.
  </para>

  <para>
235 MathML's basic set of content elements was chosen to allow for simple
  coding of most of the formulas used in secondary education, through
  the first year of university. The subject areas targeted are
  arithmetic, algebra, logic and relations, calculus and vector
  calculus, set theory, sequences and series, elementary classical
240 functions, and statistics linear algebra. Using this basic sets more
  complex constructs can be built.
  </para>

  <para>
245 The list of the content elements of MathML follows.
  <variablelist>
    <varlistentry>
      <term>token elements</term>
      <listitem><para>
250 <sgmltag class="element">cn</sgmltag>,<sgmltag class="element">ci</sgmltag>,

```


Appendix B. A complete example of a DocBook article

```
<sgmltag class="element">csymbol</sgmltag> (MathML 2.0).
</para></listitem>
</varlistentry>
<varlistentry>
255 <term>basic content elements</term>
<listitem><para>
<sgmltag class="element">apply</sgmltag>,<sgmltag class="element">reln (deprecated)</sgmltag>,
<sgmltag class="element">fn (deprecated for externally defined
functions)</sgmltag>,
260 <sgmltag class="element">interval</sgmltag>,<sgmltag class="element">inverse</sgmltag>,
<sgmltag class="element">sep</sgmltag>,<sgmltag class="element">condition</sgmltag>,
<sgmltag class="element">declare</sgmltag>,<sgmltag class="element">lambda</sgmltag>,
<sgmltag class="element">compose</sgmltag>,<sgmltag class="element">ident</sgmltag>.
</para></listitem>
265 </varlistentry>
<varlistentry>
<term>arithmetic, algebra and logic</term>
<listitem><para>
<sgmltag class="element">quotient</sgmltag>,<sgmltag class="element">exp</sgmltag>,
270 <sgmltag class="element">factorial</sgmltag>,<sgmltag class="element">divide</sgmltag>,
<sgmltag class="element">max and min</sgmltag>,<sgmltag class="element">minus</sgmltag>,
<sgmltag class="element">plus</sgmltag>,<sgmltag class="element">power</sgmltag>,
<sgmltag class="element">rem</sgmltag>,<sgmltag class="element">times</sgmltag>,
<sgmltag class="element">root</sgmltag>,<sgmltag class="element">gcd</sgmltag>,
275 <sgmltag class="element">and</sgmltag>,<sgmltag class="element">or</sgmltag>,
<sgmltag class="element">xor</sgmltag>,<sgmltag class="element">not</sgmltag>,
<sgmltag class="element">implies</sgmltag>,<sgmltag class="element">forall</sgmltag>,
<sgmltag class="element">exists</sgmltag>,<sgmltag class="element">abs</sgmltag>,
<sgmltag class="element">conjugate</sgmltag>,<sgmltag class="element">arg</sgmltag> (MathML 2.0),
280 <sgmltag class="element">real</sgmltag> (MathML 2.0),<sgmltag class="element">imaginary</sgmltag> (MathML 2.0),
<sgmltag class="element">lcm</sgmltag> (MathML 2.0).
</para></listitem>
</varlistentry>
<varlistentry>
285 <term>relations</term>
<listitem><para>
<sgmltag class="element">eq</sgmltag>,<sgmltag class="element">neq</sgmltag>,
<sgmltag class="element">gt</sgmltag>,<sgmltag class="element">lt</sgmltag>,
<sgmltag class="element">geq</sgmltag>,<sgmltag class="element">leq</sgmltag>,
290 <sgmltag class="element">equivalent</sgmltag> (MathML 2.0),<sgmltag class="element">approx</sgmltag> (MathML 2.0).
</para></listitem>
</varlistentry>
<varlistentry>
<term>calculus and vector calculus</term>
295 <listitem><para>
<sgmltag class="element">int</sgmltag>,<sgmltag class="element">diff</sgmltag>,
<sgmltag class="element">partialdiff</sgmltag>,<sgmltag class="element">lowlimit</sgmltag>,
<sgmltag class="element">uplimit</sgmltag>,<sgmltag class="element">bvar</sgmltag>,
<sgmltag class="element">degree</sgmltag>,<sgmltag class="element">divergence</sgmltag> (MathML 2.0),
300 <sgmltag class="element">grad</sgmltag> (MathML 2.0),<sgmltag class="element">curl</sgmltag> (MathML 2.0),
<sgmltag class="element">laplacian</sgmltag> (MathML 2.0).
</para></listitem>
</varlistentry>
<varlistentry>
305 <term>theory of sets</term>
<listitem><para>
<sgmltag class="element">set</sgmltag>,<sgmltag class="element">list</sgmltag>,
<sgmltag class="element">union</sgmltag>,<sgmltag class="element">intersect</sgmltag>,
<sgmltag class="element">in</sgmltag>,<sgmltag class="element">notin</sgmltag>,
310 <sgmltag class="element">subset</sgmltag>,<sgmltag class="element">prsubset</sgmltag>,
<sgmltag class="element">notsubset</sgmltag>,<sgmltag class="element">notprsubset</sgmltag>,
<sgmltag class="element">setdiff</sgmltag>,<sgmltag class="element">card</sgmltag> (MathML 2.0).
</para></listitem>
```

```

    </varlistentry>
315 <varlistentry>
    <term>sequences and series</term>
    <listitem><para>
        <sgmltag class="element">sum</sgmltag>,<sgmltag class="element">product</sgmltag>,
        <sgmltag class="element">limit</sgmltag>,<sgmltag class="element">tendsto</sgmltag>.
320 </para></listitem>
    </varlistentry>
    <varlistentry>
    <term>elementary classical function</term>
    <listitem><para>
325 <sgmltag class="element">exp</sgmltag>,<sgmltag class="element">ln</sgmltag>,
        <sgmltag class="element">log</sgmltag>,<sgmltag class="element">sin</sgmltag>,
        <sgmltag class="element">cos</sgmltag>,<sgmltag class="element">tan</sgmltag>,
        <sgmltag class="element">sec</sgmltag>,<sgmltag class="element">csc</sgmltag>,
        <sgmltag class="element">cot</sgmltag>,<sgmltag class="element">sinh</sgmltag>,
330 <sgmltag class="element">cosh</sgmltag>,<sgmltag class="element">tanh</sgmltag>,
        <sgmltag class="element">sech</sgmltag>,<sgmltag class="element">csch</sgmltag>,
        <sgmltag class="element">coth</sgmltag>,<sgmltag class="element">arcsin</sgmltag>,
        <sgmltag class="element">arccos</sgmltag>,<sgmltag class="element">arctan</sgmltag>,
        <sgmltag class="element">arccosh</sgmltag>,<sgmltag class="element">arccot</sgmltag>,
335 <sgmltag class="element">arccoth</sgmltag>,<sgmltag class="element">arccsc</sgmltag>,
        <sgmltag class="element">arccsch</sgmltag>,<sgmltag class="element">arcsec</sgmltag>,
        <sgmltag class="element">arcsech</sgmltag>,<sgmltag class="element">arcsinh</sgmltag>,
        <sgmltag class="element">arctanh</sgmltag>.
    </para></listitem>
340 </varlistentry>
    <varlistentry>
    <term>statistics</term>
    <listitem><para>
        <sgmltag class="element">mean</sgmltag>,<sgmltag class="element">sdev</sgmltag>,
345 <sgmltag class="element">variance</sgmltag>,<sgmltag class="element">median</sgmltag>,
        <sgmltag class="element">mode</sgmltag>,<sgmltag class="element">moment</sgmltag>.
    </para></listitem>
    </varlistentry>
    <varlistentry>
350 <term>linear algebra</term>
    <listitem><para>
        <sgmltag class="element">vector</sgmltag>,<sgmltag class="element">matrix</sgmltag>,
        <sgmltag class="element">matrixrow</sgmltag>,<sgmltag class="element">determinant</sgmltag>,
        <sgmltag class="element">transpose</sgmltag>,<sgmltag class="element">selector</sgmltag>,
355 <sgmltag class="element">vectorproduct</sgmltag> (MathML 2.0),<sgmltag class="element">scalarproduct</sgmltag> (MathML 2.0),
        <sgmltag class="element">outerproduct</sgmltag> (MathML 2.0).
    </para></listitem>
    </varlistentry>
    <varlistentry>
360 <term>semantic mapping element</term>
    <listitem><para>
        <sgmltag class="element">annotation</sgmltag>,<sgmltag class="element">semantics</sgmltag>,
        <sgmltag class="element">annotation-xml</sgmltag>.
    </para></listitem>
365 </varlistentry>
    <varlistentry>
    <term>constant and symbol element (all MathML 2.0)</term>
    <listitem><para>
        <sgmltag class="element">integers</sgmltag>,<sgmltag class="element">reals</sgmltag>,
370 <sgmltag class="element">rational</sgmltag>,<sgmltag class="element">naturalnumbers</sgmltag>,
        <sgmltag class="element">complexes</sgmltag>,<sgmltag class="element">primes</sgmltag>,
        <sgmltag class="element">exponentiale</sgmltag>,<sgmltag class="element">imaginaryi</sgmltag>,
        <sgmltag class="element">notanumber</sgmltag>,<sgmltag class="element">>true</sgmltag>,
        <sgmltag class="element">>false</sgmltag>,<sgmltag class="element">emptyset</sgmltag>,
375 <sgmltag class="element">pi</sgmltag>,<sgmltag class="element">eulergamma</sgmltag>,

```

Appendix B. A complete example of a DocBook article

```
<sgmltag class="element">infinity</sgmltag>.  
</para></listitem>  
</varlistentry>  
</variablelist>  
380 </para>  
  
<sect2>  
<title>An example with content markup</title>  
  
385 <para>  
The matrix example given in the preceding section in its presentation markup  
form if recoded here using content markup.  
<programlisting>&lt;reln>  
    &lt;eq/>  
390    &lt;ci>A&lt;/ci>  
    &lt;matrix>  
        &lt;matrixrow>  
            &lt;ci>x&lt;/ci>&lt;ci>y&lt;/ci>  
        &lt;/matrixrow>  
395    &lt;matrixrow>  
        &lt;ci>z&lt;/ci>&lt;ci>w&lt;/ci>  
        &lt;/matrixrow>  
    &lt;/matrix>  
    &lt;/reln>  
400 </programlisting>  
</para>  
</sect2>  
</sect1>  
  
405 <bibliography>  
    <title>Bibliographic references</title>  
    <biblioentry id="bib.xml">  
        <abbrev id="bib.xml.abbrev">XML98</abbrev>  
        <authorgroup>  
410 <author><othername>World Wide Web Consortium</othername></author>  
        <editor><firstname>Tim</firstname><surname>Bray</surname></editor>  
        <editor><firstname>Jean</firstname><surname>Paoli</surname></editor>  
        <editor><firstname>Michael</firstname><surname>Sperberg-McQueen</surname></editor>  
        </authorgroup>  
415 <title>  
    <ulink url="http://www.w3.org/TR/REC-xml/">Extensible Markup Language  
    (XML) 1.0</ulink></title>  
    <bibliomisc>  
    See also the <ulink url="http://www.xml.com/axml/axml.html">annotated  
420 version of the XML specification</ulink>.</bibliomisc>  
    </biblioentry>  
    <biblioentry id="bib.TDG99">  
        <abbrev id="bib.TDG99.abbrev">TDG1999</abbrev>  
        <authorgroup>  
425 <author><firstname>Norman</firstname><surname>Walsh</surname></author>  
        <author><firstname>Leonard</firstname><surname>Muellner</surname></author>  
        </authorgroup>  
        <title>Docbook. The Definitive Guide.</title>  
        <publisher>  
430 <publishername>O'Reilly & Associates, Inc.</publishername>  
        </publisher>  
        <copyright><year>1999</year></copyright>  
        <isbn>1-56592-580-7</isbn>  
        <releaseinfo><ulink  
435 url= "http://www.oreilly.com/catalog/docbook/index.html">Oreilly's  
        catalog entry</ulink>.  
        </releaseinfo>  
    </bibliomisc>
```

```

You can also consult the <ulink url=
440 "http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html">
online version of the DocBook reference guide</ulink> and download the
<ulink url="http://sourceforge.net/projects/docbook/">Docbook DTD and XSL
stylesheets</ulink>.</bibliomisc>
</biblioentry>
445 <biblioentry id="bib.mathml">
<abbrev id="bib.mathml.abbrev">MATHML2</abbrev>
<authorgroup>
<author><othername>World Wide Web Consortium</othername></author>
<editor><firstname>David</firstname><surname>Carlisle</surname></editor>
450 <editor><firstname>Patrick</firstname><surname>Ion</surname></editor>
<editor><firstname>Robert</firstname><surname>Miner</surname></editor>
<editor><firstname>Nico</firstname><surname>Poppelier</surname></editor>
</authorgroup>

455 <title><ulink url="http://www.w3.org/TR/MathML2/">Mathematical
Markup Language (MathML) Version 2.0</ulink>
</title>
</biblioentry>
</bibliography>
460 </article>

```

The result of transforming the DocBook source of this appendix into PDF (with docbook2pdf, see Example 1.6 in Section 1.7) is shown in Figure B.1.

Figure B.1. DocBook example with MathML translated into PDF

A Docbook document featuring a few formulae

Michel Goossens
Sunday, 17 March 2002

This XML document is marked up according to the DocBook schema. It shows a few elements of the DocBook vocabulary, as well as a couple of examples of mathematical expressions where we used MathML markup.

1. The Docbook model

DocBook [TDG1999] proposes an XML model [XML98] for marking up technical documents. It is particularly well-suited for software reference guides and computer equipment manuals.

Docbook contains hundreds of elements to markup up clearly and explicitly the different components of an electronic document (book, article, reference guide, etc.), not only displaying its hierarchical structure but also indicating the semantical meaning of the various elements. The structure of the DTD is optimized to allow for customization, thus making it relatively straightforward to add or eliminate certain elements or attributes, to change the content model for certain structural groups, or to restrict the value that given attributes can take.

Norman Walsh developed a set of XSL stylesheets to transform XML documents marked up using the DocBook DTD into HTML or XSL formatting objects. The latter can be interpreted by PassiveTeX and xsltproc to obtain PDF or PostScript output.

2. MathML and mathematics

MathML [MATHML2] is a W3C recommendation whose aim is to encode mathematical material for teaching and scientific communication at all levels.

MathML consists of two parts: presentation (notation) et contents (meaning). MathML permits the conversion of mathematical information into various representations and output formats, including graphical displays, speech synthesizers, computer algebra programs, other mathematics typesetting languages, such as TeX, plain text, printers (PostScript), braille, etc.

MathML has support for efficiently browsing long and complex mathematical expressions and offers an extension mechanism. MathML code is human readable, easy to generate and process by software applications, and well suited for editing (even though MathML syntax might appear unnecessarily verbose and complex to the human reader).

The W3C MathML Working Group released the Second version of the the MathML Specification at the beginning of 2001. Two public initiatives that allow the display of MathML code directly and that are under active development are W3C's Amaya and Mozilla. Commercial programs, such as IBM's techexplorer (a plugin for Netscape or Microsoft's Internet Explorer) or Design Science Webeg (using the Java applet technology) can display MathML formulae in present day browsers. Several computer

algebra programs, e.g., Mathematica, or editors using e.g., mathtype, offer a user-friendly interface to enter, produce or read mathematics material marked up in MathML.

3. Presentation markup

The *presentation* part of MathML describes the spacial layout of the different elements of a mathematical expression. MathML presentation markup has about thirty elements, that form the basis of a mathematical syntax using classical visual layout model. Some fifty attributes provide precise control on the exact positioning of the various components of the math expression.

The list of presentation elements follows.

Token elements

m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14, m15, m16, m17, m18, m19, m20, m21, m22, m23, m24, m25, m26, m27, m28, m29, m30, m31, m32, m33, m34, m35, m36, m37, m38, m39, m40, m41, m42, m43, m44, m45, m46, m47, m48, m49, m50, m51, m52, m53, m54, m55, m56, m57, m58, m59, m60, m61, m62, m63, m64, m65, m66, m67, m68, m69, m70, m71, m72, m73, m74, m75, m76, m77, m78, m79, m80, m81, m82, m83, m84, m85, m86, m87, m88, m89, m90, m91, m92, m93, m94, m95, m96, m97, m98, m99, m100, m101, m102, m103, m104, m105, m106, m107, m108, m109, m110, m111, m112, m113, m114, m115, m116, m117, m118, m119, m120, m121, m122, m123, m124, m125, m126, m127, m128, m129, m130, m131, m132, m133, m134, m135, m136, m137, m138, m139, m140, m141, m142, m143, m144, m145, m146, m147, m148, m149, m150, m151, m152, m153, m154, m155, m156, m157, m158, m159, m160, m161, m162, m163, m164, m165, m166, m167, m168, m169, m170, m171, m172, m173, m174, m175, m176, m177, m178, m179, m180, m181, m182, m183, m184, m185, m186, m187, m188, m189, m190, m191, m192, m193, m194, m195, m196, m197, m198, m199, m200, m201, m202, m203, m204, m205, m206, m207, m208, m209, m210, m211, m212, m213, m214, m215, m216, m217, m218, m219, m220, m221, m222, m223, m224, m225, m226, m227, m228, m229, m230, m231, m232, m233, m234, m235, m236, m237, m238, m239, m240, m241, m242, m243, m244, m245, m246, m247, m248, m249, m250, m251, m252, m253, m254, m255, m256, m257, m258, m259, m260, m261, m262, m263, m264, m265, m266, m267, m268, m269, m270, m271, m272, m273, m274, m275, m276, m277, m278, m279, m280, m281, m282, m283, m284, m285, m286, m287, m288, m289, m290, m291, m292, m293, m294, m295, m296, m297, m298, m299, m300, m301, m302, m303, m304, m305, m306, m307, m308, m309, m310, m311, m312, m313, m314, m315, m316, m317, m318, m319, m320, m321, m322, m323, m324, m325, m326, m327, m328, m329, m330, m331, m332, m333, m334, m335, m336, m337, m338, m339, m340, m341, m342, m343, m344, m345, m346, m347, m348, m349, m350, m351, m352, m353, m354, m355, m356, m357, m358, m359, m360, m361, m362, m363, m364, m365, m366, m367, m368, m369, m370, m371, m372, m373, m374, m375, m376, m377, m378, m379, m380, m381, m382, m383, m384, m385, m386, m387, m388, m389, m390, m391, m392, m393, m394, m395, m396, m397, m398, m399, m400, m401, m402, m403, m404, m405, m406, m407, m408, m409, m410, m411, m412, m413, m414, m415, m416, m417, m418, m419, m420, m421, m422, m423, m424, m425, m426, m427, m428, m429, m430, m431, m432, m433, m434, m435, m436, m437, m438, m439, m440, m441, m442, m443, m444, m445, m446, m447, m448, m449, m450, m451, m452, m453, m454, m455, m456, m457, m458, m459, m460, m461, m462, m463, m464, m465, m466, m467, m468, m469, m470, m471, m472, m473, m474, m475, m476, m477, m478, m479, m480, m481, m482, m483, m484, m485, m486, m487, m488, m489, m490, m491, m492, m493, m494, m495, m496, m497, m498, m499, m500, m501, m502, m503, m504, m505, m506, m507, m508, m509, m510, m511, m512, m513, m514, m515, m516, m517, m518, m519, m520, m521, m522, m523, m524, m525, m526, m527, m528, m529, m530, m531, m532, m533, m534, m535, m536, m537, m538, m539, m540, m541, m542, m543, m544, m545, m546, m547, m548, m549, m550, m551, m552, m553, m554, m555, m556, m557, m558, m559, m560, m561, m562, m563, m564, m565, m566, m567, m568, m569, m570, m571, m572, m573, m574, m575, m576, m577, m578, m579, m580, m581, m582, m583, m584, m585, m586, m587, m588, m589, m590, m591, m592, m593, m594, m595, m596, m597, m598, m599, m600, m601, m602, m603, m604, m605, m606, m607, m608, m609, m610, m611, m612, m613, m614, m615, m616, m617, m618, m619, m620, m621, m622, m623, m624, m625, m626, m627, m628, m629, m630, m631, m632, m633, m634, m635, m636, m637, m638, m639, m640, m641, m642, m643, m644, m645, m646, m647, m648, m649, m650, m651, m652, m653, m654, m655, m656, m657, m658, m659, m660, m661, m662, m663, m664, m665, m666, m667, m668, m669, m670, m671, m672, m673, m674, m675, m676, m677, m678, m679, m680, m681, m682, m683, m684, m685, m686, m687, m688, m689, m690, m691, m692, m693, m694, m695, m696, m697, m698, m699, m700, m701, m702, m703, m704, m705, m706, m707, m708, m709, m710, m711, m712, m713, m714, m715, m716, m717, m718, m719, m720, m721, m722, m723, m724, m725, m726, m727, m728, m729, m730, m731, m732, m733, m734, m735, m736, m737, m738, m739, m740, m741, m742, m743, m744, m745, m746, m747, m748, m749, m750, m751, m752, m753, m754, m755, m756, m757, m758, m759, m760, m761, m762, m763, m764, m765, m766, m767, m768, m769, m770, m771, m772, m773, m774, m775, m776, m777, m778, m779, m780, m781, m782, m783, m784, m785, m786, m787, m788, m789, m790, m791, m792, m793, m794, m795, m796, m797, m798, m799, m800, m801, m802, m803, m804, m805, m806, m807, m808, m809, m810, m811, m812, m813, m814, m815, m816, m817, m818, m819, m820, m821, m822, m823, m824, m825, m826, m827, m828, m829, m830, m831, m832, m833, m834, m835, m836, m837, m838, m839, m840, m841, m842, m843, m844, m845, m846, m847, m848, m849, m850, m851, m852, m853, m854, m855, m856, m857, m858, m859, m860, m861, m862, m863, m864, m865, m866, m867, m868, m869, m870, m871, m872, m873, m874, m875, m876, m877, m878, m879, m880, m881, m882, m883, m884, m885, m886, m887, m888, m889, m890, m891, m892, m893, m894, m895, m896, m897, m898, m899, m900, m901, m902, m903, m904, m905, m906, m907, m908, m909, m910, m911, m912, m913, m914, m915, m916, m917, m918, m919, m920, m921, m922, m923, m924, m925, m926, m927, m928, m929, m930, m931, m932, m933, m934, m935, m936, m937, m938, m939, m940, m941, m942, m943, m944, m945, m946, m947, m948, m949, m950, m951, m952, m953, m954, m955, m956, m957, m958, m959, m960, m961, m962, m963, m964, m965, m966, m967, m968, m969, m970, m971, m972, m973, m974, m975, m976, m977, m978, m979, m980, m981, m982, m983, m984, m985, m986, m987, m988, m989, m990, m991, m992, m993, m994, m995, m996, m997, m998, m999, m1000, m1001, m1002, m1003, m1004, m1005, m1006, m1007, m1008, m1009, m1010, m1011, m1012, m1013, m1014, m1015, m1016, m1017, m1018, m1019, m1020, m1021, m1022, m1023, m1024, m1025, m1026, m1027, m1028, m1029, m1030, m1031, m1032, m1033, m1034, m1035, m1036, m1037, m1038, m1039, m1040, m1041, m1042, m1043, m1044, m1045, m1046, m1047, m1048, m1049, m1050, m1051, m1052, m1053, m1054, m1055, m1056, m1057, m1058, m1059, m1060, m1061, m1062, m1063, m1064, m1065, m1066, m1067, m1068, m1069, m1070, m1071, m1072, m1073, m1074, m1075, m1076, m1077, m1078, m1079, m1080, m1081, m1082, m1083, m1084, m1085, m1086, m1087, m1088, m1089, m1090, m1091, m1092, m1093, m1094, m1095, m1096, m1097, m1098, m1099, m1100, m1101, m1102, m1103, m1104, m1105, m1106, m1107, m1108, m1109, m1110, m1111, m1112, m1113, m1114, m1115, m1116, m1117, m1118, m1119, m1120, m1121, m1122, m1123, m1124, m1125, m1126, m1127, m1128, m1129, m1130, m1131, m1132, m1133, m1134, m1135, m1136, m1137, m1138, m1139, m1140, m1141, m1142, m1143, m1144, m1145, m1146, m1147, m1148, m1149, m1150, m1151, m1152, m1153, m1154, m1155, m1156, m1157, m1158, m1159, m1160, m1161, m1162, m1163, m1164, m1165, m1166, m1167, m1168, m1169, m1170, m1171, m1172, m1173, m1174, m1175, m1176, m1177, m1178, m1179, m1180, m1181, m1182, m1183, m1184, m1185, m1186, m1187, m1188, m1189, m1190, m1191, m1192, m1193, m1194, m1195, m1196, m1197, m1198, m1199, m1200, m1201, m1202, m1203, m1204, m1205, m1206, m1207, m1208, m1209, m1210, m1211, m1212, m1213, m1214, m1215, m1216, m1217, m1218, m1219, m1220, m1221, m1222, m1223, m1224, m1225, m1226, m1227, m1228, m1229, m1230, m1231, m1232, m1233, m1234, m1235, m1236, m1237, m1238, m1239, m1240, m1241, m1242, m1243, m1244, m1245, m1246, m1247, m1248, m1249, m1250, m1251, m1252, m1253, m1254, m1255, m1256, m1257, m1258, m1259, m1260, m1261, m1262, m1263, m1264, m1265, m1266, m1267, m1268, m1269, m1270, m1271, m1272, m1273, m1274, m1275, m1276, m1277, m1278, m1279, m1280, m1281, m1282, m1283, m1284, m1285, m1286, m1287, m1288, m1289, m1290, m1291, m1292, m1293, m1294, m1295, m1296, m1297, m1298, m1299, m1300, m1301, m1302, m1303, m1304, m1305, m1306, m1307, m1308, m1309, m1310, m1311, m1312, m1313, m1314, m1315, m1316, m1317, m1318, m1319, m1320, m1321, m1322, m1323, m1324, m1325, m1326, m1327, m1328, m1329, m1330, m1331, m1332, m1333, m1334, m1335, m1336, m1337, m1338, m1339, m1340, m1341, m1342, m1343, m1344, m1345, m1346, m1347, m1348, m1349, m1350, m1351, m1352, m1353, m1354, m1355, m1356, m1357, m1358, m1359, m1360, m1361, m1362, m1363, m1364, m1365, m1366, m1367, m1368, m1369, m1370, m1371, m1372, m1373, m1374, m1375, m1376, m1377, m1378, m1379, m1380, m1381, m1382, m1383, m1384, m1385, m1386, m1387, m1388, m1389, m1390, m1391, m1392, m1393, m1394, m1395, m1396, m1397, m1398, m1399, m1400, m1401, m1402, m1403, m1404, m1405, m1406, m1407, m1408, m1409, m1410, m1411, m1412, m1413, m1414, m1415, m1416, m1417, m1418, m1419, m1420, m1421, m1422, m1423, m1424, m1425, m1426, m1427, m1428, m1429, m1430, m1431, m1432, m1433, m1434, m1435, m1436, m1437, m1438, m1439, m1440, m1441, m1442, m1443, m1444, m1445, m1446, m1447, m1448, m1449, m1450, m1451, m1452, m1453, m1454, m1455, m1456, m1457, m1458, m1459, m1460, m1461, m1462, m1463, m1464, m1465, m1466, m1467, m1468, m1469, m1470, m1471, m1472, m1473, m1474, m1475, m1476, m1477, m1478, m1479, m1480, m1481, m1482, m1483, m1484, m1485, m1486, m1487, m1488, m1489, m1490, m1491, m1492, m1493, m1494, m1495, m1496, m1497, m1498, m1499, m1500, m1501, m1502, m1503, m1504, m1505, m1506, m1507, m1508, m1509, m1510, m1511, m1512, m1513, m1514, m1515, m1516, m1517, m1518, m1519, m1520, m1521, m1522, m1523, m1524, m1525, m1526, m1527, m1528, m1529, m1530, m1531, m1532, m1533, m1534, m1535, m1536, m1537, m1538, m1539, m1540, m1541, m1542, m1543, m1544, m1545, m1546, m1547, m1548, m1549, m1550, m1551, m1552, m1553, m1554, m1555, m1556, m1557, m1558, m1559, m1560, m1561, m1562, m1563, m1564, m1565, m1566, m1567, m1568, m1569, m1570, m1571, m1572, m1573, m1574, m1575, m1576, m1577, m1578, m1579, m1580, m1581, m1582, m1583, m1584, m1585, m1586, m1587, m1588, m1589, m1590, m1591, m1592, m1593, m1594, m1595, m1596, m1597, m1598, m1599, m1600, m1601, m1602, m1603, m1604, m1605, m1606, m1607, m1608, m1609, m1610, m1611, m1612, m1613, m1614, m1615, m1616, m1617, m1618, m1619, m1620, m1621, m1622, m1623, m1624, m1625, m1626, m1627, m1628, m1629, m1630, m1631, m1632, m1633, m1634, m1635, m1636, m1637, m1638, m1639, m1640, m1641, m1642, m1643, m1644, m1645, m1646, m1647, m1648, m1649, m1650, m1651, m1652, m1653, m1654, m1655, m1656, m1657, m1658, m1659, m1660, m1661, m1662, m1663, m1664, m1665, m1666, m1667, m1668, m1669, m1670, m1671, m1672, m1673, m1674, m1675, m1676, m1677, m1678, m1679, m1680, m1681, m1682, m1683, m1684, m1685, m1686, m1687, m1688, m1689, m1690, m1691, m1692, m1693, m1694, m1695, m1696, m1697, m1698, m1699, m1700, m1701, m1702, m1703, m1704, m1705, m1706, m1707, m1708, m1709, m1710, m1711, m1712, m1713, m1714, m1715, m1716, m1717, m1718, m1719, m1720, m1721, m1722, m1723, m1724, m1725, m1726, m1727, m1728, m1729, m1730, m1731, m1732, m1733, m1734, m1735, m1736, m1737, m1738, m1739, m1740, m1741, m1742, m1743, m1744, m1745, m1746, m1747, m1748, m1749, m1750, m1751, m1752, m1753, m1754, m1755, m1756, m1757, m1758, m1759, m1760, m1761, m1762, m1763, m1764, m1765, m1766, m1767, m1768, m1769, m1770, m1771, m1772, m1773, m1774, m1775, m1776, m1777, m1778, m1779, m1780, m1781, m1782, m1783, m1784, m1785, m1786, m1787, m1788, m1789, m1790, m1791, m1792, m1793, m1794, m1795, m1796, m1797, m1798, m1799, m1800, m1801, m1802, m1803, m1804, m1805, m1806, m1807, m1808, m1809, m1810, m1811, m1812, m1813, m1814, m1815, m1816, m1817, m1818, m1819, m1820, m1821, m1822, m1823, m1824, m1825, m1826, m1827, m1828, m1829, m1830, m1831, m1832, m1833, m1834, m1835, m1836, m1837, m1838, m1839, m1840, m1841, m1842, m1843, m1844, m1845, m1846, m1847, m1848, m1849, m1850, m1851, m1852, m1853, m1854, m1855, m1856, m1857, m1858, m1859, m1860, m1861, m1862, m1863, m1864, m1865, m1866, m1867, m1868, m1869, m1870, m1871, m1872, m1873, m1874, m1875, m1876, m1877, m1878, m1879, m1880, m1881, m1882, m1883, m1884, m1885, m1886, m1887, m1888, m1889, m1890, m1891, m1892, m1893, m1894, m1895, m1896, m1897, m1898, m1899, m1900, m1901, m1902, m1903, m1904, m1905, m1906, m1907, m1908, m1909, m1910, m1911, m1912, m1913, m1914, m1915, m1916, m1917, m1918, m1919, m1920, m1921, m1922, m1923, m1924, m1925, m1926, m1927, m1928, m1929

It is impossible to cover all of mathematics, so MathML proposes only a relatively small number of commonplace mathematical constructs, chosen carefully to be sufficient in a large number of applications. In addition, it provides a *extension mechanism* for associating semantics with new notational constructs, thus allowing these to be encoded even when they are not in MathML content element base collection.

MathML's basic set of content elements was chosen to allow for simple coding of most of the formulas used in secondary education, through the first year of university. The subject areas targeted are arithmetic, algebra, logic and relations, calculus and vector calculus, set theory, sequences and series, elementary classical functions, and statistics linear algebra. Using this basic sets more complex constructs can be built.

The list of the content elements of MathML follows.

token elements
cn, ci, csymbol (MathML 2.0).

basic content elements
apply, rein (deprecated), fn (deprecated for externally defined functions), interval, inverse, sep, condition, declare, lambda, compose, ident.

arithmetic, algebra and logic
quotient, exp, factorial, divide, max and min, minus, plus, power, rem, times, root, gcd, and, or, xor, not, implies, forall, exists, abs, conjugate, arg (MathML 2.0), real (MathML 2.0), imaginary (MathML 2.0), lcm (MathML 2.0).

relations
eq, neq, gt, lt, geq, leq, equivalent (MathML 2.0), approx (MathML 2.0).

calculus and vector calculus
int, diff, partialdiff, levlimit, uplimit, bvar, degree, divergence (MathML 2.0), grad (MathML 2.0), curl (MathML 2.0), laplacian (MathML 2.0).

theory of sets
set, list, union, intersect, in, notin, subset, prsubset, notsubset, notprsubset, setdiff, card (MathML 2.0).

sequences and series
sum, product, limit, tendsto.

elementary classical function
exp, ln, log, sin, cos, tan, sec, csc, cot, sinh, cosh, tanh, sech, csch, coth, arcsin, arccos, arctan, arccosh, arccot, arccoth, arccsc, arccsch, arcsec, arcsech, arcsinh, arctanh.

statistics
mean, sdev, variance, median, mode, moment.

linear algebra
vector, matrix, matrixrow, determinant, transpose, selector, vectorproduct (MathML 2.0), scalarproduct (MathML 2.0), outerproduct (MathML 2.0).

semantic mapping element
annotation, semantics, annotation-xml.

constant and symbol element (all MathML 2.0)
integers, reals, rationals, naturalnumbers, complexes, primes, exponentiale, imaginary, notanumber, true, false, emptyset, pi, eulergamma, infinity.

4.1. An example with content markup

The matrix example given in the preceding section in its presentation markup form if recoded here using content markup.

```
<rela>
  <eq>
    <ci>A</ci>
    <matrix>
      <matrixrow>
        <ci>x</ci><ci>y</ci></ci>
      </matrixrow>
      <matrixrow>
        <ci>z</ci><ci>w</ci></ci>
      </matrixrow>
    </matrix>
  </rela>
```

Bibliographic references

- [XML98] World Wide Web Consortium, Tim Bray, Jean Paoli, and Michael Sperberg-McQueen. *Extensible Markup Language (XML) 1.0* [<http://www.w3.org/TR/REC-xml/>]. See also the annotated version of the XML specification [<http://www.xml.com/axml/axml.html>].
- [TDG1999] Norman Walsh and Leonard Blueliner. *Docbook. The Definitive Guide*. O'Reilly & Associates, Inc. Copyright © 1999. 1-56592-580-7. O'Reilly's catalog entry [<http://www.oreilly.com/catalog/docbook/index.html>]. You can also consult the online version of the DocBook reference guide [<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>] and download the Docbook DTD and XSL stylesheets [<http://sourceforge.net/projects/docbook/>].
- [MATHML2] World Wide Web Consortium, David Carlisle, Patrick Ion, Robert Miner, and Nico Poppelier. *Mathematical Markup Language (MathML) Version 2.0* [<http://www.w3.org/TR/MathML2/>].

We can also transform the DocBook source into HTML with docbook2html, as explained in Section 1.5. On the one hand, we can convert it into a single output file (as in Example 1.2), and the result is shown in Figure B.2 as displayed with the Mozilla browser. This figure corresponds to the beginning of the output file. On the other hand, Figure B.3 shows one of the chunks generated when running docbook2html with the chunk option (see Example 1.4 for an explanation). For large files it is more efficient to use chunking to allow handling of reasonably-sized output files that have to be served over the Web.

Figure B.2. DocBook example with MathML translated into HTML and viewed with Mozilla (one file)

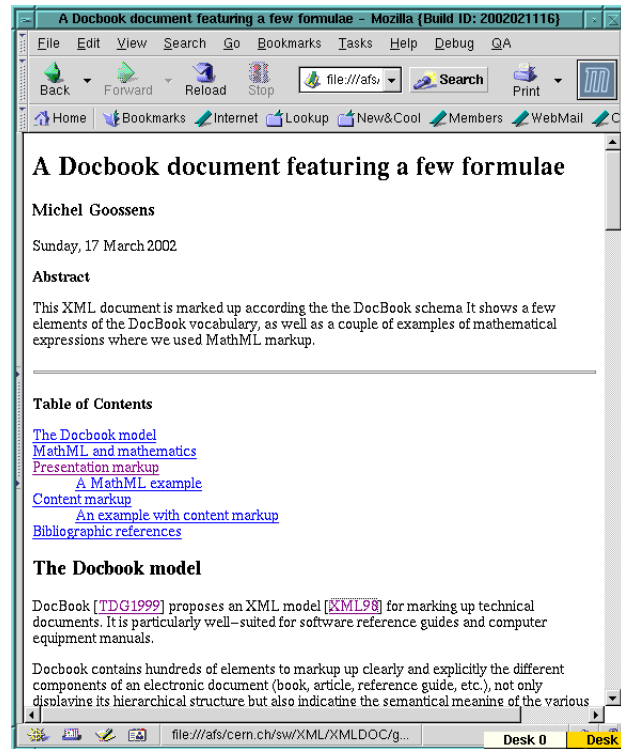
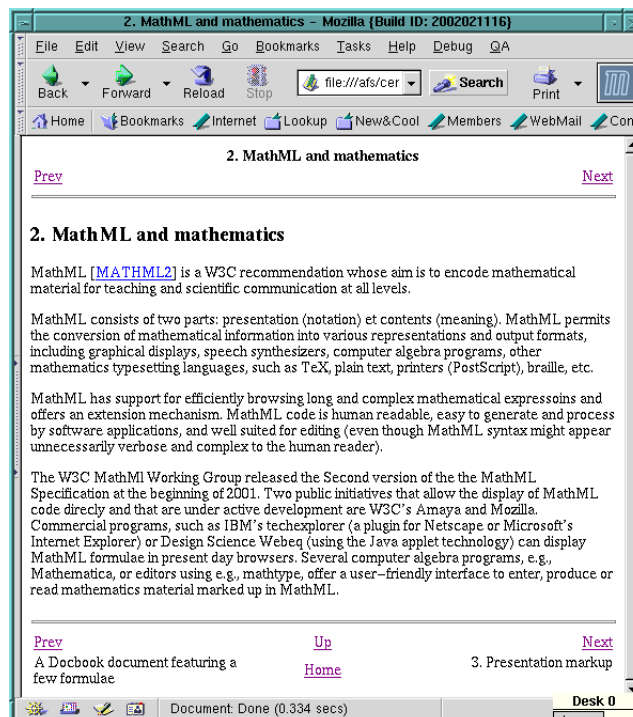


Figure B.3. DocBook example with MathML translated into HTML and viewed with Mozilla (chunked file)



Glossary

ASCII	(American Standard Code for Information Interchange) This standard character encoding scheme is used extensively in data transmission.
ANSI	(American National Standards Institute) This group is the U.S. member organization that belongs to the ISO, the International Organization for Standardization.
attribute	Attributes augment the element on which they appear; they also provide additional information about the element. Attributes appear as name-value pairs in the element's start-tag. For example, to assign the value <code>hostname</code> to the <code>role</code> attribute of <code>systemitem</code> , you would use the mark up: <code><systemitem role="hostname"></code> .
callout	A pointer, verbal or graphical or both, to a <i>component</i> of an illustration or a text object.
cooked	“Cooked” data, as distinct from “raw,” is a collection of elements and character data that's ready for presentation. The processor is not expected to rearrange, select, or suppress any of the elements, but simply present them as specified. See Also Raw.
document type declaration (DTD)	A set of declarations that defines the names of the elements and their attributes, and that specifies rules for their combination or sequence. You can store a DTD at the beginning of a document or externally in a separate file.
element	Elements define the hierarchical structure of a document. that may They contain either text or other subelements such as a paragraph, a chapter, and so on. In XML all elements have start and end tags and contain some part of the document content. Empty elements have no content.
element declaration	A statement in the DTD defining an element and declaring the order in which it may appear in the document and what other elements it may include.
entity	A name assigned (by means of a declaration) to some chunk of data so it can be referred to by that name; the data can be of various kinds: a string of characters, a special character (e.g., unavailable on a standard keyboard), a external chapter or graphics file, or a set of declarations in a DTD, for instance. The way an entity is referred to depends on the type of data and where it is being referenced. XML has parameter, external, internal, and data entities.
entity declaration	A statement in the DTD or document that assigns an XML/L name to an entity so you can reference it.
external entity	An external entity is a general entity that refers to another document. External entities are often used to incorporate parsable text documents, like legal notices or chapters, into larger units, like chapters or books.

external subset	Element, attribute, and other declarations that compose (part of) a document type definition that are stored in an external entity, and referenced from a document's document type declaration using a public or system identifier.
float	Text objects like sidebars, figures, tables, and graphics are said to float when their actual place in the document is not fixed. For presentation on a printed page, for instance, a graphic may float to the top of the next page if it is too tall to fit on the page in which it actually falls, in the sequence of words and the sequence of other like objects in a document.
general entity	An entity referenced by a name that starts with an ampersand (&) and ends with a semicolon. Most of the time general entities are used in SGML/XML documents, not in the DTD. There are two types, external and internal entities, and they refer either to special characters or to text objects like commonly repeated phrases or names or chapters.
HTML	(HyperText Markup Language) This is the format of files published on the World Wide Web. HTML is an application of SGML/XML; to author in HTML using SGML/XML-based authoring software, you simply need the HTML DTD.
internal entity	A general entity that references a piece of text (including its markup and even other internal entities), usually as a keyboard shortcut.
internal subset	Element, attribute, and other declarations that compose (part of) a document type definition that are stored in a document, within the document type declaration.
Internet	The Internet is a worldwide communications network originally developed by the U.S. Department of Defense as a distributed system with no single point of failure. The Internet has seen an explosion in commercial use since the development of easy-to-use software for accessing the Internet.
ISO	(International Organization for Standardization) The ISO is an industry-supported organization that establishes worldwide standards for everything from data interchange formats to film speed specifications.
markup	Markup is anything added to the content of the document that describes the text.
meta-information	Meta-information is information about a document, such as the specification of its author or its date of composition, as opposed to the content of a document itself.
parameter entity	An entity usually referenced in the DTD by a name that starts with a percent sign (%) and ends with a semicolon. In DocBook, parameter entities are mainly used to facilitate customization of the DTD, but they can also be used to control marked sections of a document.
parser	A parser is a specialized software program that recognizes XML markup in a document. A parser that reads a DTD and checks and reports on markup errors is a validating XML parser. A parser can be built into an XML editor to prevent incorrect tagging and to check whether a document contains all the required elements.

processing instruction	<p>An essentially arbitrary string preceded by a question mark and delimited by angle brackets that is intended to convey information to an application that processes an XML instance. For example, the processing instruction <code><?linebreak></code> might cause the formatter to introduce a line break at the position where the processing instruction occurs.</p> <p>In XML documents, processing instructions should have the form:</p> <pre><?<i>pitarget</i> param1="value1" param2="value2"?></pre> <p>The <i>pitarget</i> should be a name that the processing application will recognize. Additional information in the PI should be added using “attribute syntax.”</p>
public identifier	An abstract identifier for an SGML or XML document, DTD, or external entity.
raw	<p>“Raw” data is just a collection of elements, with no additional punctuation or information about presentation. To continue the cooking metaphor, raw data is just a set of ingredients. It’s up to the processor to select appropriate elements, arrange them for display, and add required presentational information.</p> <p>See Also Cooked.</p>
SGML	Standard Generalized Markup Language, an international standard (ISO 8879) that specifies the rules for the creation of platform-independent markup languages for electronic texts.
stylesheet	A file that specifies the presentation or appearance of a document; there are several standards for such stylesheets, including CSS, FOSIs, DSSSL, and, most recently, XSL. Vendors often have proprietary stylesheet formats as well.
system identifier	<p>In SGML/XML, a local, system-dependent identifier for a document, DTD, or external entity. Usually a filename on the local system.</p> <p>In XML, a system identifier is required to be a URI.</p>
tag	<p>In the world of SGML/XML, a tag is a marker embedded in a document that indicates the purpose or function of the element. Each element has a beginning tag and an end tag, or is empty. Its syntax is a name enclosed in angle brackets (<code><></code>). For instance, <code><para></code> is a tag in DocBook used to mark the beginning of a paragraph, <code></para></code> marks the end of a paragraph, while <code><xref . . . /></code> is an empty tag.</p>
Unicode	<p>The Unicode Consortium [http://www.unicode.org] is a non-profit organization founded to develop, extend and promote use of the Unicode Standard, which specifies the representation of multi-lingual text in modern software products and standards.</p> <p>Unicode supports characters that are one to four bytes wide rather than the 8-bit codes currently supported by most systems. In each of its seventeen 16-bit planes (numbered 0 to 16) Unicode encodes 65,536 characters rather than only 256 for one-byte encodings, such as ASCII, ISO-8859-1, etc. Thus, in total, Unicode allows one to encode well over 1 million characters.</p> <p>In its Basic Multilingual Plane (Plane 0, BMP) it encodes most of the world’s commonly-used languages, with rarer, ancient, or specialized characters encoded in the higher planes. The first 128 code points</p>

coincide with ASCII, so that for text using that character set only one byte is needed, whereas for most other languages two bytes suffice.

Of particular interest are the XML standard encodings *UTF-8* (Unicode Transformation Format, 8-bit encoding form), which serializes a Unicode scalar value (code point) as a sequence of one to four bytes, and *UTF-16* (Unicode Transformation Format, 16-bit encoding form), which serializes a Unicode scalar value (code point) as a sequence of two bytes, in either big-endian or little-endian format.

URI	Uniform Resource Identifier, the W3C's codification of the name and address syntax of present and future objects on the Internet. In its most basic form, a URI consists of a scheme name (such as file, http, ftp, news, mailto, gopher) followed by a colon, followed by a path whose nature is determined by the scheme that precedes it (see RFC 1630). URI is the umbrella term for URNs, URLs, and all other Uniform Resource Identifiers.
URL	Uniform Resource Locator, a name and address for an existing object accessible over the Internet. http://www.docbook.org is an example of a URL (see RFC 1738).
URN	Uniform Resource Name, the result of an evolving attempt to define a name and address syntax for <i>persistent</i> objects accessible over the Internet; <code>urn:foo:a123,456</code> is a legal URN consisting of three colon-separated fields: urn followed by a namespace identifier, followed by a namespace specifier (see RFC 1737 and RFC 2141 for details).
W3C	The World Wide Web Consortium (http://www.w3.org/).
World Wide Web	Often referred to as WWW or the Web, this usually refers to information available on the Internet that can be easily accessed with software usually called a "browser." Organizations publish their information on the Web in a format known as HTML (or more recently in XML with an accompanying CSS or XSL stylesheet). This information is usually referred to as their "home page" or "web site".
wrapper	Some elements, such as <code>chapter</code> , have important semantic significance. Other elements serve no obvious purpose except to contain a number of other elements. For example, <code>bookinfo</code> has no important semantics; it merely serves as a container for the meta-information about a book. Elements that are just containers are sometimes called "wrappers."
XML	The Extensible Markup Language [http://www.w3.org/TR/REC-xml], a subset of SGML designed specifically for use over the Web.
XSL	XML Style Language, an evolving language for stylesheets to be attached to XML documents. The stylesheet is itself an XML document.

Appendix C. Overview of all DocBook elements

The present appendix list alphabetically all elements that are defined in version 4 of the DocBook DTD. For each element a one line description is given. For more details the [DocBook Reference Guide] [<http://xml.cern.ch/www.docbook.org/tdg/en/html/docbook.html>] should be consulted.

- **abbrev** - An abbreviation, especially one followed by a period
- **abstract** - A summary
- **accel** - A graphical user interface (GUI) keyboard shortcut
- **ackno** - Acknowledgements in an Article
- **acronym** - An often pronounceable word made from the initial (or selected) letters of a name or phrase
- **action** - A response to a user event
- **address** - A real-world address, generally a postal address
- **affiliation** - The institutional affiliation of an individual
- **alt** - Text representation for a graphical element
- **anchor** - A spot in the document
- **answer** - An answer to a question posed in a QandASet
- **appendix** - An appendix in a Book or Article
- **appendixinfo** - Meta-information for an Appendix
- **application** - The name of a software program
- **area** - A region defined for a Callout in a graphic or code example
- **areaset** - A set of related areas in a graphic or code example
- **areaspec** - A collection of regions in a graphic or code example
- **arg** - An argument in a CmdSynopsis
- **article** - An article
- **articleinfo** - Meta-information for an Article
- **artpagenums** - The page numbers of an article as published
- **attribution** - The source of a block quote or epigraph
- **audiodata** - Pointer to external audio data
- **audioobject** - A wrapper for audio data and its associated meta-information
- **author** - The name of an individual author
- **authorblurb** - A short description or note about an author

- **authorgroup** - Wrapper for author information when a document has multiple authors or collaborators
- **authorinitials** - The initials or other short identifier for an author
- **beginpage** - The location of a page break in a print version of the document
- **bibliodiv** - A section of a Bibliography
- **biblioentry** - An entry in a Bibliography
- **bibliography** - A bibliography
- **bibliographyinfo** - Meta-information for a Bibliography
- **bibliomisc** - Untyped bibliographic information
- **bibliomixed** - An entry in a Bibliography
- **bibliomset** - A cooked container for related bibliographic information
- **biblioset** - A raw container for related bibliographic information
- **blockquote** - A quotation set off from the main text
- **book** - A book
- **bookinfo** - Meta-information for a Book
- **bridgehead** - A free-floating heading
- **callout** - A “called out” description of a marked Area
- **calloutlist** - A list of Callouts
- **caption** - A caption
- **caution** - A note of caution
- **chapter** - A chapter, as of a book
- **chapterinfo** - Meta-information for a Chapter
- **citation** - An inline bibliographic reference to another published work
- **citerefentry** - A citation to a reference page
- **citetitle** - The title of a cited work
- **city** - The name of a city in an address
- **classname** - The name of a class, in the object-oriented programming sense
- **classsynopsis** - The syntax summary for a class definition
- **classsynopsisinfo** - Information supplementing the contents of a ClassSynopsis
- **cmdsynopsis** - A syntax summary for a software command
- **co** - The location of a callout embedded in text
- **collab** - Identifies a collaborator

Appendix C. Overview of all DocBook elements

- `collabname` - The name of a collaborator
- `colophon` - Text at the back of a book describing facts about its production
- `colspec` - Specifications for a column in a table
- `command` - The name of an executable program or other software command
- `computeroutput` - Data, generally text, displayed or presented by a computer
- `confdates` - The dates of a conference for which a document was written
- `confgroup` - A wrapper for document meta-information about a conference
- `confnum` - An identifier, frequently numerical, associated with a conference for which a document was written
- `confsponsor` - The sponsor of a conference for which a document was written
- `conftitle` - The title of a conference for which a document was written
- `constant` - A programming or system constant
- `constraint` (EBNF) - A constraint in an EBNF production
- `constraintdef` (EBNF) - The definition of a constraint in an EBNF production
- `constructorsynopsis` - A syntax summary for a constructor
- `contractnum` - The contract number of a document
- `contractsponsor` - The sponsor of a contract
- `contrib` - A summary of the contributions made to a document by a credited source
- `copyright` - Copyright information about a document
- `corpauthor` - A corporate author, as opposed to an individual
- `corpname` - The name of a corporation
- `country` - The name of a country
- `database` - The name of a database, or part of a database
- `date` - The date of publication or revision of a document
- `dedication` - A wrapper for the dedication section of a book
- `destructorsynopsis` - A syntax summary for a destructor
- `edition` - The name or number of an edition of a document
- `editor` - The name of the editor of a document
- `email` - An email address
- `Emphasis` - Emphasized text
- `entry` - A cell in a table
- `entrytbl` - A subtable appearing in place of an Entry in a table

- `envar` - A software environment variable
- `epigraph` - A short inscription at the beginning of a document or component
- `equation` - A displayed mathematical equation
- `errorcode` - An error code
- `errorname` - An error name
- `errortype` - The classification of an error message
- `example` - A formal example, with a title
- `exceptionname` - The name of an exception
- `fax` - A fax number
- `fieldsynopsis` - The name of a field in a class definition
- `figure` - A formal figure, generally an illustration, with a title
- `filename` - The name of a file
- `firstname` - The first name of a person
- `firstterm` - The first occurrence of a term
- `footnote` - A footnote
- `footnoteref` - A cross reference to a footnote (a footnote mark)
- `foreignphrase` - A word or phrase in a language other than the primary language of the document
- `formalpara` - A paragraph with a title
- `funcdef` - A function (subroutine) name and its return type
- `funcparams` - Parameters for a function referenced through a function pointer in a synopsis
- `funcprototype` - The prototype of a function
- `funcsynopsis` - The syntax summary for a function definition
- `funcsynopsisinfo` - Information supplementing the FuncDefs of a FuncSynopsis
- `function` - The name of a function or subroutine, as in a programming language
- `glossary` - A glossary
- `glossaryinfo` - Meta-information for a Glossary
- `glossdef` - A definition in a GlossEntry
- `glossdiv` - A division in a Glossary
- `glossentry` - An entry in a Glossary or GlossList
- `glosslist` - A wrapper for a set of GlossEntrys
- `glosssee` - A cross-reference from one GlossEntry to another

Appendix C. Overview of all DocBook elements

- `glossseealso` - A cross-reference from one GlossEntry to another
- `glossterm` - A glossary term
- `graphic` - A displayed graphical object (not an inline)
- `graphicco` - A graphic that contains callout areas
- `group` - A group of elements in a CmdSynopsis
- `guibutton` - The text on a button in a GUI
- `guiicon` - Graphic and/or text appearing as a icon in a GUI
- `guilabel` - The text of a label in a GUI
- `guimenu` - The name of a menu in a GUI
- `guimenuitem` - The name of a terminal menu item in a GUI
- `guisubmenu` - The name of a submenu in a GUI
- `hardware` - A physical part of a computer system
- `highlights` - A summary of the main points of the discussed component
- `holder` - The name of the individual or organization that holds a copyright
- `honorific` - The title of a person
- `imagedata` - Pointer to external image data
- `imageobject` - A wrapper for image data and its associated meta-information
- `imageobjectco` - A wrapper for an image object with callouts
- `important` - An admonition set off from the text
- `index` - An index
- `indexdiv` - A division in an index
- `indexentry` - An entry in an index
- `indexinfo` - Meta-information for an Index
- `indexterm` - A wrapper for terms to be indexed
- `informalequation` - A displayed mathematical equation without a title
- `informalexample` - A displayed example without a title
- `informalfigure` - A untitled figure
- `informaltable` - A table without a title
- `initializer` - The initializer for a FieldSynopsis
- `inlineequation` - A mathematical equation or expression occurring inline
- `inlinegraphic` - An object containing or pointing to graphical data that will be rendered inline

- `inlinemediaobject` - An inline media object (video, audio, image, and so on)
- `interface` - An element of a GUI
- `interfacename` - The name of an interface
- `invpartnumber` - An inventory part number
- `isbn` - The International Standard Book Number of a document
- `issn` - The International Standard Serial Number of a periodical
- `issuenum` - The number of an issue of a journal
- `itemizedlist` - A list in which each entry is marked with a bullet or other dingbat
- `itermset` - A set of index terms in the meta-information of a document
- `jobtitle` - The title of an individual in an organization
- `keycap` - The text printed on a key on a keyboard
- `keycode` - The internal, frequently numeric, identifier for a key on a keyboard
- `keycombo` - A combination of input actions
- `keysym` - The symbolic name of a key on a keyboard
- `keyword` - One of a set of keywords describing the content of a document
- `keywordset` - A set of keywords describing the content of a document
- `label` - A label on a Question or Answer
- `legalnotice` - A statement of legal obligations or requirements
- `lhs` (EBNF) - The left-hand side of an EBNF production
- `lineage` - The portion of a person's name indicating a relationship to ancestors
- `lineannotation` - A comment on a line in a verbatim listing
- `link` - A hypertext link
- `listitem` - A wrapper for the elements of a list item
- `literal` - Inline text that is some literal value
- `literallayout` - A block of text in which line breaks and white space are to be reproduced faithfully
- `lot` - A list of the titles of formal objects (as tables or figures) in a document
- `lotentry` - An entry in a list of titles
- `manvolnum` - A reference volume number
- `markup` - A string of formatting markup in text that is to be represented literally
- `mml:math` (MathML) - A MathML equation
- `medialabel` - A name that identifies the physical medium on which some information resides

Appendix C. Overview of all DocBook elements

- `mediaobject` - A displayed media object (video, audio, image, etc.)
- `mediaobjectco` - A media object that contains callouts
- `member` - An element of a simple list
- `menuchoice` - A selection or series of selections from a menu
- `methodname` - The name of a method
- `methodparam` - Parameters to a method
- `methodsynopsis` - A syntax summary for a method
- `modespec` - Application-specific information necessary for the completion of an OLink
- `modifier` - Modifiers in a synopsis
- `mousebutton` - The conventional name of a mouse button
- `msg` - A message in a message set
- `msgaud` - The audience to which a message in a message set is relevant
- `msgentry` - A wrapper for an entry in a message set
- `msgexplan` - Explanatory material relating to a message in a message set
- `msginfo` - Information about a message in a message set
- `msglevel` - The level of importance or severity of a message in a message set
- `msgmain` - The primary component of a message in a message set
- `msgorig` - The origin of a message in a message set
- `msgrel` - A related component of a message in a message set
- `msgset` - A detailed set of messages, usually error messages
- `msgsub` - A subcomponent of a message in a message set
- `msgtext` - The actual text of a message component in a message set
- `nonterminal` (EBNF) - A non-terminal in an EBNF production
- `note` - A message set off from the text
- `objectinfo` - Meta-information for an object
- `olink` - A link that addresses its target indirectly, through an entity
- `ooclass` - A class in an object-oriented programming language
- `ooexception` - An exception in an object-oriented programming language
- `oointerface` - An interface in an object-oriented programming language
- `option` - An option for a software command
- `optional` - Optional information

- `orderedlist` - A list in which each entry is marked with a sequentially incremented label
- `orgdiv` - A division of an organization
- `orgname` - The name of an organization other than a corporation
- `otheraddr` - Uncategorized information in address
- `othercredit` - A person or entity, other than an author or editor, credited in a document
- `othername` - A component of a persons name that is not a first name, surname, or lineage
- `pagenums` - The numbers of the pages in a book, for use in a bibliographic entry
- `para` - A paragraph
- `paramdef` - Information about a function parameter in a programming language
- `parameter` - A value or a symbolic reference to a value
- `part` - A division in a book
- `partinfo` - Meta-information for a Part
- `partintro` - An introduction to the contents of a part
- `phone` - A telephone number
- `phrase` - A span of text
- `pob` - A post office box in an address
- `postcode` - A postal code in an address
- `preface` - Introductory matter preceding the first chapter of a book
- `prefaceinfo` - Meta-information for a Preface
- `primary` - The primary word or phrase under which an index term should be sorted
- `primaryie` - A primary term in an index entry, not in the text
- `printhistory` - The printing history of a document
- `procedure` - A list of operations to be performed in a well-defined sequence
- `production (EBNF)` - A production in a set of EBNF productions
- `productionrecap (EBNF)` - A cross-reference to an EBNF production
- `productionset (EBNF)` - A set of EBNF productions
- `productname` - The formal name of a product
- `productnumber` - A number assigned to a product
- `programlisting` - A literal listing of all or part of a program
- `programlistingco` - A program listing with associated areas used in callouts
- `prompt` - A character or string indicating the start of an input field in a computer display

Appendix C. Overview of all DocBook elements

- **property** - A unit of data associated with some part of a computer system
- **pubdate** - The date of publication of a document
- **publisher** - The publisher of a document
- **publishername** - The name of the publisher of a document
- **pubsnumber** - A number assigned to a publication other than an ISBN or ISSN or inventory part number
- **qandadiv** - A titled division in a QandASet
- **qandaentry** - A question/answer set within a QandASet
- **qandaset** - A question-and-answer set
- **question** - A question in a QandASet
- **quote** - An inline quotation
- **refclass** - The scope or other indication of applicability of a reference entry
- **refdescriptor** - A description of the topic of a reference page
- **refentry** - A reference page (originally a UNIX man-style reference page)
- **refentryinfo** - Meta-information for a Refentry
- **refentrytitle** - The title of a reference page
- **reference** - A collection of reference entries
- **referenceinfo** - Meta-information for a Reference
- **refmeta** - Meta-information for a reference entry
- **refmiscinfo** - Meta-information for a reference entry other than the title and volume number
- **refname** - The name of (one of) the subject(s) of a reference page
- **refnamediv** - The name, purpose, and classification of a reference page
- **refpurpose** - A short (one sentence) synopsis of the topic of a reference page
- **refsect1** - A major subsection of a reference entry
- **refsect1info** - Meta-information for a RefSect1
- **refsect2** - A subsection of a RefSect1
- **refsect2info** - Meta-information for a RefSect2
- **refsect3** - A subsection of a RefSect2
- **refsect3info** - Meta-information for a RefSect3
- **refsynopsisdiv** - A syntactic synopsis of the subject of the reference page
- **refsynopsisdivinfo** - Meta-information for a RefSynopsisDiv
- **releaseinfo** - Information about a particular release of a document

- **remark** - A remark (or comment) intended for presentation in a draft manuscript
- **replaceable** - Content that may or must be replaced by the user
- **returnvalue** - The value returned by a function
- **revdescription** - A extended description of a revision to a document
- **revhistory** - A history of the revisions to a document
- **revision** - An entry describing a single revision in the history of the revisions to a document
- **revnumber** - A document revision number
- **revremark** - A description of a revision to a document
- **rhs (EBNF)** - The right-hand side of an EBNF production
- **row** - A row in a table
- **sbr** - An explicit line break in a command synopsis
- **screen** - Text that a user sees or might see on a computer screen
- **screenco** - A screen with associated areas used in callouts
- **screeninfo** - Information about how a screen shot was produced
- **screenshot** - A representation of what the user sees or might see on a computer screen
- **secondary** - A secondary word or phrase in an index term
- **secondaryie** - A secondary term in an index entry, rather than in the text
- **sect1** - A top-level section of document
- **sect1info** - Meta-information for a Sect1
- **sect2** - A subsection within a Sect1
- **sect2info** - Meta-information for a Sect2
- **sect3** - A subsection within a Sect2
- **sect3info** - Meta-information for a Sect3
- **sect4** - A subsection within a Sect3
- **sect4info** - Meta-information for a Sect4
- **sect5** - A subsection within a Sect4
- **sect5info** - Meta-information for a Sect5
- **section** - A recursive section
- **sectioninfo** - Meta-information for a recursive section
- **see** - Part of an index term directing the reader instead to another entry in the index
- **seealso** - Part of an index term directing the reader also to another entry in the index

Appendix C. Overview of all DocBook elements

- `seealsoie` - A See also entry in an index, rather than in the text
- `seeie` - A See entry in an index, rather than in the text
- `seg` - An element of a list item in a segmented list
- `seglistitem` - A list item in a segmented list
- `segmentedlist` - A segmented list, a list of sets of elements
- `segtitle` - The title of an element of a list item in a segmented list
- `seriesvolnums` - Numbers of the volumes in a series of books
- `set` - A collection of books
- `setindex` - An index to a set of books
- `setindexinfo` - Meta-information for a SetIndex
- `setinfo` - Meta-information for a Set
- `sgmltag` - A component of SGML markup
- `shortaffil` - A brief description of an affiliation
- `shortcut` - A key combination for an action that is also accessible through a menu
- `sidebar` - A portion of a document that is isolated from the main narrative flow
- `sidebarinfo` - Meta-information for a Sidebar
- `simpara` - A paragraph that contains only text and inline markup, no block elements
- `simplelist` - An undecorated list of single words or short phrases
- `simplemsgentry` - A wrapper for a simpler entry in a message set
- `simplesect` - A section of a document with no subdivisions
- `spanspec` - Formatting information for a spanned column in a table
- `state` - A state or province in an address
- `step` - A unit of action in a procedure
- `street` - A street address in an address
- `structfield` - A field in a structure (in the programming language sense)
- `structname` - The name of a structure (in the programming language sense)
- `subject` - One of a group of terms describing the subject matter of a document
- `subjectset` - A set of terms describing the subject matter of a document
- `subjectterm` - A term in a group of terms describing the subject matter of a document
- `subscript` - A subscript (as in H₂O, the molecular formula for water)
- `substeps` - A wrapper for steps that occur within steps in a procedure

- `subtitle` - The subtitle of a document
- `superscript` - A superscript (as in x^2 , the mathematical notation for x multiplied by itself)
- `surname` - A family name; in western cultures the last name
- `svg:svg` (SVG) - An SVG graphic
- `symbol` - A name that is replaced by a value before processing
- `synopfragment` - A portion of a `CmdSynopsis` broken out from the main body of the synopsis
- `synopfragmentref` - A reference to a fragment of a command synopsis
- `synopsis` - A general-purpose element for representing the syntax of commands or functions
- `systemitem` - A system-related item or term
- `table` - A formal table in a document
- `tbody` - A wrapper for the rows of a table or informal table
- `term` - The word or phrase being defined or described in a variable list
- `tertiary` - A tertiary word or phrase in an index term
- `tertiaryie` - A tertiary term in an index entry, rather than in the text
- `textobject` - A wrapper for a text description of an object and its associated meta-information
- `tfoot` - A table footer consisting of one or more rows
- `tgroup` - A wrapper for the main content of a table, or part of a table
- `thead` - A table header consisting of one or more rows
- `tip` - A suggestion to the user, set off from the text
- `title` - The text of the title of a section of a document or of a formal block-level element
- `titleabbrev` - The abbreviation of a Title
- `toc` - A table of contents
- `tocback` - An entry in a table of contents for a back matter component
- `tocchap` - An entry in a table of contents for a component in the body of a document
- `tocentry` - A component title in a table of contents
- `tocfront` - An entry in a table of contents for a front matter component
- `toclevel1` - A top-level entry within a table of contents entry for a chapter-like component
- `toclevel2` - A second-level entry within a table of contents entry for a chapter-like component
- `toclevel3` - A third-level entry within a table of contents entry for a chapter-like component
- `toclevel4` - A fourth-level entry within a table of contents entry for a chapter-like component
- `toclevel5` - A fifth-level entry within a table of contents entry for a chapter-like component

Appendix C. Overview of all DocBook elements

- `tocpart` - An entry in a table of contents for a part of a book
- `token` - A unit of information
- `trademark` - A trademark
- `type` - The classification of a value
- `ulink` - A link that addresses its target by means of a URL (Uniform Resource Locator)
- `userinput` - Data entered by the user
- `varargs` - An empty element in a function synopsis indicating a variable number of arguments
- `variablelist` - A list in which each entry is composed of a set of one or more terms and an associated description
- `varlistentry` - A wrapper for a set of terms and the associated description in a variable list
- `varname` - The name of a variable
- `videodata` - Pointer to external video data
- `videoobject` - A wrapper for video data and its associated meta-information
- `void` - An empty element in a function synopsis indicating that the function in question takes no arguments
- `volumenum` - The volume number of a document in a set (as of books in a set or articles in a journal)
- `warning` - An admonition set off from the text
- `wordasword` - A word meant specifically as a word and not representing anything else
- `xref` - A cross reference to another part of the document
- `year` - The year of publication of a document

Index

A

- acroread 8
- addresses, Internet syntax 64
- angle brackets
 - SGML tags 63
 - XML tags 63
- appearance
 - cooked data 61
 - raw data 63
 - stylesheets 63
- Attributes
 - align 27
 - cols 23
 - endterm 29
 - format 27
 - id 29
 - scale 28
- attributes 61

C

- callouts 61
- chunk
 - docbook2html 5, 59
- cooked data 61
 - raw data
- customization (DocBook DTD)
 - parameter entities, using 62

D

- data entities 61
- declarations
 - document type declaration 61
- DocBook DTD
 - customizing
 - parameter entities 62
- docbook2html 5, 42
 - chunk 5, 59
- docbook2pdf 7, 42, 58
- document type declaration 61
 - external subset 61
 - internal subset 62
- documents
 - external, references to 61
 - meta-information 62
- DTDs
 - public identifiers 63

E

- Elements
 - abbrev 39, 65
 - abstract 15, 39, 65
 - accel 65
 - accel - keycap 32
 - ackno 65

acronym 65
 action 32 , 65
 address 39 , 65
 affiliation 39 , 65
 alt 65
 anchor 29 , 65
 answer 17 , 65
 appendix 65
 appendixinfo 65
 application 45 , 65
 area 65
 areaset 65
 areaspec 65
 arg 33 , 65
 article 65
 articleinfo 15 , 65
 artpagenums 39 , 65
 attribution 65
 audiodata 65
 audioobject 65
 author 15 , 39 , 65
 authorblurb 39 , 65
 authorgroup 15 , 39 , 65
 authorinitials 15 , 39 , 65
 beginpage 65
 bibliodiv 39 , 65
 biblioentry 39 , 66
 bibliography 39 , 66
 bibliographyinfo 39 , 66
 bibliomisc 39 , 66
 bibliomix 39
 bibliomixed 39 , 66
 bibliomset 39 , 66
 biblioset 39 , 66
 blockquote 66
 book 66
 bookinfo 15 , 66
 bridgehead 66
 callout 66
 calloutlist 66
 caption 66
 caution 45 , 66
 chapter 66
 chapterinfo 66
 citation 39 , 66
 citerefentry 39 , 66
 citetitle 39 , 66
 city 39 , 66
 classname 36 , 66
 classsynopsis 36 , 66
 classsynopsisinfo 36 , 66
 cmdsynopsis 34 , 66
 co 66
 collab 39 , 66
 collabname 39 , 66
 colophon 66
 colspec 66

command 34 , 67
computeroutput 34 , 67
confdates 67
confgroup 39 , 67
confnum 67
confsponsor 67
conftitle 67
constant 67
constraint 67
constraintdef 67
constructorsynopsis 67
contractnum 39 , 67
contractsponsor 39 , 67
contrib 39 , 67
copyright 39 , 67
corpauthor 40 , 67
corpname 40 , 67
country 40 , 67
database 67
date 15 , 40 , 67
dedication 67
destructorsynopsis 67
edition 40 , 67
editor 40 , 67
email 29 , 67
Emphasis 67
emphasis 45
entry 23 24, 67 24
entrytbl 23 24, 67 24
envar 34 , 67
epigraph 67
equation 67
errorcode 68
errorname 68
errortype 68
example 31 , 68
exceptionname 36 , 68
fax 68
fieldsynopsis 36 , 68
figure 68
filename 34 , 68
firstname 15 , 40 , 68
firstterm 68
footnote 68
footnoteref 68
foreignphrase 68
formalpara 68
funcdef 34 , 68
funcparams 68
funcprototype 34 , 68
funcsynopsis 34 , 68
funcsynopsisinfo 34 , 68
function 34 , 68
glossary 68
glossaryinfo 68
glossdef 68
glossdiv 68

glossentry 68
 glosslist 68
 glossee 68
 glosseesalso 68
 glossterm 68
 graphic 68
 graphicco 68
 group 69
 guibutton 32 , 69
 guiicon 32 , 69
 guilabel 32 , 69
 guimenu 32 , 69
 guimenuitem 32 , 69
 guisubmenu 32 , 69
 hardware 69
 highlights 69
 holder 69
 honorific 40 , 69
 imagedata 27 , 69
 imageobject 28 , 69
 imageobjectco 69
 important 45 , 69
 index 69
 indexdiv 69
 indexentry 69
 indexinfo 69
 indexterm 69
 informalequation 69
 informalexample 31 , 69
 informalfigure 69
 informatable 23 , 69
 initializer 69
 inlineequation 69
 inlinegraphic 69
 inlinemediaobject 27 , 69
 interface 32 , 69
 interfacedefinition 32
 interfacename 69
 invpartnumber 69
 isbn 40 , 70
 issn 40 , 70
 issuenum 40 , 70
 itemizedlist 17 , 70
 itemset 70
 jobtitle 70
 keycap 32 , 70
 keycode 32 , 70
 keycombo 32 , 70
 keysym 32 , 70
 keyword 15 , 70
 keywordset 15 , 70
 label 70
 legalnotice 70
 lhs 70
 lineage 70
 lineannotation 70
 link 29 , 70

- link and xref 29
- listitem 17 , 70
- literal 34 , 70
- literallayout 31 , 70
- lot 70
- lotentry 70
- manvolnum 70
- markup 45 , 70
- medialabel 70
- mediaobject 27 , 70
- mediaobjectco 70
- member 17 , 70
- menuchoice 32 , 71
- methodname 36 , 71
- methodparam 36 , 71
- methodsynopsis 36 , 71
- mml:math 70
- modespec 71
- modifier 36 , 71
- mousebutton 32 , 71
- msg 71
- msgaud 71
- msgentry 71
- msgexplan 71
- msginfo 71
- msglevel 71
- msgmain 71
- msgorig 71
- msgrel 71
- msgset 71
- msgsub 71
- msgtext 71
- nonterminal 71
- note 45 , 71
- objectinfo 71
- olink 29 , 71
- ooclass 36 , 71
- ooexception 36 , 71
- oointerface 36 , 71
- option 34 , 71
- optional 71
- orderedlist 17 , 71
- orgdiv 71
- orgname 40 , 71
- otheraddr 71
- othercredit 40 , 72
- othername 15 , 40 , 72
- pagenums 40 , 72
- para 72
- paramdef 34 , 72
- parameter 34 , 72
- part 72
- partinfo 72
- partintro 72
- phone 72
- phrase 72
- pob 72

postcode 72
 preface 72
 prefaceinfo 72
 primary 72
 primaryie 72
 printhistory 40 , 72
 procedure 17 , 72
 production 72
 productionrecap 72
 productionsset 72
 productname 72
 productnumber 72
 programlisting 31 , 72
 programlistingco 72
 prompt 34 , 72
 property 72
 pubdate 40 , 72
 publisher 40 , 72
 publishername 40 , 72
 pubsnumber 40 , 73
 qandadiv 73
 qandaentry 17 , 73
 qandaset 17 , 73
 question 17 , 73
 quote 73
 refclass 73
 refdescriptor 73
 refentry 73
 refentryinfo 73
 refentrytitle 73
 reference 73
 referenceinfo 73
 refmeta 73
 refmiscinfo 73
 refname 73
 refnamediv 73
 refpurpose 73
 refsect1 73
 refsect1info 73
 refsect2 73
 refsect2info 73
 refsect3 73
 refsect3info 73
 refsynopsisdiv 73
 refsynopsisdivinfo 73
 releaseInfo 15
 releaseinfo 40 , 73
 remark 73
 replaceable 34 , 73
 returnvalue 73
 revdescription 74
 revhistory 15 , 40 , 74
 revision 15 , 74
 revnumber 15 , 74
 revremark 15 , 74
 rhs 74
 row 23 24, 74 24

sbr 74
screen 31 , 74
screenco 74
screeninfo 27 , 74
screenshot 27 , 74
secondary 74
secondaryie 74
sect1 74
sect1info 74
sect2 74
sect2info 74
sect3 74
sect3info 74
sect4 74
sect4info 74
sect5 74
sect5info 74
section 74
sectioninfo 74
see 74
seealso 74
seealsoie 74
seeie 74
seg 17 , 74
seglistitem 17 , 74
segmentedlist 17 , 75
segtitle 17 , 75
seriesvolnums 40 , 75
set 75
setindex 75
setindexinfo 75
setinfo 75
sgmltag 75
shortaffil 75
shortcut 75
sidebar 75
sidebarinfo 75
simpara 75
simplelist 17 , 75
simplemsgentry 75
simplesect 75
spanspec 75
state 75
step 17 , 75
street 75
structfield 75
structname 75
subject 75
subjectset 75
subjectterm 75
subscript 45 , 75
substeps 17 , 75
subtitle 40 , 75
superscript 45 , 75
surname 15 , 40 , 75
svg:svg 75
symbol 34 , 76

- synopfragment 76
- synopfragmentref 76
- synopsis 76
- systemitem 76
- table 23, 76
- tbody 23, 76
- term 17, 76
- tertiary 76
- tertiaryie 76
- textobject 76
- tfoot 23, 76
- tgroup 23, 76
- thead 23, 76
- tip 45, 76
- title 15, 23, 29, 40, 76
- titleabbrev 40, 76
- toc 76
- tocback 76
- tocchap 76
- tocentry 76
- tocfront 76
- toclevel1 76
- toclevel2 76
- toclevel3 76
- toclevel4 76
- toclevel5 76
- tocpart 76
- token 76
- trademark 76
- type 34, 76
- ulink 29, 77
- userinput 34, 77
- varargs 77
- variablelist 17, 77
- varlistentry 17, 77
- varname 77
- videodata 77
- videoobject 77
- void 77
- volumenum 40, 77
- warning 45, 77
- wordasword 77
- xref 29, 77
- year 77
- elements 61
 - attributes 61
 - tags 63
 - wrappers 64
- empty elements 61
- entities 61
 - general 62
- Extensible Markup Language
 - XML
- external general entities 61
 - public identifiers 63
- external subset 61

F

float 61

G

general entities 62

- external 61

- internal 62

ghostview 8

H

hierarchical structure

- elements, defining 61

HTML

- HyperText Markup Language 62

HyperText Markup Language (HTML) 62

I

internal general entities 62

internal subset 62

Internet names and addresses 64

ISO standards

- SGML 63

M

marked sections

- parameter entities, controlling 62

meta-information 62

- wrappers 64

N

names

- Internet, syntax 64

P

parameter entities 62

PI

- processing instructions

processing instructions 62

public identifiers 63

- external subset 61

R

raw data 63

- cooked data

references

- external documents 61

S

SGML 63

- public identifiers 63

- system identifiers 63

- tags 63

- XML and 64

special characters, encoding as entities 62

- start tags
 - attribute ID, containing 61
 - empty element 61
- stylesheets 63
- system identifiers
 - external subset 61
 - SGML 63
 - XML 63

T

- tags 63
- text
 - float 61

U

- Unicode 63
 - UTF16 63
 - UTF8 63
- URI 64
 - XML system identifiers 63
- URL 64
- URN 64
- UTF16
 - Unicode 63
- UTF8
 - Unicode 63

W

- W3C (World Wide Web Consortium) 64
- Web 64
- World Wide Web (WWW) 64
- World Wide Web Consortium (W3C) 64
- wrappers 64

X

- XML 64
 - processing instructions 63
 - public identifiers 63
 - system identifiers 63
 - tags 63
 - XSL stylesheets 64
- XSL 64