

Docbook开发手记

版本：v2.4

Crifan Li

摘要

本文先简介了什么是Docbook,然后详细介绍了如何搭建中文的Docbook开发环境,再详细记录了Docbook开发过程中所遇到的各种注意事项和遇到的问题及解决办法,同时给出很多常见的Docbook中各种元素的示例代码,另外也记录了一些值得参考的Docbook相关的资料。



本文提供多种格式供：

在线阅读	HTML¹	HTMLS²	PDF³	CHM⁴	TXT⁵	RTF⁶	WEBHELP⁷
下载（7zip压缩包）	HTML⁸	HTMLS⁹	PDF¹⁰	CHM¹¹	TXT¹²	RTF¹³	WEBHELP¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/docbook_dev_note/

修订历史

修订 2.4	2013-10-09	crl
<ol style="list-style-type: none">1. 完善了代码示例部分2. 更新整理了appendix附录部分，把最新的所有的book都整理进来了3. 添加了filename，screenco等示例代码4. 更新了所有的xml:id5. 添加示例代码：figure内包含2张mediaobject图片6. 添加示例代码：内嵌元素中的keycap7. 修正了个别的笔误，修改了部分细节内容，添加了引用reference8. 添加整理了：Docbook相关的资源和工具下载		

¹ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html

² http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/pdf/docbook_dev_note.pdf

⁴ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/chm/docbook_dev_note.chm

⁵ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/txt/docbook_dev_note.txt

⁶ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/rtf/docbook_dev_note.rtf

⁷ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html.7z

⁹ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/pdf/docbook_dev_note.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/chm/docbook_dev_note.chm.7z

¹² http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/txt/docbook_dev_note.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/rtf/docbook_dev_note.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/webhelp/docbook_dev_note.webhelp.7z

9. 添加了：Docbook相关的知识和概念		
10. 添加示例：programlisting和screen		
11. 添加示例：filename是后缀名		
修订 1.1	2012-06-14	crl
1. 完成此文的框架		
2. 各个章节都已添加了一些基本的内容，关于参考资料的部分，已添加大部分内容		
3. 收集了一些常见问题及解决办法		
4. 分割成多个文件		
5. 添加了关于实体定义部分的介绍		
6. 添加了调试fo部分的内容		
7. 添加了如何搭建docbook环境		
8. 添加了什么是Docbook		
9. 添加了代码示例部分		

Docbook开发手记:

Crifan Li

版本 : v2.4

出版日期 2013-10-09

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

Docbook介绍	x
1. 什么是Docbook	x
1.1. Docbook出现的背景	x
1.2. Docbook是什么	x
1.2.1. 截图说明Docbook的：一种（xml）格式输入，多种 （html，htmls，chm，pdf，txt，rtf等等）格式输出	xii
1.2.1.1. Docbook的一种输入：xml源码（文件）	xii
1.2.1.2. Docbook的多种输出： HTML，HTMLs，PDF，CHM，RTF，TXT，WEBHELP等格式	xiii
2. 我为何选择Docbook	xviii
1. Windows下的Docbook的环境搭建	1
1.1. 搭建Docbook之前需要知道的最基本的事情	1
1.1.1. 我们的目标	1
1.1.2. 关于格式转换方面的知识	1
1.2. 纯Windows环境下的Docbook开发环境的搭建	1
1.2.1. 建立好文件夹	2
1.2.2. 下载windos版本的xsltproc	5
1.2.3. 下载docbook-xsl-ns-1.77.1	6
1.2.4. 下载fop-1.0并添加fop配置文件	6
1.2.4.1. 下载fop-1.0	6
1.2.4.2. 配置FOP的环境变量FOP_HOME	6
1.2.4.3. 添加fop配置文件fop.xconf	7
1.2.5. 准备docbook的源码：xml文件	7
1.2.6. 从xml生成html	8
1.2.6.1. 为生成html准备配置文件	8
1.2.6.2. 将xml转换为html	9
1.2.7. 从xml生成pdf	10
1.2.7.1. 为生成pdf准备配置文件	10
1.2.7.2. 将xml转换为fo	11
1.2.7.3. 将fo转换为pdf	11
1.3. Windows环境下的，基于Cygwin的Docbook开发环境的搭建	13
1.3.1. 安装Cygwin	13
1.3.2. 下载cygwin下使用fop-1.0	14
2. Docbook开发过程中的注意事项和心得	16
2.1. Docbook 4和Docbook 5的区别	16
2.2. 关于实体定义	16
2.3. qanda系列的用法	18
2.4. FOP相关注意事项	18
2.4.1. FOP的字体设置	18
2.5. Docbook开发过程中的一些有用的提示	19
2.5.1. Docbook最佳实践	19
2.5.1.1. Docbook中id的命名规则	20
2.6. Docbook开发过程中如何调试错误	21
2.6.1. 举例说明如何实现方便地调试Docbook生成pdf过程中所出现的错误	21
2.7. Docbook开发过程中的一些感悟	28
2.7.1. Docbook更关注内容	28
2.7.2. Docbook与盖楼	28
2.8. Docbook环境进化过程简单记录	28
3. Docbook开发过程中的常见问题及解答	30
常见问题	30
3.1. ValidationException: Property ID xxx previously used	30
3.2. ValidationException: xxx is not a valid child of xxx	30
3.3. ValidationException: xxx is missing child elements	30
3.4. docbook5中指定表格table中列column的宽度	31

3.5. 除了章 (Chapter) 有编号之外, 其他不同层级的小节 (section) 都是没有编号的, 想要给各小节添加索引编号	32
3.6. 已经include了xsl配置文件了, 但是自己添加的参数配置, 不起效果	32
3.7. xsltproc出错: Element xxx in namespace xxx encountered in book, but no template matches	32
3.8. pdf中没有出现/显示revhistory所对应的历史版本	34
3.9. 生成pdf的时候出错: 严重: Image not found. URI: xxx.png. (See position xxx)	36
3.10. 想要给book中的标题下面添加一个版本号	36
3.11. 输出的pdf中的revhistory历史版本的内容中的表格无边框, 想要添加边框	36
3.12. 如何去掉docbook中输出的pdf中, 除了带链接的文字之外的那个链接地址	37
3.13. 给docbook生成的pdf中的带链接的文字, 添加下划线和设置字体颜色为蓝色	37
3.14. 如何给docbook的pdf添加bookmark书签的功能	37
3.15. Docbook的pdf, 去除正文对于标题的缩进indent	38
3.16. 生成的pdf中emphasis无效果, 想要实现给pdf中emphasis的文字加粗或斜体	38
3.17. 如何给 (docbook中的) xsl中定义实体Entity	39
3.18. 给docbook的pdf中的等式(equation)设置背景色	40
3.19. 如何将默认的输出单个的HTML页面, 改为输入多个HTML页面?	40
3.20. 如何给docbook中的表格 (table) 的第一行 (firstrow或thead) 添加背景色?	40
3.21. 如何用docbook生成chm (微软帮助文件) ?	43
3.22. docbook生成的chm的左边目录等内容都是乱码	43
3.23. 如何将qanda的标题添加到TOC目录中去?	44
3.24. 如何将docbook转换为 (和word兼容的) RTF文档?	44
3.25. 如何用Docbook生成Webhelp?	45
3.26. 用fop生成pdf过程中出现警告: warning: Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400"	45
3.27. Docbook中使用fop过程中出错: 严重: Couldn' t find hyphenation pattern zh_cn	45
3.28. cygwin/linux下的fop运行出错: Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/xmlgraphics/image/loader/ImageContext	46
3.29. Docbook中用xsltproc生成HTML过程中出错: runtime error xxx/chunker.xsl line 226 element document, invalid value for indent,xsltRunStylesheet: run failed	46
3.30. docbook的xsltproc生成的HTML中Charset的值是空的, 不是所设置的UTF-8	46
3.31. 如何给Docbook中添加abbrev缩略词一项, 且放置在目录之后, 正文之前?	47
3.32. 给Docbook的HTML的等式 (equation), 程序源码 (programlisting) 等添加背景色	48
3.33. PDF中和HTML中的图片显示, 始终不能统一, 要么HTML中是放大的, 要么PDF中不能完全显示, 所以希望PDF和HTML中的图片显示可以用同一套代码来统一	49
3.34. 想要设置HTML和PDF中的图片的标题都是居中对齐	50
3.35. 生成PDF中, 程序代码programlisting中单行代码太长, 超出页面宽度, 无法显示	51
3.36. Docbook生成的pdf中, 当表格在quote中时, 表格头部却在底端显示, 而且如果内容是多行的, 还会显示出多行	51
3.37. Docbook中的callout图片在programlisting中不显示	51
3.38. An fo:block is wider than the available room in inline-progression-dimension	52
3.39. TransformerException: The column-number or number of cells in the row overflows the number of fo:table-columns specified for the table	52
3.40. 如何通过xref (在别处) 引用某个段落 (para) ?	53
3.41. WARNING: Content overflows the viewport of the fo:region-body on page 9 in block-progression direction by 99828 millipoints. (See position 2:5700)	53

3.42. PDF中callout不能点击跳转（而HTML中却可以）	53
3.43. callout图片在programlisting的源码中不显示	56
3.44. 如何在（只使用co而没有使用areaspec的）源码中多个位置，指向同一个co（callout）？	57
3.45. 新的docbook-xsl-ns-1.77.0生成的html中的revhistory中单元格无边框	58
3.46. WARNING: Glyph "?" (0x21b5, carriagereturn) not available in font "MicrosoftYaHei".	60
3.47. 如何实现给table中（多个）特定位置添加相应的注释？	61
4. Docbook参考资料	63
4.1. Docbook相关的资源和工具下载	63
4.2. 关于对于Docbook的介绍和解释方面的资料	63
4.3. 关于对于Docbook的开发过程中可以参考和查阅的资料	64
5. 未解决的问题	70
5.1. 支持各种语言的代码的语法高亮	70
5.2. 中文hypertext断字	70
5.3. 给callout的co重新编号	70
6. Docbook各种元素用法举例	71
6.1. chapter, section, sect1-sect5	71
6.2. 段落	72
6.3. 图片figure	73
6.4. 表格，表格嵌套，表格单元格合并	74
6.5. 注释callout	76
6.6. 链接，引用	79
6.7. 等式	79
6.8. list列表系列	80
6.9. 简单列表simplelist，literallayout	81
6.10. xinclude文件，xinclude文件中的其中一部分的内容	82
6.11. filename：文件，文件夹，后缀名，	84
6.12. admonitions：note，caution，warning，tip和important	84
6.12.1. note	84
6.12.2. caution	85
6.12.3. warning	85
6.12.4. tip	85
6.12.5. important	86
6.13. 一些inline元素：keycap，guibutton，keycombo	86
6.13.1. keycap	86
6.14. programlisting代码和screen屏幕输出	86
6.14.1. programlisting代码	86
6.14.1.1. 带CDATA的programinglist示例	86
6.14.1.2. 不带CDATA的programinglist示例	87
6.14.2. screen屏幕输出	87
6.14.2.1. 带CDATA的屏幕输出screen	88
6.14.2.2. 不带CDATA的屏幕输出screen	88
7. Docbook相关的知识和概念	89
7.1. 输出类型：html和打印输出print out	89
7.2. docbook中的条件编译：profiling	89
7.3. 关于docbook中的图片方面的内容	89
7.4. docbook的一些变体	90
7.4.1. 简化版的docbook	90
7.4.2. docbook(制作)幻灯片	90
7.4.3. docbook(制作)网站	91
A. 已通过Docbook发布的资料	92
A.1. 基础知识相关	92
A.2. 嵌入式相关	92
A.3. 上层软件相关	93
A.4. 其他	94
参考书目	95

插图清单

1. 截图举例：Docbook的XML源码	xii
2. 截图举例：Docbook的可以生成多种格式	xiii
3. 截图举例：Docbook的XML源码生成的HTML	xiv
4. 截图举例：Docbook的XML源码生成的HTMLs	xiv
5. 截图举例：Docbook的XML源码生成的PDF	xv
6. 截图举例：Docbook的XML源码生成的CHM	xv
7. 截图举例：Docbook的XML源码生成的TXT	xvi
8. 截图举例：Docbook的XML源码生成的RTF	xvii
9. 截图举例：Docbook的XML源码生成的WEBHELP	xvii
3.1. coref在HTML中的效果	58

表格清单

2.1. Docbook中id命名规则	20
---------------------------	----

范例清单

3.1. co用法示例	56
3.2. coref用法示例	57
6.1. chapter,section,sect1-sect5	71
6.2. 普通的段落:para	72
6.3. 带标题的正式的段落:formalpara	72
6.4. 带标题的图片:figure	73
6.5. figure内包含2张mediaobject图片	73
6.6. 不带标题的非正式的图片:informalfigure	73
6.7. 行内图片:inlinemediaobject	74
6.8. 表格:table	74
6.9. 带嵌套 (embedded) 表格:informaltable	75
6.10. programlisting的callout:programlistingco	76
6.11. screen的callout:screenco	77
6.12. 没有callout的co实现在任意位置使用co然后在别处使用coref去引用	78
6.13. 带xlink域名的link	79
6.14. xref引用	79
6.15. 等式equation	79
6.16. itemizedlist	80
6.17. 带emphasis的itemizedlist	80
6.18. 带指定编号类型的orderedlist	81
6.19. 简单列表simplelist	81
6.20. 简单列举literallayout	81
6.21. xinclude主文件中使用xinclude	82
6.22. 被xinclude包含的子文件: preface.xml	83
6.23. 被xinclude包含的子文件: ch01_build_env.xml	83
6.24. filename是文件, 文件夹的例子	84
6.25. 示例: filename是后缀名	84
6.26. note举例	84
6.27. caution举例	85
6.28. tip举例	85
6.29. keycap举例	86
6.30. 举例: 带CDATA的programinglist	86
6.31. 举例: 不带CDATA的programinglist	87
6.32. 举例: 带CDATA的screen	88
6.33. 举例: 不带CDATA的screen	88
7.1. 添加role参数实现条件编译	89

Docbook介绍

1. 什么是Docbook

1.1. Docbook出现的背景

想要了解Docbook是啥之前，需要知道一下其出现的大概背景，这样才能更好的理解为何会有docbook，以及其到底是啥。

随着计算机发展，有很多专业的书籍，技术文章等，需要发布。

这句话需要解释一下：

- 专业的书籍，技术文章

此处所说的书籍book和文章article，主要指的是和计算机相关的

当然后面也会提到，docbook的应用，不仅仅局限于发布计算机相关的书籍和文章。

即，其他专业，方向的书籍和文章等，也都可以通过docbook发布。

- 发布

此处的发布，主要包括两种，一种是在线Online的，一种是离线的Offline的。

在线的，多指的是文章通过HTML等网页形式，可以在线浏览。

离线的，也可以是将HTML网页下载下来留作本地阅读，更多的时候指的是，发布为PDF，CHM等格式，方便本地打开对应的文件，阅读该书籍或文章。

而对于各种书籍和文章的发布，由于没有统一的系统支持，导致想要发布多种格式，多人同时编辑同一份文档等情况的时候，之前的直接编辑生成相应文档的方法，就显得力不从心了。

所以才会有Docbook这种东西出现。其统一规划了（尤其是计算机类的）书籍所需要的“单词”，以及规定了“单词”之间的关系，

作为文档发布者，按照其规定，写好相应的**源码**

之后，有其他第三方（多数是开源的，免费的）工具，实现格式的定义，再转换为多种格式。

而在写**源码**期间，由于是直接写的xml或sgml等文件，属于文本类的文件，所以可以很方便的实现版本的控制，由此实现了**Docbook的优点**

1.2. Docbook是什么

根据**Docbook官网的解释**¹，Docbook是一种schema。

schema，中文一般翻译为**模式**，但是如此解释，还是很抽象，很难理解。



关于Schema

- 关于DTD和XML Schema之间的区别，可以参看w3school的解释：[DTD vs XML Schema](http://www.w3school.com.cn/schema/schema_howto.asp)²
- 关于RELAX NG，可以参看wiki中的举例说明：[RELAX NG - XML syntax](http://en.wikipedia.org/wiki/RELAX_NG#XML_syntax)³

此处，可以简单的将Docbook理解为，类似于某种语言，比如英语。

¹ <http://www.docbook.org/whatis>

² http://www.w3school.com.cn/schema/schema_howto.asp

³ http://en.wikipedia.org/wiki/RELAX_NG#XML_syntax

英语，包括了对应的单词，词汇，以及对应的语法，规定了单词和词汇之间的关系，如何使用它们才是正确的。

对应的Docbook，也规定了一堆基本的单词和相应的语法。

1. Docbook中的单词 == 元素(Element)

Docbook中的单词，被称为元素（element），比如最常见的book，chapter，section等。

2. Docbook的中语法 == 用其他语言（Relax NG/SGML/XML DTDs/W3C XML Schema）来解释

Docbook中各个元素之间的关系，是Docbook规定好的。

此所谓的规定，就是用某种方法，解释出来，并且要解释的清楚易懂。

而直接用文字去描述的话，比如book下面可以嵌套多个chapter，chapter下面可以有多个section等，

首先真容易把Docbook的人累死，因为单纯用文字描述清楚如此复杂的关系，真的需要太多文字的解释。

其次，用纯文字描述，语义上不是足够明确。

而计算机领域内，有对应的，专门用于描述，类似于Docbook中的各个“单词”之间的关系的语言。

而Docbook就是采用他们，来解释清楚，Docbook中有哪些元素，以及这些元素之间的关系又是啥样的。

目前已经实现了的，包括Relax NG，SGML，XML DTDs，W3C XML Schema

通过如此描述，可以简介而精确的描述出Docbook的内容，即有哪些元素和元素之间的关系。



到底这些描述语言，比如DTD，XML Schema等，都是啥样子的

关于到底这些描述语言，都长的啥样子的，都是如何工作的，可以去看：[DTD vs XML Schema](#)⁴

看完后，就可以大概的了解到，这些描述语言，是怎么回事了。

而对于具体的更多的描述语言的细节，作为Docbook使用者来说，我们可以不必太关心。

我们需要关心的是，docbook中有哪些元素，以及他们之间的关系。然后如何去正确的使用它们。

所以，简单总结可以理解为：

Docbook是一种模式，其规定了包括哪些元素以及这些元素之间的关系。

而包含的元素以及元素之间的关系，是用对应的Schema Language去描述的。

目前已经Docbook实现了的Schema Language有：Relax NG，SGML，XML DTDs，W3C XML Schema

Docbook（虽然不仅仅只支持这些，但是）更加适用于计算机（软件和硬件）相关的书籍和文章。

Docbook的<=4.5的版本，很多都是用SGML语言书写Docbook源码的，现在已不再推荐。

Docbook最新版本是5.0。推荐使用XML语言书写Docbook的源码。

⁴ http://www.w3school.com.cn/schema/schema_howto.asp

1.2.1. 截图说明Docbook的：一种（xml）格式输入，多种（html，htmls，chm，pdf，txt，rtf等等）格式输出

第1节 “什么是Docbook” 中虽然用文字解释了什么是Docbook，但是还是不够直观。

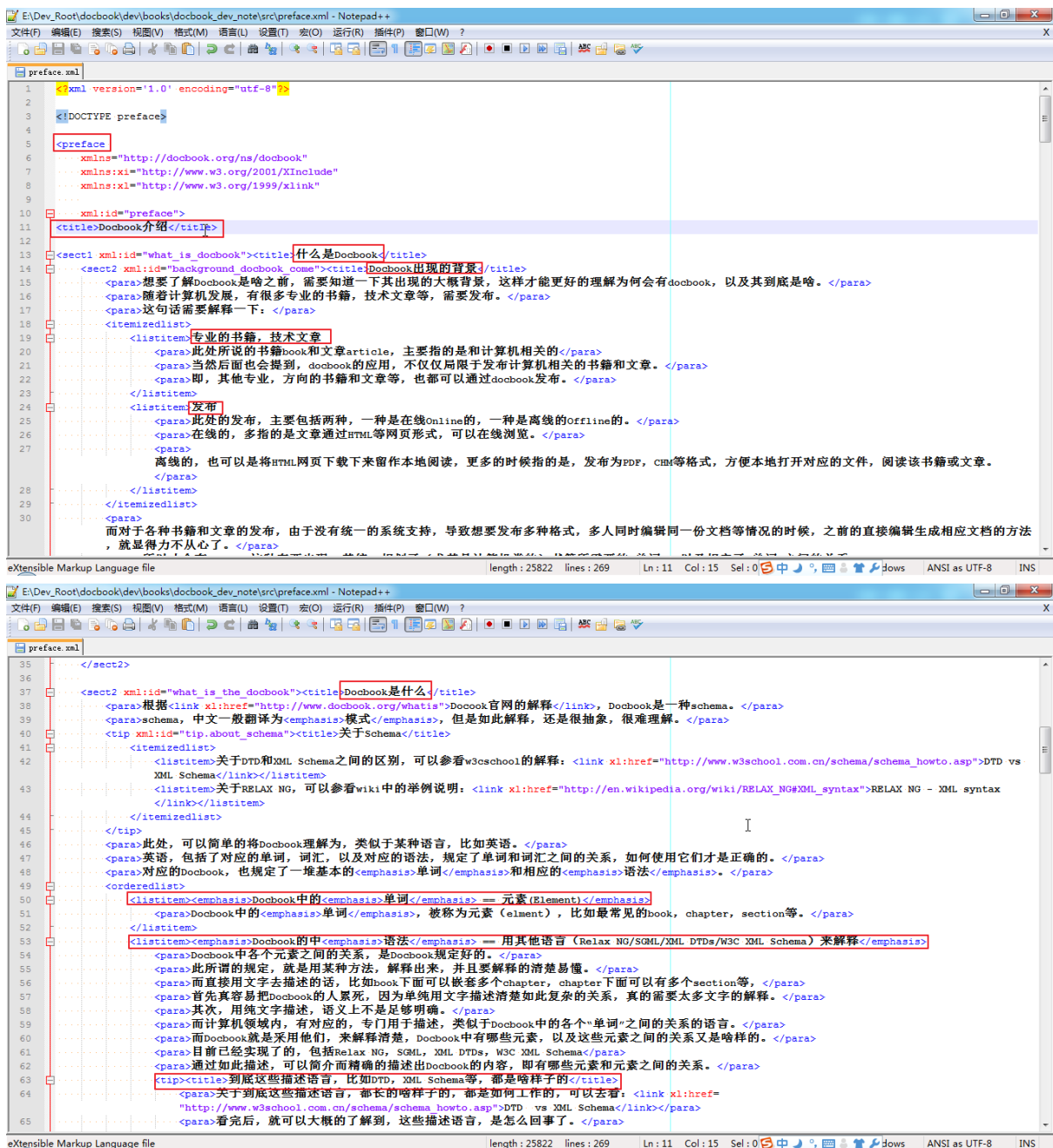
所以，此处专门通过截图来给不熟悉的人具体解释解释，Docbook长啥样，以及什么是一种输入，多种输出。

1.2.1.1. Docbook的一种输入：xml源码（文件）

Docbook的xml源码，就是常见的xml文件。

比如图1 “截图举例：Docbook的XML源码” 中的这段Docbook的xml源码，就是很常见的xml文件而已。

图1. 截图举例：Docbook的XML源码



1.2.1.2. Docbook的多种输出： HTML, HTMLs, PDF, CHM, RTF, TXT, WEBHELP等格式

上述的图 1 “截图举例：Docbook的XML源码”，可以通过工具，将其转换为多种输出格式。

我目前暂时只实现了其中的7种格式：

图 2. 截图举例：Docbook的可以生成多种格式

Docbook开发手记

版本：v1.9

Crifan Li

摘要

本文先简介了什么是Docbook,然后详细介绍了如何搭建中文的Docbook开发环境,再详细记录了Docbook开发过程中所Docbook中各种元素的示例代码,另外也记录了一些值得参考的Docbook相关的资料。

 本文提供多种格式供：

在线阅读	HTML	HTMLs	PDF	CHM	TXT	RTF	WEBHELP
下载（7zip压缩包）	HTML	HTMLs	PDF	CHM	TXT	RTF	WEBHELP

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/docbook_dev_note/

2013-10-04

下面分别简单介绍一下每种格式：

- HTML
单一HTML网页

包含了所有的内容

图 1 “截图举例：Docbook的XML源码” 所生成的HTML网页的效果：

图 3. 截图举例：Docbook的XML源码生成的HTML

1. 什么是Docbook

1.1. Docbook出现的背景

想要了解Docbook是啥之前，需要知道一下其出现的大概背景，这样才能更好的理解为何会有docbook，以及其到底是啥。

随着计算机发展，有很多专业的书籍，技术文章等，需要发布。

这句话需要解释一下：

- 专业的书籍，技术文章

此处所说的书籍book和文章article，主要指的是和计算机相关的

当然后面也会提到，docbook的应用，不仅仅局限于发布计算机相关的书籍和文章。

即，其他专业，方向的书籍和文章等，也都可以通过docbook发布。

- 发布

此处的发布，主要包括两种，一种是在线Online的，一种是离线的Offline的。

在线的，多指的是文章通过HTML等网页形式，可以在线浏览。

离线的，也可以是将HTML网页下载下来留作本地阅读，更多的时候指的是，发布为PDF，CHM等格式，方便本地打开对应的文件，阅读该书籍或文章。

而对于各种书籍和文章的发布，由于没有统一的系统支持，导致想要发布多种格式，多人同时编辑同一份文档等情况的时候，之前的直接编辑生成相应文档的方法，就显得力不从心了。

所以才会有Docbook这种东西出现。其统一规划了（尤其是计算机类的）书籍所需要的“单词”，以及规定了“单词”之间的关系，

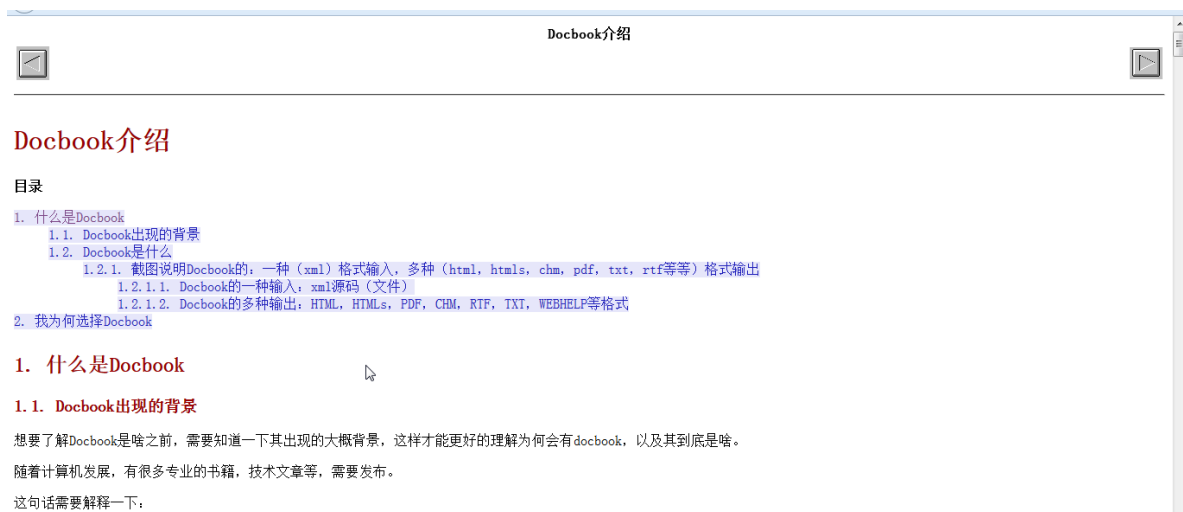
作为文档发布者，按照其规定，写好相应的源码

- HTMLs
多个HTML页面

每个章（甚至小节）都是一个单独的HTML网页

图 1 “截图举例：Docbook的XML源码”所生成的HTMLs网页的效果：

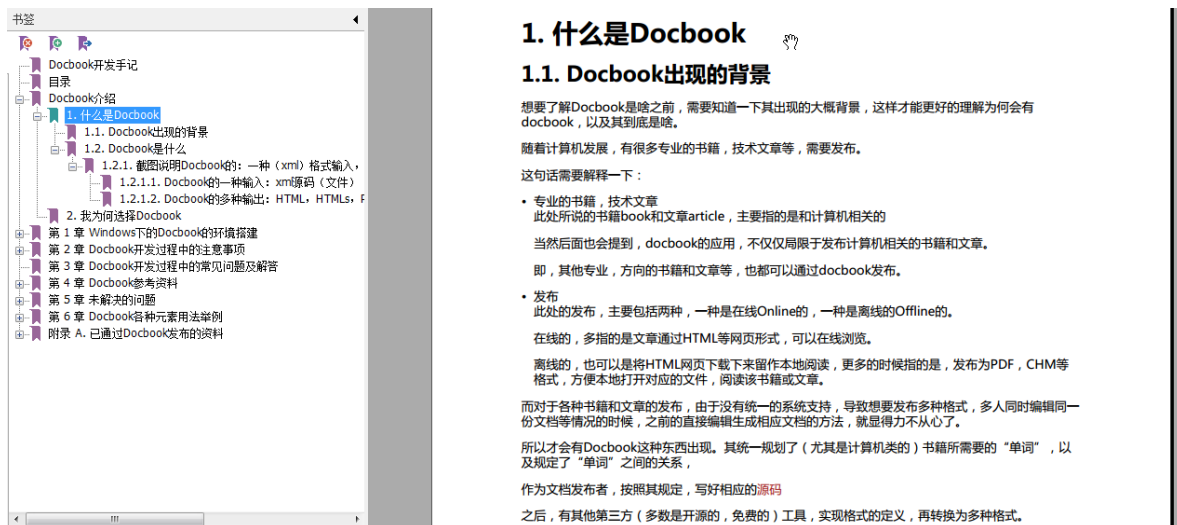
图 4. 截图举例：Docbook的XML源码生成的HTMLs



- PDF
PDF文档

图 1 “截图举例：Docbook的XML源码”所生成的PDF的效果：

图 5. 截图举例：Docbook的XML源码生成的PDF

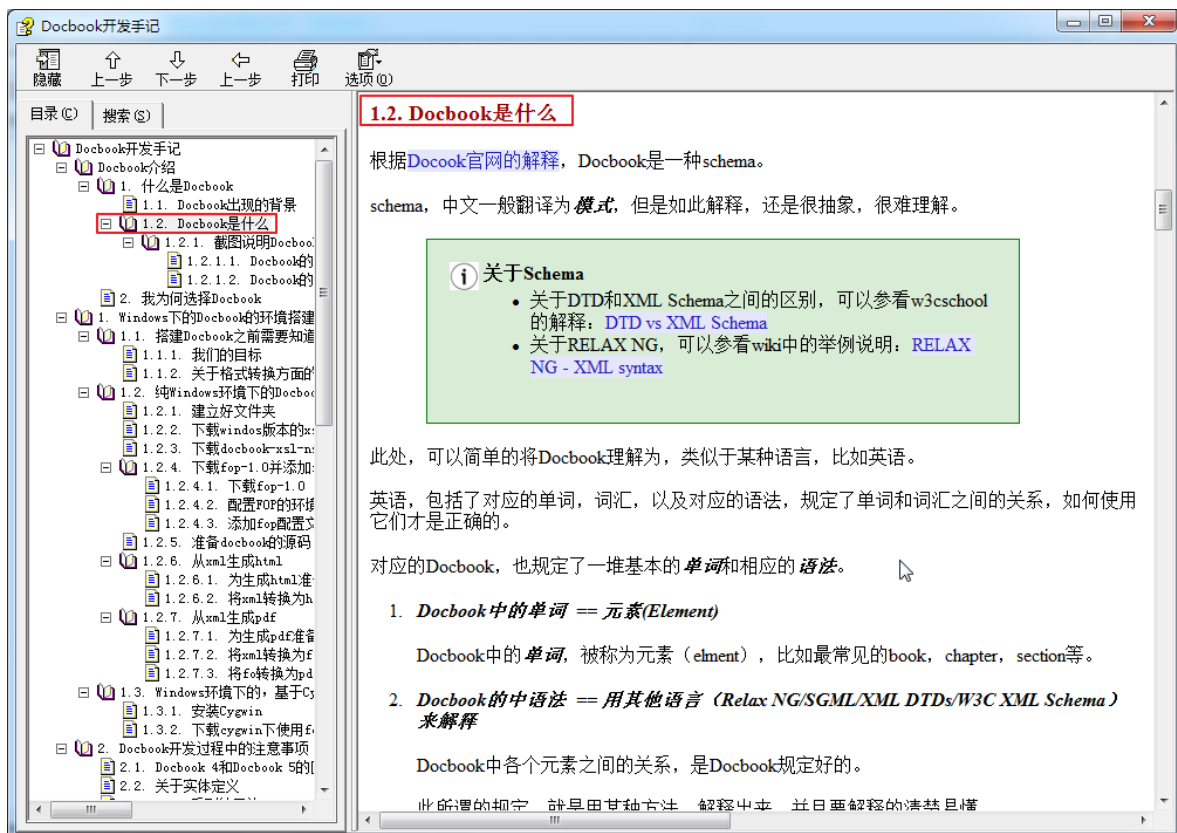


- CHM
docbook对应的是htmlhelp类型

生成的是微软的chm帮助文件

图 1 “截图举例：Docbook的XML源码”所生成的CHM的效果：

图 6. 截图举例：Docbook的XML源码生成的CHM

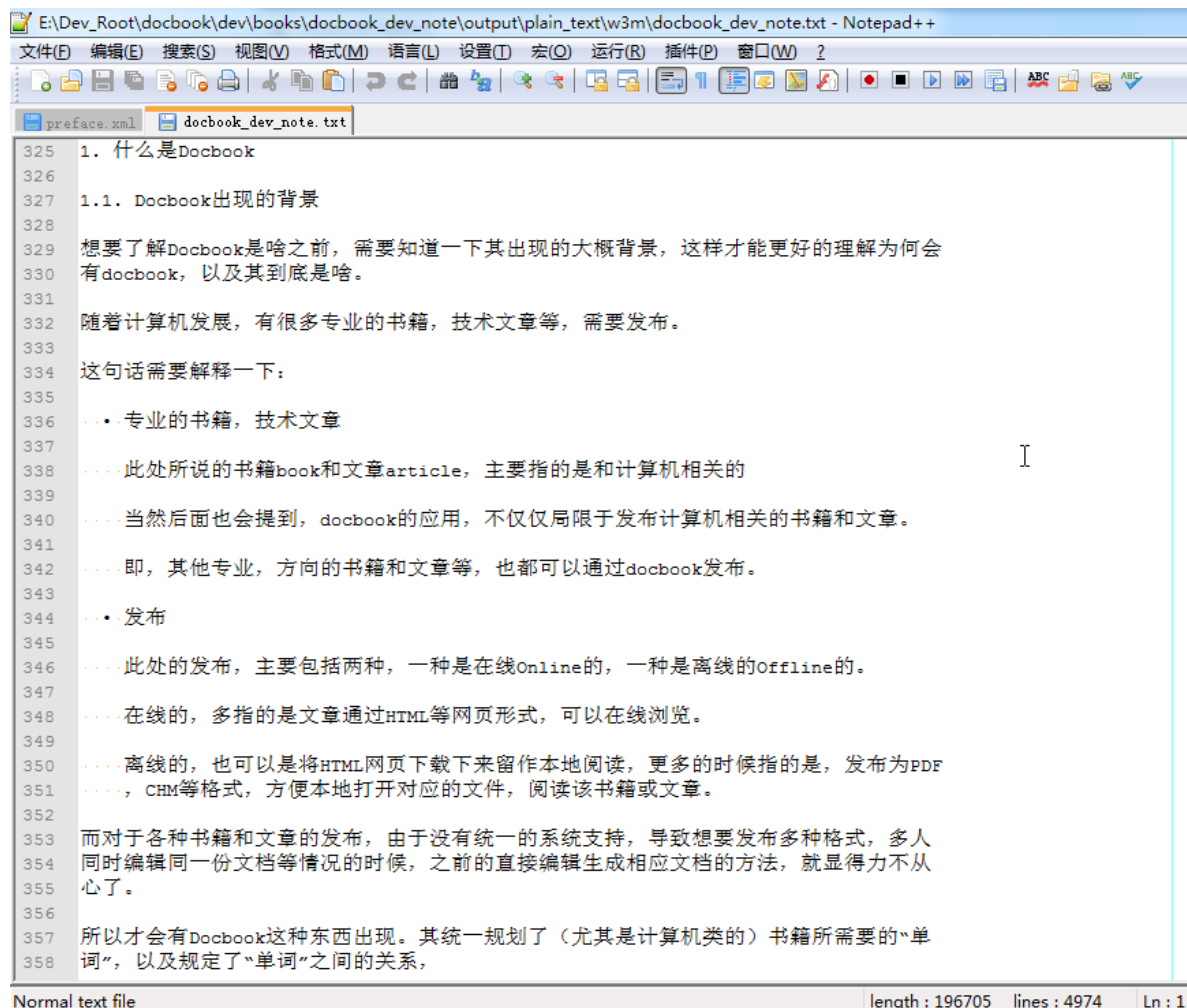


- TXT
纯文本格式

是用工具从HTML网页转换出来的

图 1 “截图举例：Docbook的XML源码”所生成的TXT的效果：

图 7. 截图举例：Docbook的XML源码生成的TXT



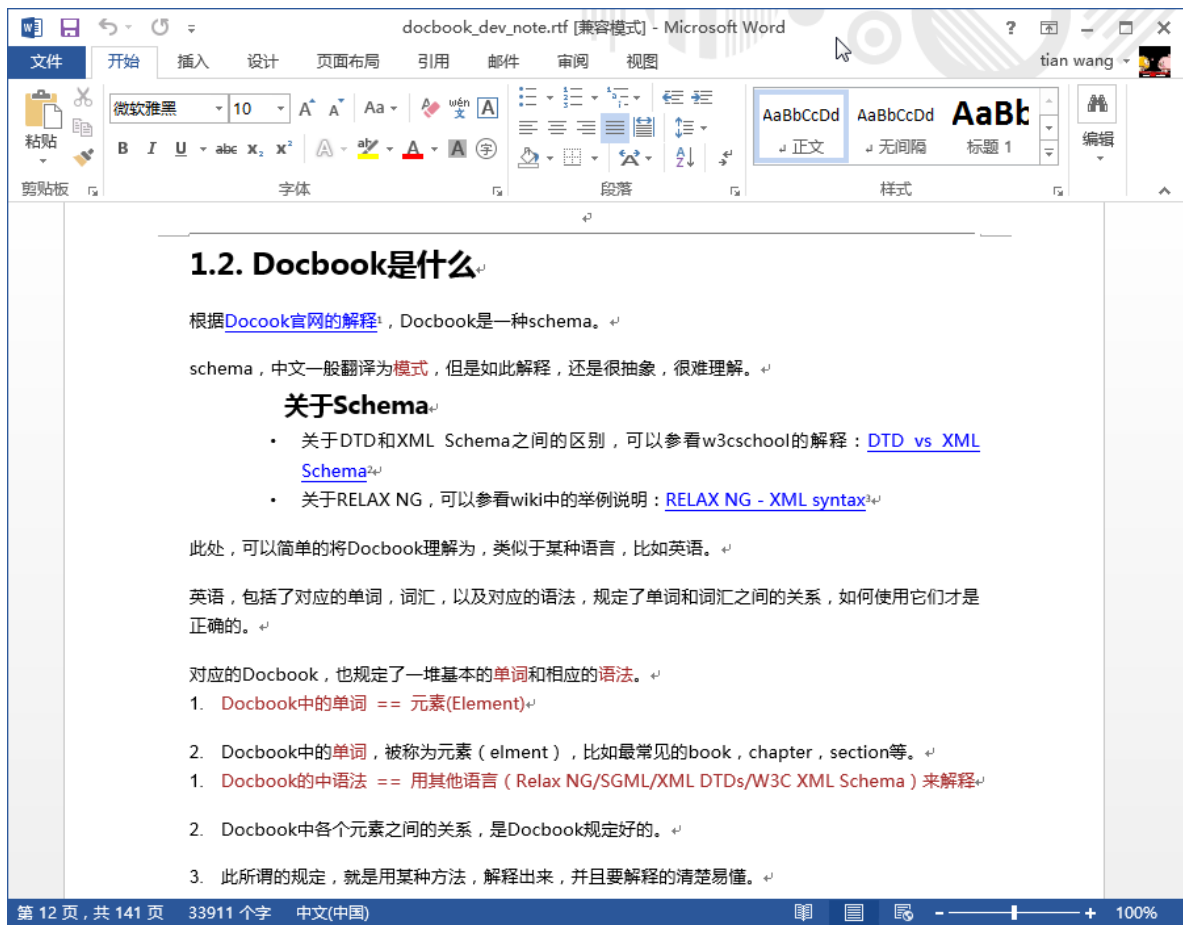
```
325 1. 什么是Docbook
326
327 1.1. Docbook出现的背景
328
329 想要了解Docbook是啥之前, 需要知道一下其出现的大概背景, 这样才能更好的理解为何会有docbook, 以及其到底是啥。
330
331 随着计算机发展, 有很多专业的书籍, 技术文章等, 需要发布。
332
333 这句话需要解释一下:
334
335
336 ...专业的书籍, 技术文章
337
338 ...此处所说的书籍book和文章article, 主要指的是和计算机相关的
339
340 ...当然后面也会提到, docbook的应用, 不仅仅局限于发布计算机相关的书籍和文章。
341
342 ...即, 其他专业, 方向的书籍和文章等, 也都可以通过docbook发布。
343
344 ...发布
345
346 ...此处的发布, 主要包括两种, 一种是在线Online的, 一种是离线的Offline的。
347
348 ...在线的, 多指的是文章通过HTML等网页形式, 可以在线浏览。
349
350 ...离线的, 也可以是将HTML网页下载下来留作本地阅读, 更多的时候指的是, 发布为PDF
351 ...CHM等格式, 方便本地打开对应的文件, 阅读该书籍或文章。
352
353 而对于各种书籍和文章的发布, 由于没有统一的系统支持, 导致想要发布多种格式, 多人
354 同时编辑同一份文档等情况的时候, 之前的直接编辑生成相应文档的方法, 就显得力不从
355 心了。
356
357 所以才会有Docbook这种东西出现。其统一规划了(尤其是计算机类的)书籍所需要的“单
358 词”, 以及规定了“单词”之间的关系,
```

- RTF
RTF文档, 和word兼容

可以用微软的word软件打开

图 1 “截图举例：Docbook的XML源码”所生成的RTF的效果：

图 8. 截图举例：Docbook的XML源码生成的RTF



- WEBHELP
Webhelp格式

图 1 “截图举例：Docbook的XML源码”所生成的WEBHELP的效果：

图 9. 截图举例：Docbook的XML源码生成的WEBHELP



其他docbook还可以生成其他格式，比如 website, slide, roundtrip, manpage, xhtml, epub, javadoc, eclipse help等，此处暂时没有去弄而已。

通过上述截图的介绍，相信即使不熟悉的人，也大概知道docbook是个什么东西了。

2. 我为何选择Docbook

主要是之前写的不少帖子，都是用word编辑，然后转换为PDF后发布的，但是此法存在几个问题

1. 转换后的PDF不完美

此处说的不完美，不是表面看上去的效果不好，而是pdf本身内部所包含的内容，有点瑕疵。

主要是，当你选择一些内容，然后拷贝出来，粘贴到别的地方，比如word中，你会发现，几乎每一行的内容，都是对应的两行，第二行是重复第一行的，而且拷贝出来的内容，格式也是很乱的。

这让我很不爽，因为我要追求的是完美的效果，即你用Word 2007/2010导出的PDF，也应该是在内部的内容和格式上，都是和原先word一致的才对。

所以，后来有机会知道了Docbook，才决定不用word了

2. word的不爽

• 稳定性

word作为基本的文字编辑工具，总体上来说，还是挺不错的，至少对于多数人如此。

而对于我，有时候会遇到单个word文件，包含内容太多，比如99页，而遇到过，在编辑过程中偶尔死掉，导致差点丢失了一些已经编辑的内容。

需要说明的是，我对于使用word和其他工具，已经养成了够好的习惯了，因为是编辑了任何地方，都是顺便Ctrl+S去保存一下的。保证不会出现由于没有及时保存而丢失内容的情况。

但是对于上述所说的word的偶尔死掉而导致文件内容丢失的可能性，却是无法排除的。

所以，还是觉得不太靠谱。

• 兼容性

微软的东西，兼容性已经做得足够好了，包括Office系列工具中的word

但是对于之前使用word所遇到的情况，比如2003，2007和2010等的兼容程度，以及从长远来看，都无法排除，用之前某个版本的word写的东西，无法用新的版本正常打开，且完全兼容旧版本，完全保留之前的所有的内容和设置。

所以，保险起见，还是换用Docbook等xml源码的形式，不用担心在未来任何时刻，都可以完整的打开原始文档，保留所有原始内容和设置。

• 格式排版

如果只是写写一般短内容的帖子和文章，word的格式，排版等功能，算是非常好用的了。

但是如果是写的内容非常多，涉及到各个部分的格式调整，即使用了word的模板去统一格式，也显得很繁琐。

因此，内容比较多的话，用word去写，而且还要统一风格和设置，就显得效率比较低了。此时，换用Docbook等，就是很好的选择了。

3. 发布格式单一

之前的做法，都是用word写好，然后再转换为PDF。而新的2007/2010的word，支持直接导出PDF，而且已经支持页内链接，标签等（详见之前的帖子：[【整理】制作pdf文件后，保持原先](#)

[word文档中的页内链接和超链接+常见pdf打印机安装使用方法简介⁵](#)), 已经算是蛮方便的了, 但是还是只能发布PDF一种格式。

而发布PDF格式的文件, 好处是利于下载阅读, 而且无法更改原文内容, 坏处是无法方便的拷贝粘贴所需要的内容, 不利于内容的传播和分享。

因此, 希望支持更多的格式, 尤其是在线的HTML页面。

而后来自从知道了Docbook, 才发现, 终于找到了我所想要的, 同样的原文内容(xml源码), 可以支持多种格式发布, 至少包括HTML和PDF, 这样就方便发布文章, 利于文章内容传播了。

然后就是遇到了Docbook, 基本上完美解决了我之前的所有问题, 相对有很多好处:

Docbok的优点.

1. 一种输入, 多种输出

这个优点, 对于我来说, 是最看重的。

即, 写任何帖子, 文章, 的时候, 只需要写好对应的xml源码, 然后用已经搭建好的环境, 编译生成多种格式。

我目前已经搭建好的环境, 是在Windows下面的Cygwin, 已经自己写好了Makefile, 这样, 每次写完xml源码, 直接make, 就可以生成多种版本了, 目前已经实现的有: 单个的HTML网页文件, 多个的HTML网页文件, PDF, RTF (即word文档, 用word可以直接打开和编辑的), TXT的纯文本文件, 微软的帮助文件CHM

这样, 就方便了文章以多种格式发布, 和统一设置各种风格(显示效果)了。

此所谓, 一种输入, 多种输出。

2. 可以实现版本控制

关于版本控制这一点, 前面说word的缺点的时候, 其实都没提到。那是因为此处我对写的帖子的版本控制, 需求还不是特别高。

但是的确, 对于更加专业的人员, 多人协作写同一本书, 这种时候, 版本控制, 就显得尤为重要了。

Docbook是原始输入, 就是xml源码, 都属于普通的文本Plain Text, 所以很方便使用SVN等版本控制工具, 对齐实现版本控制。

这样, 任何时候的任何改动, 都可以实现追溯和管理了。

当然, 相对来说, Docbook, 至少对于我来说, 也有一些缺点:

Docbok的缺点.

1. 前期的入门有点难

此处所说的Docbook入门, 指的是两方面

- 搞懂Docbook是啥, 有点难
之前是一直多Docbook是啥, 没太搞懂。

其实现在也不是完全清楚, 因为想彻底搞清楚, 得多花时间去搞懂很多概念, 包括DTD, Schema等等。

不过, 想要达到基本了解, 够用, 其实也就行了。更复杂的东西, 可以在折腾的过程中去多了解了解。

- 真正完全把Docbook的环境搭建好, 也不容易

⁵ http://www.crifan.com/order_make_pdf_file_to_keep_the_original_word_document_page_links_and_hyperlinks_within_the_common_pdf_printer_installation_method_introduction/

其实折腾完毕后，才知道，想要搭建最基本的Docbook环境，能生成HTML和PDF，对于Docbook 5，其实并不难。

但是对于之前最开始接触的那个Docbook 4.5以及之前版本，其实还是蛮麻烦的。

然后搭建了最基本的Docbook 5的环境，在继续定制自己的配置，是需要一点点学习，了解，才能搞定如何配置的。还是需要花些精力的。

当前，搭建好了自己的Docbook开发环境后，以后写完帖子的发布工作，那就简单了，直接Make或者输入命令，实现转换即可。

2. 缺少所见即所得(WYSIWYG)



WYSIWYG

WYSIWYG是What You See Is What You Get的缩写。

意思很清楚，就是，你所看到的，就是你所得到的。

我们常见的微软的很多工具，包括word，就属于WYSIWYG，因为你是正在编辑word文档的时候，所看到的内容，即除了本身输入的文字之外，还有那些已经设置好了的字体，颜色，排版等，都是和你最终所得到的结果，即你再次打开或别人拿到你的word文件去打开，所看到的效果是一样的。所以称，所见即所得。

Docbook本身的核心就在于，非所见即所得，内容和形式（显示）分开，即你所输入的文章的内容，都是在xml源码中，而具体对应的这些内容的显示效果，都是由相关的配置文件和参数，有工具转换后，才能看到具体显示效果的。即本身就不是所见即所得的。

而此处之所以把WYSIWYG列为Docbook的缺点之一，就是因为我们很多人，习惯了微软的Word等所见即所得的环境，突然换成了内容与显示效果分开的Docbook的环境，还真的需要一段时间的适应。

另外，至于我自己，有时候，也的确想要去控制某些内容的显示效果，而无法很方便的配置出来，的时候，也偶尔会觉得，Docbook（加上配置文件和转换工具）在（所得到的）显示效果方面，的确是没有word方便。

举个例子，就比如我想要源码高亮显示，目前所能找到的资料，除了官网的解释，是除了需要开启相应高亮的参数配置之外，还要添加相应extension，然后还需要是Saxon等环境和工具才可以，对于我所用的xsltproc+fop，竟然还不支持，所以很悲催的暂时放弃了代码高亮的功能。详见：[【已放弃】给Docbook中代码programlisting实现高亮显示](http://www.crfan.com/docbook_highlight_code_for_programlisting_using_xsltproc_fop/)⁶

不过总的来说，类似于这些显示效果的控制方面，即使有时候没法实现或者不方便实现，在保证内容完整的前提下，没了这些效果，基本也是可以接受的。

⁶ http://www.crfan.com/docbook_highlight_code_for_programlisting_using_xsltproc_fop/

第 1 章 Windows下的Docbook的环境搭建

1.1. 搭建Docbook之前需要知道的最基本的事情

1.1.1. 我们的目标

首先要清楚，我们想要得到什么。

我们想要得到的就是，写了符合Docbook规定的xml源文件之后，然后用工具可以生成多种格式的输
出，此所谓“一种输入，多种输出”。

为了达到此目标，所以才要搭建一个docbook环境，配置一些东西，使用一些工具，将xml文件转换为
HTML，PDF等多种格式。

简单的说，此时我们的目的就是，搭建一个最基本的Docbook环境，至少先支持这些基本的功能：

1. 可以将Docbook的xml文件转换生成为HTML和PDF两种格式
2. 支持中文字体
3. 支持多个分割的（单独的）文件
此种情况主要适用于，当你写了一本内容很多的书籍的话，肯定有必要将其分为多个章节
而每个章节，希望放在一个单独的xml文件中，然后用一个总的xml，将每个章节包含进来。
此时，这多个独立的章节所对应的各个xml文件，就称为多个分割的（xml）文件

1.1.2. 关于格式转换方面的知识

将docbook的xml源码，转换为HTML，PDF等格式的话，有多种方法，其中涉及多种工具。

此处暂时不需要理会太多，只需要知道这两点即可：

1. 如何得到HTML
xsltproc可以将xml文件转换为HTML文件
2. 如何获得PDF
xsltproc将xml转换为FO文件
然后用fop将FO文件转换为PDF文件

下面要介绍的docbook的环境，也都是关于xsltproc和fop这两方面。

1.2. 纯Windows环境下的Docbook开发环境的搭建



docbook demo所有相关文件下载

和下面纯windows下docbook环境搭建有关的文件，可以在这里下载
到：[docbook5_demo_noTools.7z](http://crifan.com/files/res/docbook/demo/docbook5_demo_noTools.7z)¹

¹ http://crifan.com/files/res/docbook/demo/docbook5_demo_noTools.7z

此处介绍如何在纯Windows环境下（即不需要额外安装Cygwin等），而搭建最基本的Docbook环境。

使得可以生成对应的HTML和PDF。

以下内容，主要参考：[一个简单的Docbook 5.0例子](http://easwy.com/blog/archives/a-simple-docbook-5-example/)²和[在Windows上安装配置Apache FOP](http://easwy.com/blog/archives/install-apache-fop-on-windows/)³

此处之所以重新介绍一遍，主要目的在于，当前使用的fop 1.0的版本，可以省略关于fop字体设置的部分，以减少步骤。

1.2.1. 建立好文件夹

为了以后更好的使用和扩展Docbook环境，所以此处专门设计了一个简单的架构。

即，不是为了演示，就随便糊弄一下。

下面的架构搭好之后，真正想要使用docbook的人，就可以在此基础上，一点点的扩展功能了。

当然，此处搭建的环境，也尽量的保持简洁，此处暂时只支持生成我们所需要的HTML和PDF。

下面就是要搭建的docbook环境所对应的文件夹结构

请自行在某个文件夹，比如我的是E:\DevRoot\docbook,下面建立对应的文件夹。

```
books①
|
|-- docbook5_demo②
|   |
|   |-- output③
|   |   |
|   |   |-- fo④
|   |   |-- html⑤
|   |   |-- pdf⑤
|   |
|   |-- src⑥
|
config⑦
|
|-- docbook-xsl-ns-1.77.1⑧
|-- fop⑨

tools
|
|-- docbook-xsl-ns-1.77.1⑩
|   |
|   |-- fo
|   |-- html
|   ...
|
|-- fop-1.0⑪
|   |
|   |-- build
|   |-- conf
```

² <http://easwy.com/blog/archives/a-simple-docbook-5-example/>

³ <http://easwy.com/blog/archives/install-apache-fop-on-windows/>

```
| |-- ...  
| |  
|-- xslt⑩  
| |  
| |-- bin  
| |-- include  
| |-- lib
```

- ❶ 用于存放多个Docbook的book（书籍）

后期扩展：可以建更多的book，比如我当前的关于docbook开发笔记，就可以对应新建一个docbook_dev_note的book了。

- ❷ 接下来所要写的docbook 5的demo的book

- ❸ 用于存放各种格式的输出。

当前暂时为了演示，只建了fo，html，pdf

后期扩展：可以建更多的文件夹，比如htmls，rtf，chm，txt等存放更多的输出格式

- ❹ 用于存放fo文件。其可以被fop等工具转换为PDF，RTF等格式

- ❺ 用于存放最终生成的html，pdf等格式的文件。这些，是我们所想要的最终的结果。

- ❻ 用于存放docbook的xml源码文件。

后期扩展：当单个xml内容太多，可以考虑用xinclude分出多个xml文件，此时多个xml文件，都可以都放在此src文件夹下了。

- ❼ 用于存放所有的和Docbook相关的配置文件，包括xsl格式stylesheet文件，以及其他如fop的配置文件等。

后期扩展：其他的，比如，实体定义entity，catalog等等一些内容，也可以放在此config下。

- ❽ 此处之所以文件夹叫做docbook-xsl-ns-1.77.1，是由于最新的stylesheet是1.77.1版本的。

其作用是，设置对应的参数，可以控制html，pdf等输出的效果，包括显示效果方面的一些控制参数和输出内容格式组织方面的。

- ❾ 此处存放fop的配置。

目前暂时只需要一个fop.xconf文件即可。

- ❿ 下载并解压docbook-xsl-ns-1.77.1，然后放到此处即可。

如何下载docbook-xsl-ns-1.77.1参见[第 1.2.3 节 “下载docbook-xsl-ns-1.77.1”](#)

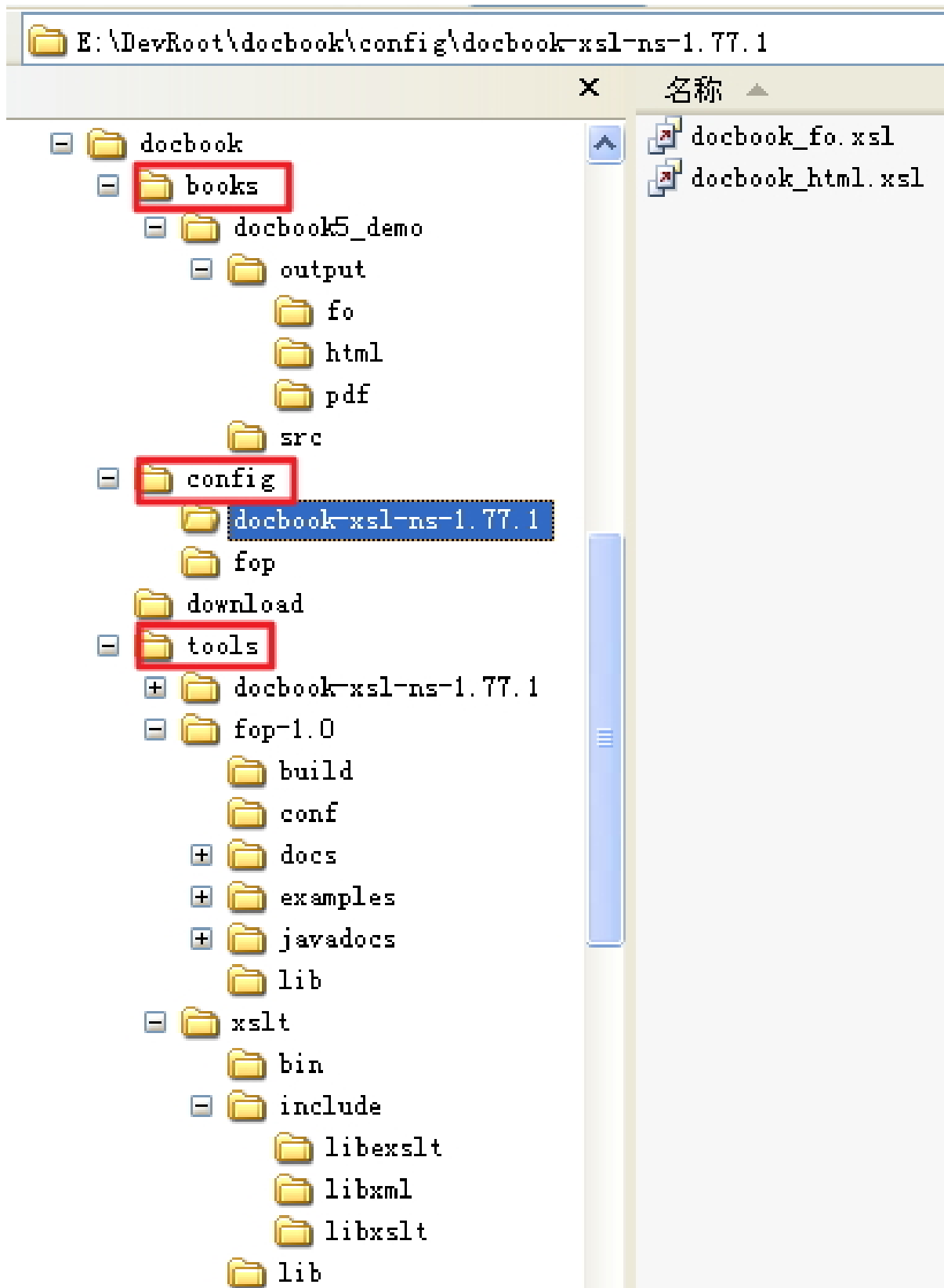
- ❶ 下载fop 1.0并解压，放到此处即可。

如何下载fop 1.0参见[第 1.2.4.1 节 “下载fop-1.0”](#)

- ❷ 下载xsltproc及相关文件，并解压，放到此处即可。

如何下载xsltproc及相关文件，参见[第 1.2.2 节 “下载windos版本的xsltproc”](#)

顺便贴出后来建立好的docbook环境的文件夹结构的截图：



安装上述介绍，建立好对应的文件夹后，就可以参考下面的步骤，下载相关文件，添加相关配置，编译生成html和pdf，然后最终搭建好docbook环境。

1.2.2. 下载windos版本的xsltproc

在：[xsltproc主页](#)⁴ ⇒ [下载页面](#)⁵ ⇒ [Libxml](#)⁶ ⇒ [Igor Zlatkovic提供的Windows下二进制版本的xsltproc](#)⁷中，去下载：

1. xsltproc主程序
[libxslt-1.1.26.win32.zip](#)⁸

其中包含我们所需要的xsltproc可执行文件：xsltproc.exe

2. 运行xsltproc所需要的一些库
为了在windows下面运行xsltproc，还需要一些其所依赖的库：libxml2，iconv，zlib

下面依次贴出下载地址和简单的解释：

- [libxml2-2.7.8.win32.zip](#)⁹

C语言实现的XML解析器。

其支持N多规范或协议，比如XML，XML-NS，XPath，XPointer，XInclude，UTF-8/UTF-16，XML Catalog，Canonical XML，Relax NG等。

- [iconv-1.9.2.win32.zip](#)¹⁰

字符编码转换。比如从UTF-8转换为GB18030，就可以用iconv。

- [zlib-1.2.5.win32.zip](#)¹¹

压缩工具。

下载了这四个文件后，解压，然后都会得到一个包括bin，include，lib三个文件夹。

全部都合并到一起后，放到上述介绍的📁中，即可。

把xsltproc的路径添加到PATH环境变量中。 为了可以在windows的命令行cmd中直接输入xsltproc就可以执行，而不需要绝对路径，此处需要把xsltproc所在路径，我此处的是：E:\DevRoot\docbook\tools\xslt\bin，添加到windows的环境变量PATH中去。



如何设置windows环境变量

关于如何添加相应路径到windows的环境变量中，不熟悉的可以参考：

[Windows的环境变量](#)¹²



更新版本的xsltproc相关的二进制文件

如果想要用更新版本的xsltproc，即相关的各个所依赖的二进制文件，想要下载更新版本的话，

可以去参考：

⁴ <http://xmlsoft.org/xslt/index.html>

⁵ <http://xmlsoft.org/xslt/downloads.html>

⁶ <http://www.zlatkovic.com/libxml.en.html>

⁷ <ftp://ftp.zlatkovic.com/libxml/>

⁸ <ftp://ftp.zlatkovic.com/libxml/libxslt-1.1.26.win32.zip>

⁹ <ftp://ftp.zlatkovic.com/libxml/libxml2-2.7.8.win32.zip>

¹⁰ <ftp://ftp.zlatkovic.com/libxml/iconv-1.9.2.win32.zip>

¹¹ <ftp://ftp.zlatkovic.com/libxml/zlib-1.2.5.win32.zip>

¹² http://www.crfan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#win_env_var

【已解决】docbook中用xsltproc处理xml文件时用xsl:import引用本地css, js等文件时, 必须加上file:///或file://的前缀才可以¹³

中的"如何组合出来相对比较新的版本的xsltproc"即可。

1.2.3. 下载docbook-xsl-ns-1.77.1

为了将xml源码转换为html, fo等文件的话, 需要相应的配置文件, 以控制输入的效果和格式, 这就需要相应的配置文件。

此处的配置文件, 就是xsl格式的stylesheet (样式表)。

docbook官方网站中, 有对应的最新的办法的xsl样式表文件, 是放在sourceforge上的: [最新的docbook-xsl-ns所在的页面](#)¹⁴

截止当前2012-06-10, 最新版本的是[1.77.1版本的docbook-xsl-ns](#)¹⁵, 下载下来对应的docbook-xsl-ns-1.77.1.tar.bz2后, 解压, 放到上面刚建立的⑩中

1.2.4. 下载fop-1.0并添加fop配置文件



如果没有安装java, 请先安装java

fop的运行, 依赖于java运行环境。

如果你的windows中, 还没有安装java, 请先安装java运行环境 (JRE=java runtime environment)。

关于java的基本知识和如何安装, 不了解的请参考: [Java](#)¹⁶

java已经安装好前提下, 另外记得确保JAVA_HOME环境变量已经正确设置好了。

比如, 我的系统中的是:

```
JAVA_HOME=D:\Program Files\Java\jre7
```

1.2.4.1. 下载fop-1.0

从: [fop主页](#)¹⁷ ⇒ [Apache™ FOP Version 1.0](#)¹⁸ ⇒ [fop下载页面](#)¹⁹ ⇒ [FOP Distribution mirror](#)²⁰ 中, 可以找到第一链接, 是官网推荐给你的, 然后进去后可以找到binaries文件夹, 进去后, 下载到fop-1.0-bin.tar.gz, 比如我这里得到的地址是: [bjtu的fop-1.0-bin.tar.gz](#)²¹

下载解压后, 放到上面建的⑩中即可。

1.2.4.2. 配置FOP的环境变量FOP_HOME

为了更好的让fop自动搜索对应的lib库, 需要去设置fop的一个环境变量FOP_HOME。

此处即为, 设置对应环境变量为:

¹³ http://www.crifan.com/cygwin_docbook_refer_local_file_must_use_file_colon_three_slash/

¹⁴ <http://sourceforge.net/projects/docbook/files/docbook-xsl-ns/>

¹⁵ <http://sourceforge.net/projects/docbook/files/docbook-xsl-ns/1.77.1/>

¹⁶ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#java

¹⁷ <http://xmlgraphics.apache.org/fop/>

¹⁸ <http://xmlgraphics.apache.org/fop/1.0/index.html>

¹⁹ <http://xmlgraphics.apache.org/fop/download.html>

²⁰ <http://www.apache.org/dyn/closer.cgi/xmlgraphics/fop>

²¹ <http://mirror.bjtu.edu.cn/apache/xmlgraphics/fop/binaries/fop-1.0-bin.tar.gz>

FOP_HOME=E:\DevRoot\docbook\tools\fop-1.0

1.2.4.3. 添加fop配置文件fop.xconf

将原先的❶中的conf文件夹中的fop.xconf拷贝一份出来，放到❷下

然后打开该文件，找到fop⇒renderers中的fonts部分，然后添加对应的配置，变成：

```
<fop version="1.0">
  ...
  <renderers>
    <renderer mime="application/pdf">
      ...
      <fonts>
        ...

        <!-- new added start -->
        <!-- register all the fonts found in a directory and all of its sub directories (use with care)
-->
        <directory recursive="true">file:///c:/windows/fonts/</directory>

        <!-- automatically detect operating system installed fonts -->
        <auto-detect/>
        <!-- new added end -->
      </fonts>
    </renderer>
  </renderers>
</fop>
```

即完成了对应的配置。



关于FOP的字体配置

关于fop的更多字体配置方面的内容，可以去参考官网的解释：[Custom Fonts](http://xmlgraphics.apache.org/fop/1.0/fonts.html#custom)²²

1.2.5. 准备docbook的源码：xml文件

将如下内容：

```
<?xml version='1.0' encoding="utf-8"?>

<book version="5.0"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xml:lang="zh_CN"
>

<info>
  <title>Docbook5环境搭建演示</title>
  <abstract>
    <para>本文主要介绍了如何在纯Windows的环境下搭建Docbook 5开发环境</para>
  </abstract>
</info>
```

²² <http://xmlgraphics.apache.org/fop/1.0/fonts.html#custom>

```
<author> <personname> <firstname>Crifan</firstname> <surname>Li</surname> </
personname> </author>
<pubdate>2012-06-05</pubdate>
<copyright>
  <year>2012</year>
  <holder>Crifan</holder>
  <holder> <link xl:href="http://crifan.com" /> </holder>
</copyright>
<legalnotice>
  <para>本文章遵从：<link xl:href="http://creativecommons.org/licenses/by-nc/2.5/cn/">
署名-非商业性使用 2.5 中国大陆(CC BY-NC 2.5)</link> </para>
</legalnotice>
</info>

<chapter> <title>第一章的标题</title>
  <para>关于更多的Docbook方面的内容，可以去参考：<link xl:href="http://www.crifan.com/
files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html">Docbook开发手
记</link> </para>

  <sect1> <title>第一章第一小节的标题</title>
    <para>可以在这里输入第一章第一小节中的内容</para>

    <sect2> <title>第一章第二小节的标题</title>
      <para></para>
    </sect2>
  </sect1>

  <sect1> <title>第一章第二小节标题</title>
    <para></para>
  </sect1>
</chapter>

<chapter> <title>第二章</title>
  <para></para>
</chapter>

</book>
```

存为UTF-8编码的docbook5_demo.xml文件。

用于后期演示使用。



如何建立UTF-8编码的文件

可以参考[Notepad++的多种编码支持](http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/htmls/npp_func_multi_enc.html)²³，去用Notepad++建立相应的UTF-8编码的文件。

1.2.6. 从xml生成html

1.2.6.1. 为生成html准备配置文件

将如下内容：

²³ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/htmls/npp_func_multi_enc.html

```
<?xml version='1.0'?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version='1.0'>

<xsl:import href="E:/DevRoot/docbook❶/tools/❷docbook-xsl-ns-1.77.1/html/docbook.xsl"/>

<!--=====
=====
font setting
=====
=====-->
<xsl:param name="title.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="body.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="monospace.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="symbol.font.family">Cambria Math</xsl:param>

<!--=====
=====
TOC index setting
=====
=====-->
<xsl:param name="section.autolabel" select="1"/>
<xsl:param name="section.label.includes.component.label" select="1"/>
<xsl:param name="bibliography.numbered" select="1"/>
<xsl:param name="toc.section.depth">8</xsl:param>

</xsl:stylesheet>
```

- ❶ 此处请自行将E:/DevRoot/docbook换为你自己当前的目录。
- ❷ 注意，此处的路径中是正斜杠 '/'，而不是反斜杠 '\', 否则会导致xsltproc无法识别。

存为UTF-8格式的docbook_html.xml

放到前面建的❶中。

1.2.6.2. 将xml转换为html

将xml源码转换为html话，只需要用xsltproc一步操作就可以了：

打开windows的命令行cmd，切换到上面建的❶下

然后运行如下命令：

```
xsltproc -o ../output/html/docbook5_demo.html E:\DevRoot\docbook\config\docbook_html.xml docbook5_demo.xml
```

即可生成相应的docbook5_demo.html

截图效果如下：

Docbook5环境搭建演示

Crifan Li

版权 © 2012 Crifan, <http://crifan.com>

本文章遵从: [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)

2012-06-05

摘要

本文主要介绍了如何在纯Windows的环境下搭建Docbook 5开发环境

目录

- [1. 第一章的标题](#)
 - [1.1. 第一章第一小节的标题](#)
 - [1.1.1. 第一章第二小节的标题](#)
 - [1.2. 第一章第二小节标题](#)
- [2. 第二章](#)

第 1 章 第一章的标题

目录

1.2.7. 从xml生成pdf

1.2.7.1. 为生成pdf准备配置文件

将如下内容：

```
<?xml version='1.0'?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version='1.0'>

<xsl:import href="E:/DevRoot/docbook/tools/docbook-xsl-ns-1.77.1/fo/docbook.xsl"/>
```

```
<!--=====
=====
font setting
=====
=====-->
<xsl:param name="title.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="body.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="monospace.font.family">Microsoft YaHei</xsl:param>
<xsl:param name="symbol.font.family">Cambria Math</xsl:param>

<!--=====
=====
TOC index setting
=====
=====-->
<xsl:param name="section.autolabel" select="1"/>
<xsl:param name="section.label.includes.component.label" select="1"/>
<xsl:param name="bibliography.numbered" select="1"/>
<xsl:param name="toc.section.depth">8</xsl:param>

<!--=====
=====
misc setting
=====
=====-->
<!-- add bookmark and support more type of figure -->
<xsl:param name="fop1.extensions">1</xsl:param>

<!-- no indent for body content -->
<xsl:param name="body.start.indent">0pt</xsl:param>

</xsl:stylesheet>
```

存为UTF-8格式的docbook_fo.xsl

同理，不要忘了参考上面的❶，把E:/DevRoot/docbook换为你自己当前的目录。

把docbook_fo.xsl放到前面建的❸中。

1.2.7.2. 将xml转换为fo

同理，在❹下

用下列命令去生成fo：

```
xsltproc -o ../output/fo/docbook5_demo.fo E:\DevRoot\docbook\config\docbook-xsl-ns-1.77.1\docbook_fo.xsl docbook5_demo.xml
```

相应的屏幕输出为：

```
E:\DevRoot\docbook\books\docbook5_demo\src>xsltproc -o ../output/fo/docbook5_demo.fo
E:\DevRoot\docbook\config\docbook-xsl-ns-1.77.1\docbook_fo.xsl docbook5_demo.xml
Making portrait pages on USletter paper (8.5inx11in)
```

1.2.7.3. 将fo转换为pdf

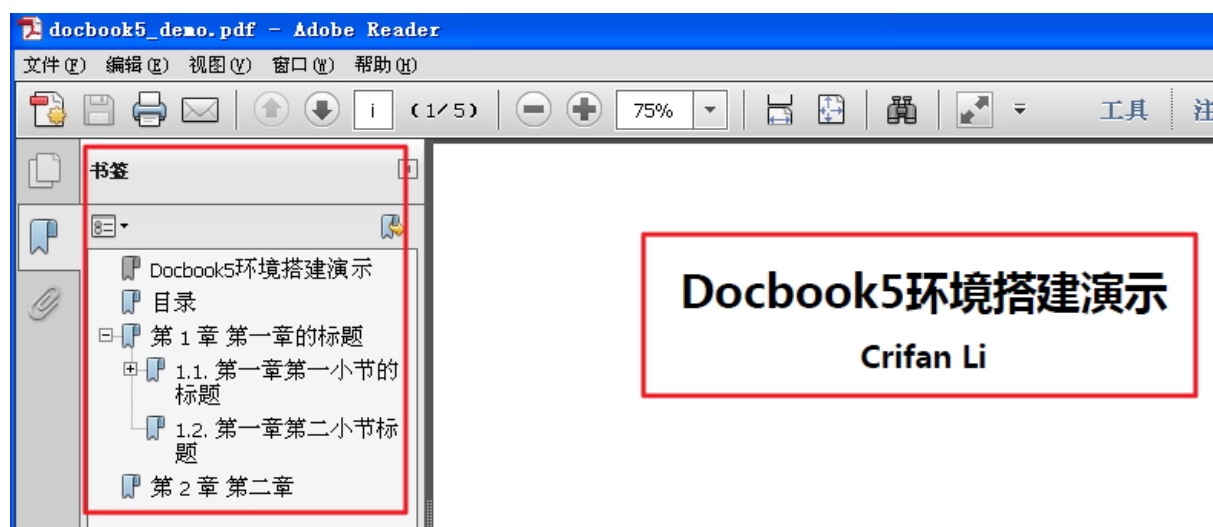
再继续执行下面命令：

```
E:\DevRoot\docbook\tools\fop-1.0\fop.cmd -c E:\DevRoot\docbook\config\fop\fop.xconf -fo ../output/fo/docbook5_demo.fo -pdf ../output/pdf/docbook5_demo.pdf
```

相应的屏幕输出为：

```
E:\DevRoot\docbook\books\docbook5_demo\src>E:\DevRoot\docbook\tools
\fop-1.0\fop.cmd -c E:\DevRoot\docbook\config\fop\fop.xconf -fo ../output/fo/docbook5_d
-pdf ../output/pdf/docbook5_demo.pdf
六月 07, 2012 11:28:38 下午 org.apache.fop.apps.FopFactoryConfigurator configure
信息: Default page-height set to: 11in
六月 07, 2012 11:28:38 下午 org.apache.fop.apps.FopFactoryConfigurator configure
信息: Default page-width set to: 8.26in
六月 07, 2012 11:28:41 下午 org.apache.fop.hyphenation.Hyphenator getHyphenationTree
严重: Couldn't find hyphenation pattern zh_cn
```

如此，就可以正常生成对应的pdf文件了，截图效果如下：



目录

1. 第一章的标题	1
1.1. 第一章第一小节的标题	1
1.1.1. 第一章第二小节的标题	1
1.2. 第一章第二小节标题	1
2. 第二章	2

第 1 章 第一章的标题

关于更多的Docbook方面的内容，可以去参考：Docbook开发手记 [http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html]

1.1. 第一章第一小节的标题

可以在这里输入第一章第一小节中的内容

1.1.1. 第一章第二小节的标题

1.2. 第一章第二小节标题

1.3. Windows环境下的，基于Cygwin的 Docbook开发环境的搭建

其实如果只是普通的用docbook去写技术文档，那么上述的纯windows的docbook开发环境，即：

`xsltproc + fop + docbook-xsl-ns`

已经够用了，即对于普通的docbook开发者的话，看到此处，已经足够了。

但是如果你也是像我一样，需要用到写更多的技术文档，需要用makefile等去管理各个book的编译，去传递定制的参数给xsltproc

以及利用Linux下面的一些常用工具，去实现特定的需求的话，那么就要用到cygwin了。

1.3.1. 安装Cygwin

关于什么是Cygwin和如何安装Cygwin，请参考：[Cygwin](#)²⁴

需要注意的是，安装时候，记得要选择以下几个库/模块：

1. xsltproc
用于将xml转换为HTML和FO等格式
2. xsltproc所依赖的一些库
xsltproc运行的话，需要依赖一些库，所以也要选上这些库。

同时，后期的一些其他处理，比如编码转换等，也需要相关的像iconv等工具，所以要安装上这些工具，方便后期使用。

- libxml2

²⁴ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/html/cygwin_intro.html

- zlib
- libxslt
- iconv

3. [docbook-xsl]

如前面[第 1.2.3 节 “下载docbook-xsl-ns-1.77.1”](#) 中的解释，为了生成各种格式的文件，是需要对应的xsl配置文件的。

其实这部分配置文件，新版本的cygwin中，也有了，所以可以选择安装，也可以不安装：

- 安装的话，就是选择对应的模块docbook-xsl

安装的理由是，cygwin自带的xsl，其实版本已经够新了，也基本足够用了。而且总类也很多，包括docbook-xsl，docbook-xsl-ns等等。

- 不安装的理由是，可以自己去按照前面的[第 1.2.3 节 “下载docbook-xsl-ns-1.77.1”](#) 一样去手动自己下载最新版本的，自己所需要的docbook-xsl。

因为最新的版本，往往功能上更加全。

所以，是否安装此docbook-xsl的话，可以根据自己的需要决定。

个人更加倾向于后者，自己手动下载最新版本的。

安装完毕cygwin后，可以查看对应的这些工具的版本等信息：

```
CLi@PC-CLI-1 ~/develop/docbook/books/soft_dev_basic/src
$ which xsltproc.exe
/usr/bin/xsltproc.exe

CLi@PC-CLI-1 ~/develop/docbook/books/soft_dev_basic/src
$ xsltproc --version
Using libxml 20708, libxslt 10126 and libexslt 815
xsltproc was compiled against libxml 20708, libxslt 10126 and libexslt 815
libxslt 10126 was compiled against libxml 20708
libexslt 815 was compiled against libxml 20708
```

1.3.2. 下载cygwin下使用fop-1.0

按理来说，cygwin下，直接是可以正常使用1.0版本的fop的，但是后来遇到那个很多人都遇到的NoClassDefFoundError的问题。

所以，要在cygwin下使用fop-1.0之前，去解决掉此问题：

将fop中（大概是189行）的：

```
LCP_TEMP=`cygpath --path --unix "$LOCALCLASSPATH"`
```

改为：

```
#LCP_TEMP=`cygpath --path --unix "$LOCALCLASSPATH"`
# for cygpath has bug, so use follow workaround
# edit by admin AT crifan DOT com
LCP_TEMP=`cygpath --path --$format "$LOCALCLASSPATH"`
```

```
LCP_TEMP=`cygpath --path --unix "$LCP_TEMP"`
```

详情参看：[【终极解决】fop错误：Exception in thread "main" java.lang.NoClassDefFoundError:org/apache/xmlgraphics/image/loader/ImageContext 的终极解决办法，即cygpath有bug，转换路径出错，导致部分路径被截断](http://www.crifan.com/fop_java_lang_noclassdeffounderror_xmlgraphics_cygpath_bug_path_truncated/)²⁵

²⁵ http://www.crifan.com/fop_java_lang_noclassdeffounderror_xmlgraphics_cygpath_bug_path_truncated/

第 2 章 Docbook开发过程中的注意事项和心得

2.1. Docbook 4和Docbook 5的区别

其实对于普通的Docbook的使用者来说，其实不必关心如何从Docbook4转到Docbook5，但是对于像我之前借鉴别人的docbook的xml去写docbook的源码的话，就需要关注一下了。

因为之前的很多Docbook的元素（Element），其实都是Docbook5之前的，所以现在是在用Docbook5的框架，但是却使用了部分的之前的元素。因此，需要列出一些需要注意的事项。

此处列出几个在使用时候需要注意事情：

1. 记得要给book，article等添加命名空间：

```
<?xml version="1.0"?>
<book xmlns="http://docbook.org/ns/docbook" version="5.0">
...
```

2. ulink被link取代了
关于交叉引用，link等方面的改变，去看[Improved cross-referencing and linking](#)¹
3. bookinfo，articleinfo等都被info取代了
关于更多的改变了名字的元素，去看[Renamed elements](#)²
关于更多在docbook5中去除了的元素，去看[Removed elements](#)³
4. 支持了用alt和annotation给元素添加注释
5. 新加了XSLT 2.0的支持

更多详细的内容，请参考：[DocBook 5 differences](#)⁴

2.2. 关于实体定义

如果想要输入一些特殊字符的话，那么可以考虑使用实体（Entity）定义。

比如：

- 用⇒来表示双等号右箭头：⇒
- 用™来表示trademark商标：™
- 用↔来表示单横线双向箭头：↔

而对这类特殊字符，之前是通关使用自己单独定义的crl.ent文件而实现的。其包含如下内容：

¹ <http://docbook.org/docs/howto/#changes-linking>

² <http://docbook.org/docs/howto/#changes-renamed>

³ <http://docbook.org/docs/howto/#changes-removed>

⁴ <http://www.sagehill.net/docbookxsl/Db5Tools.html#Db5Diffs>

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- more can refer: http://barzilai.org/math_sym.htm -->
<!ENTITY larr "&#x2190;"> <!-- ← -->
<!ENTITY rarr "&#x2192;"> <!-- → -->
<!ENTITY darr "&#x2193;"> <!-- ↓ -->
<!ENTITY uarr "&#x2191;"> <!-- ↑ -->
<!ENTITY harr "&#x2194;"> <!-- ↔ -->
<!ENTITY crar "&#x21B5;"> <!-- ∩ -->
<!ENTITY lArr "&#x21D0;"> <!-- ≤ -->
<!ENTITY uArr "&#x21D1;"> <!-- ∥ -->
<!ENTITY rArr "&#x21D2;"> <!-- ≥ -->
<!ENTITY dArr "&#x21D3;"> <!-- ∥ -->
<!ENTITY hArr "&#x21D4;"> <!-- ≡ -->
```

但是实际上，对于这些字符，官网已经有了标准的定义了。

然后在参考了<http://docbook.org/docs/howto/#faq-authoring-general-entities>，得知如下地址中有全部的定义：

<http://www.w3.org/2003/entities/2007/w3centities-f.ent>

对应的，<http://www.w3.org/2003/entities/2007/>中，也有其他相关的ent文件。

其中，也可以看到那个很常见的，包含了全部定义的文件：<http://www.w3.org/2003/entities/2007/w3centities.ent>

更多关于实体定义的解释，可以去：[XML Entity Declarations for Characters](#)⁵

而知道了有这么好的资源，就可以直接利用了，即，将上面的那个文件，改成：

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- refer: http://docbook.org/docs/howto/#faq-authoring-general-entities -->
<!ENTITY % allent SYSTEM "http://www.w3.org/2003/entities/2007/w3centities-f.ent">
%allent;
```

即可包含了全部的各种特殊字符的定义了，这样也就可以随处引用这些特殊字符了。

当然对于上述文件，每次都是需要联网下载w3centities-f.ent的，所以，为了不依赖网络和加快速度，可以下载w3centities-f.ent到本地，放在和crl.ent同文件夹下，然后crl.ent就可以再改为这样既可：

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % allent SYSTEM "w3centities-f.ent">
%allent;
```

⁵ <http://www.w3.org/2003/entities/>

2.3. qanda系列的用法

QA相关的关键字包括：qandaset，qandadiv，qandaentry，question，answer

其相关的用法举例，自己去参考官网的解释：[qandaset](http://www.docbook.org/tdg/en/html/qandaset.html)⁶

此处需要特别提示的是，对于qandadiv和qandaentry等，是不能随便放在某个section,sect1,para等之下使用的

我就是没注意，结果折腾了半天，始终出现错误：

```
org.apache.fop.apps.FOException: org.apache.fop.fo.ValidationException: "{http://
www.w3.org/1999/XSL/Format}block" is not a valid child of "fo:root"! (See position 29:6604)
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException: "{http://
www.w3.org/1999/XSL/Format}block" is not a valid child of "fo:root"! (See position 29:6604)
    at org.apache.fop.cli.InputHandler.transformTo(InputHandler.java:302)
    at org.apache.fop.cli.InputHandler.renderTo(InputHandler.java:130)
    at org.apache.fop.cli.Main.startFOP(Main.java:174)
    at org.apache.fop.cli.Main.main(Main.java:205)
Caused by: javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException:
"{http://www.w3.org/1999/XSL/Format}block" is not a valid child of "fo:root"! (See position
29:6604)
    at org.apache.xalan.transformer.TransformerIdentityImpl.transform(
TransformerIdentityImpl.java:501)
    at org.apache.fop.cli.InputHandler.transformTo(InputHandler.java:299)
    ... 3 more
```

结果始终不知道错在哪里，调试到最后，终于发现了，原来是qandaset等关键字，只能在chapter,article等之下使用，不能随便放在某个section,sect1,para等之下使用

所以，不要像我一样，想要在别处使用，导致始终出错，浪费了很长时间。。。

2.4. FOP相关注意事项

2.4.1. FOP的字体设置

关于FOP的字体的设置，对于最新版本的1.0的FOP来说，其设置很简单，参考官网的解释：[Apache™ FOP: Fonts - Custom Fonts](http://xmlgraphics.apache.org/fop/1.0/fonts.html#custom)⁷，只需要在配置文件fop.xconf中的renderers→renderer→fonts中，添加如下内容即可。

```
<directory recursive="true">file:///c:/windows/fonts/</directory>
<auto-detect/>
```

而与此相比，之前旧版本的FOP对于字体的设置，则相对比较繁琐，因为除了要用对应的命令去生成字体文件所对应的Metric文件之外，还要把相关配置添加到上述fop.xconf中去。

⁶ <http://www.docbook.org/tdg/en/html/qandaset.html>

⁷ <http://xmlgraphics.apache.org/fop/1.0/fonts.html#custom>

打开windows下面的cmd，然后切换到对应的fop的根目录，执行下面的命令：

```
java -cp build\fop.jar;lib\avalon-framework-4.2.0.jar;lib\commons-logging-1.0.4.jar;lib\commons-io-1.3.1.jar;lib\xmlgraphics-commons-1.4.jar org.apache.fop.fonts.apps.TTFReader -ttcname SimHei C:\Windows\Fonts\simhei.ttf fonts\simhei.xml

java -cp build\fop.jar;lib\avalon-framework-4.2.0.jar;lib\commons-logging-1.0.4.jar;lib\commons-io-1.3.1.jar;lib\xmlgraphics-commons-1.4.jar org.apache.fop.fonts.apps.TTFReader -ttcname msyh C:\Windows\Fonts\msyh.ttf fonts\msyh.xml

java -cp build\fop.jar;lib\avalon-framework-4.2.0.jar;lib\commons-logging-1.0.4.jar;lib\commons-io-1.3.1.jar;lib\xmlgraphics-commons-1.4.jar org.apache.fop.fonts.apps.TTFReader -ttcname msyhbd C:\Windows\Fonts\msyhbd.ttf fonts\msyhbd.xml
```

就可以生成相应的字体的XML Metrics文件了，然后再把相应的如下的设置：

```
<font metrics-url="E:\Dev_Root\docbook\dev\config\fop\fonts\simhei.xml"
kerning="yes" embed-url="C:\Windows\Fonts\simhei.ttf">
  <font-triplet name="SimHei" style="normal" weight="normal"/>
  <font-triplet name="SimHei" style="normal" weight="bold"/>
  <font-triplet name="SimHei" style="italic" weight="normal"/>
  <font-triplet name="SimHei" style="italic" weight="bold"/>
</font>

<font metrics-url="E:\Dev_Root\docbook\dev\config\fop\fonts\msyh.xml" kerning="yes"
embed-url="C:\Windows\Fonts\msyh.ttf">
  <font-triplet name="msyh" style="normal" weight="normal"/>
  <font-triplet name="msyh" style="normal" weight="bold"/>
  <font-triplet name="msyh" style="italic" weight="normal"/>
  <font-triplet name="msyh" style="italic" weight="bold"/>
</font>

<font metrics-url="E:\Dev_Root\docbook\dev\config\fop\fonts\msyhbd.xml"
kerning="yes" embed-url="C:\Windows\Fonts\msyhbd.ttf">
  <font-triplet name="msyhbd" style="normal" weight="normal"/>
  <font-triplet name="msyhbd" style="normal" weight="bold"/>
  <font-triplet name="msyhbd" style="italic" weight="normal"/>
  <font-triplet name="msyhbd" style="italic" weight="bold"/>
</font>

<directory recursive="true">file:///c:/windows/fonts/</directory>
<auto-detect/>
```

添加到fop.xconf中去。

2.5. Docbook开发过程中的一些有用的提示

2.5.1. Docbook最佳实践

此处总结一些，在写docbook源码的时候，一些常见的做法。

这些做法，多数都是我自己总结出来的，仅作参考。

2.5.1.1. Docbook中id的命名规则

在给docbook的元素添加id（或xml:id）属性时，名字的命名，我的做法是：

表 2.1. Docbook中id命名规则

Docbook元素	id命名规则	举例
figure	fg.xxx	<figure xml:id="fg.npp_code_collapse"> <title>XML文件中的Notepad++的代码折腾功能</title>
table	tbl.xxx	<table xml:id="tbl.docbook_id_abbrev"> <title>Docbook中id命名规则</title>
example	eg.xxx	<example xml:id="eg.npp_show_special_char"> <title>Notepad++可以查看特殊字符的用途举例</title>
某section	xxx	<sect1 xml:id="docbook_dev_notes"> <title>Docbook开发过程中的一些有用的提示</title>
co和 callout	co.xxx和 co.note.xxx	<pre><programlistingco> <programlisting> ... &lt;revhistory<co id="co.rev" linkends="co.note.rev" /> text-align="left" ... </programlisting> <calloutlist> <callout id="co.note.rev" arearefs="co.rev"> ... </callout> ... </calloutlist> </programlistingco></pre>

1. 最新的stylesheet：xsl-ns
想要去下载最新的xsl-ns，可以去这里下载：[DocBook Project Snapshots](http://docbook.sourceforge.net/snapshots/)⁸，其中有最新的xsl-ns下载。
2. 参数
docbook-xsl-ns-1.76.1\fo下面，有个param.xsl，该文件是对应的param.xml生成出来的。

该文件中包括了，所有的参数的默认的设置。而对应的每一个参数，都还有一个对应的配置文件，都放在了docbook-xsl-ns-1.76.1\params路径下了。

之所以说这个，是因为，关于各种参数的配置，最开始是没有头绪，找不到哪些内容都有哪些参数去控制输出的效果。现在找到了这些，就可以在遇到问题的时候，来找找这些参数了。
3. NS=namespace
关于xsl配置文件，之前叫做docbook-xsl-1.76.1，后来又有个docbook-xsl-ns-1.76.1

其中的ns，意思是namespace的意思，即支持命名空间版本的xsl

⁸ <http://docbook.sourceforge.net/snapshots/>

4. formal的含义
参考[Graphics](#)⁹而知道了之前就想要知道的formal的含义：formal是包含了figure, table, example和equation的
5. 只包含部分内容
根据[Using XIncludes with DocBook 5](#)¹⁰中的描述，如果想要只包含（include）某个xml文件的其中一部分内容，比如某个章节，也是可以通过include相应的xml:id去实现的此需求的。
详情参考：[Selecting part of a file](#)¹¹
6. 关于xml文件中，为何不直接写id而要写xml:id
之前一直没搞懂，为何很多地方，在解释在xml文件中的添加某个id的时候，写成xml:id，而为何不直接写id，毕竟很多地方，也都是这么写的，为何还非要麻烦地写成xml:id呢？

后来在学习了[xml的命名空间namespace](#)¹²之后，才知道，原来为由于docbook5中，会通过：

```
<book version="5.0"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"
>
```

的写法，设置默认的命名空间（default namespace）为http://docbook.org/ns/docbook，通过也添加了xi和xl的命令空间。

因此，在xml文件中，如果直接写id的话，其实意味着是属于docbook的属性的id，此处由于id没有和其他关键字冲突，所以，xml:id和id都是可以的，只是前者更加能保证不会发生关键字冲突。

相应的，如果是别的某个关键字，而多个域名中都可能会出现，则必须要添加对应的命名空间的前缀了。

比如我之前遇到的，将ulink改为link的时候，在写<link xl:href="xxx">的时候，如果写成<link href="xxx">，即没有加xl的命名空间的话，则生成出来的HTML和PDF中，都是没有链接的效果的，必须写成xl:href才可以的。

所以，更好的习惯，还是写成xml:id之类的，添加了命名空间的形式，这样更加准确，也可以避免未来可能的和其他命名空间发生关键字冲突的问题。

2.6. Docbook开发过程中如何调试错误

在Docbook开发过程中，难免会出现各种各样的错误。

遇到常见的错误的话，可以去参考我所总结的：[第3章 Docbook开发过程中的常见问题及解答](#)

但是有时候会遇到一些相对比较复杂的问题。

2.6.1. 举例说明如何实现方便地调试Docbook生成pdf过程中所出现的错误

比如，我所遇到的，一个docbook的xml源文件，本身就有500多行，内容很多。

⁹ <http://www.sagehill.net/docbookxsl/Graphics.html>

¹⁰ <http://www.sagehill.net/docbookxsl/ProcessingDb5.html>

¹¹ <http://www.sagehill.net/docbookxsl/ModularDoc.html#UsingXinclude>

¹² http://www.w3school.com.cn/xml/xml_namespaces.asp

其中可以看出，已经明确指出了出错的具体位置：27:127554

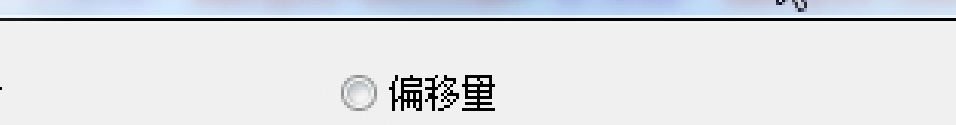
即，错误行号linenumber是27，行内偏移量offset为127554

调试无缩进的（xml格式的）fo文件。但是，当你去用Notepad++打开fo文件的话，所看到的内容是这样的：

A screenshot of a Notepad++ window titled "D:\tmp\lmp_dev_root\cgwin\home\CL\develop\docbook\books\char_encoding\output\fo\char_encoding-fo - Notepad++. The menu bar includes File(F), Edit(E), Search(S), View(V), Format(M), Language(L), Settings(I), Go(O), Run(R), Plugins(P), Window(W), and Help(H). The toolbar contains icons for file operations like Open, Save, Print, Copy, Paste, Undo, Redo, Find, and various view toggles. Below the toolbar, several tabs are open: char_encoding.xml, Makefile, ch02_enc_type.xml, reference.xml, appendix.xml, new_9, ch03_faq.xml, w3centities.ent, and new_10. The main editing area shows XML code starting with a declaration: <?xml version="1.0" encoding="UTF-8"?>. Line numbers 1 and 2 are visible on the left margin. The code defines a root element with various attributes and nested fo:region elements for page layout, including margins, padding, and orientation settings.

然后去通过选择：**语言** → **XML**，进行语法高亮后，变成：

```
<xsl:output method="xml"
  encoding="UTF-8"
  indent="no"/>
```



定位行号

☒ 行 ☐ 偏移量

当前位置： 1

目标位置： 27

截止位置： 28

定位(G)

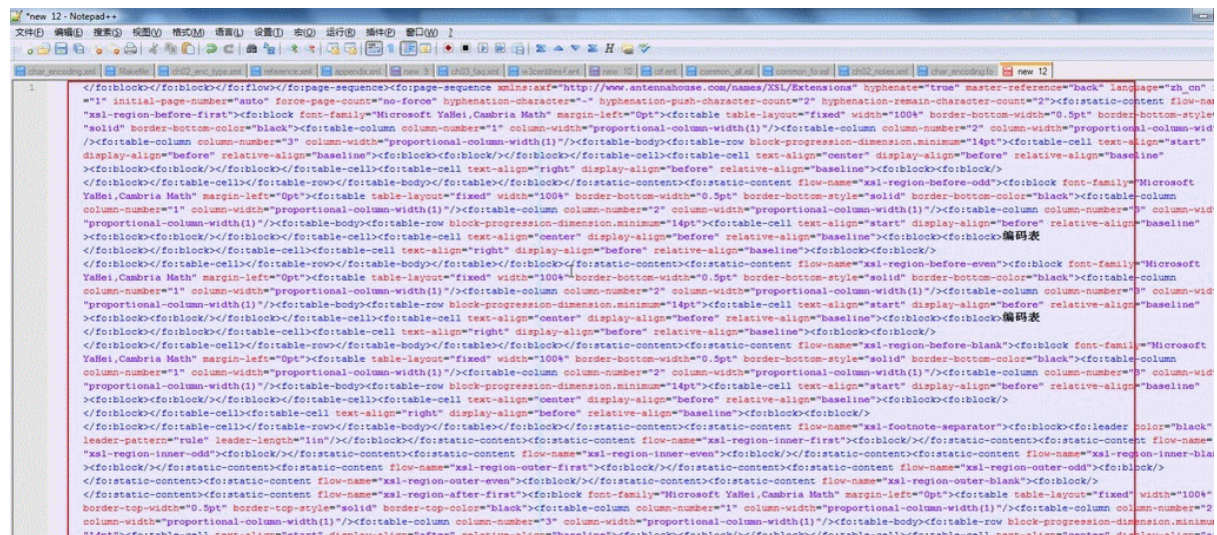
取消

Docbook开发过程中的注意事项和心得

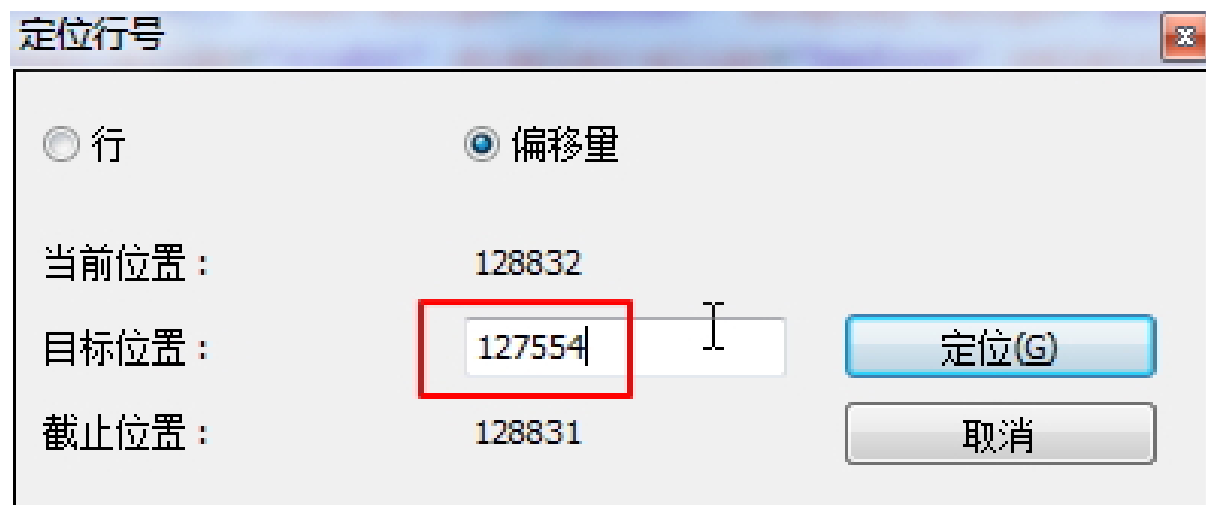
然后点击"定位"就可以跳转到27行了。



可以看出，仅仅是单一的27行，其内容已经太多了，多到鼠标也要拖动很多个屏幕，才能完全选中此行的内容。选中后，然后将此行的拷贝出来，放到新建的文件中去：



然后再去定位到行尾偏移量127554的位置：



找到出错的位置：

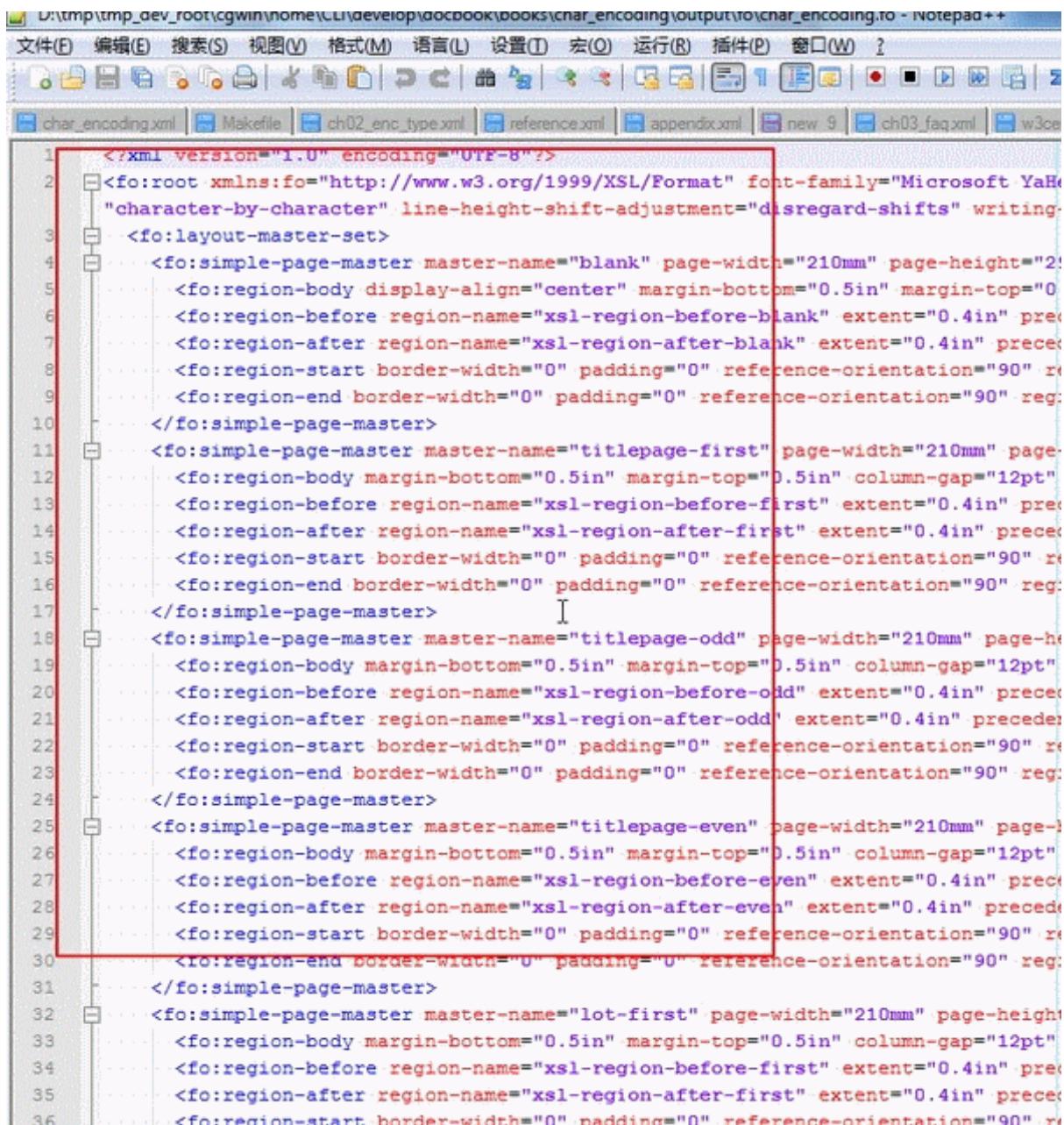
```
space-after.minimum="0.5em" space-after.optimum="1em" space-after.maximum="2em"><fo:block font-weight="bold" font-size="12pt"
'0.6em" space-after.maximum="0.8em" text-align="center" space-before.optimum="1em" space-before.minimum="0.8em"
>表 A.2. ISO/IEC 8859编码标准中的15种字符集</fo:block><fo:table border-before-width.conditionality="retain" border-collapse=
der-top-style="solid" border-bottom-style="solid" border-start-width="1.5pt" border-end-width="1.5pt" border-top-width="1.5pt"
color="black" border-top-color="black" border-bottom-color="black" table-layout="fixed" width="100%"><fo:table-column
:table-column column-number="2" column-width="proportional-column-width(1)"/><fo:table-column column-number="3" column-width=
www.renderx.com/XSL/Extensions" start-indent="0pt" end-indent="0pt"><fo:table-row keep-with-next.within-column="always"
'2pt" padding-bottom="2pt" background-color="antiquewhite" border-end-width="0.5pt" border-end-style="solid" border-end-color=
:table-cell><fo:table-cell padding-start="2pt" padding-end="2pt" padding-top="2pt" padding-bottom="2pt" background-color=
```

可以看出，此处即使找到了所指示的出错位置，也很难找到具体错在哪里。因为代码实在太多，很难找到哪里错误的。

调试有缩进的（xml格式的）fo文件. 对于上述所面临的很难定位错误的问题，解决办法是，配置xml格式的fo的输出中，是带代码缩进的：

```
<xsl:output method="xml"
  encoding="UTF-8"
  indent="yes"/>
```

这样，生成的xml格式的fo文件，就变成这样了：



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Microsoft YaHei"
  "character-by-character" line-height-shift-adjustment="disregard-shifts" writing-mode="vertical-rl">
3   <fo:layout-master-set>
4     <fo:simple-page-master master-name="blank" page-width="210mm" page-height="297mm">
5       <fo:region-body display-align="center" margin-bottom="0.5in" margin-top="0.5in" column-gap="12pt">
6       <fo:region-before region-name="xsl-region-before-blank" extent="0.4in" preceeds="body">
7       <fo:region-after region-name="xsl-region-after-blank" extent="0.4in" preceeds="body">
8       <fo:region-start border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-start">
9       <fo:region-end border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-end">
10    </fo:simple-page-master>
11    <fo:simple-page-master master-name="titlepage-first" page-width="210mm" page-height="297mm">
12      <fo:region-body margin-bottom="0.5in" margin-top="0.5in" column-gap="12pt">
13      <fo:region-before region-name="xsl-region-before-first" extent="0.4in" preceeds="body">
14      <fo:region-after region-name="xsl-region-after-first" extent="0.4in" preceeds="body">
15      <fo:region-start border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-start">
16      <fo:region-end border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-end">
17    </fo:simple-page-master>
18    <fo:simple-page-master master-name="titlepage-odd" page-width="210mm" page-height="297mm">
19      <fo:region-body margin-bottom="0.5in" margin-top="0.5in" column-gap="12pt">
20      <fo:region-before region-name="xsl-region-before-odd" extent="0.4in" preceeds="body">
21      <fo:region-after region-name="xsl-region-after-odd" extent="0.4in" preceeds="body">
22      <fo:region-start border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-start">
23      <fo:region-end border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-end">
24    </fo:simple-page-master>
25    <fo:simple-page-master master-name="titlepage-even" page-width="210mm" page-height="297mm">
26      <fo:region-body margin-bottom="0.5in" margin-top="0.5in" column-gap="12pt">
27      <fo:region-before region-name="xsl-region-before-even" extent="0.4in" preceeds="body">
28      <fo:region-after region-name="xsl-region-after-even" extent="0.4in" preceeds="body">
29      <fo:region-start border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-start">
30      <fo:region-end border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-end">
31    </fo:simple-page-master>
32    <fo:simple-page-master master-name="lot-first" page-width="210mm" page-height="297mm">
33      <fo:region-body margin-bottom="0.5in" margin-top="0.5in" column-gap="12pt">
34      <fo:region-before region-name="xsl-region-before-first" extent="0.4in" preceeds="body">
35      <fo:region-after region-name="xsl-region-after-first" extent="0.4in" preceeds="body">
36      <fo:region-start border-width="0" padding="0" reference-orientation="90" region-name="xsl-region-start">
```

对应的出错的提示中，行号和行内位置，也都变了：

```
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException: "fo:table" is
missing child elements. Required content model: (marker*,table-column*,table-header?,table-
footer?,table-body+) (See position 7594:441)
at org.apache.fop.cli.InputHandler.transformTo(InputHandler.java:302)
at org.apache.fop.cli.InputHandler.renderTo(InputHandler.java:130)
at org.apache.fop.cli.Main.startFOP(Main.java:174)
at org.apache.fop.cli.Main.main(Main.java:205)
```

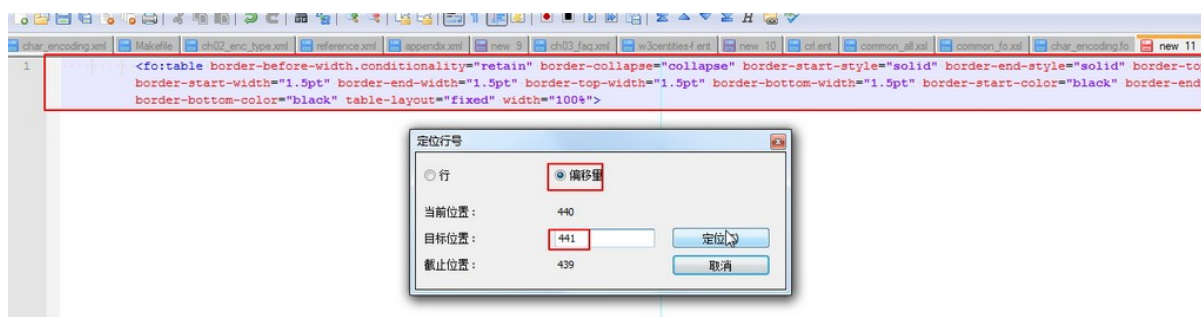
变成错误行号是7594，行内偏移量是441了。然后就去跳转到对应的行：



找到错误的位置：



然后类似的，把整行内容拷贝出来，然后再去跳转到行内位置441：



这样，在不是很多的一行内，就很容易找到对应错误位置，然后具体分析错误的原因了。

总结一下就是，对于fo文件的话，想要调试的话，记得设置输出的indent为yes：

```
<xsl:output method="xml"
  encoding="UTF-8"
  indent="no"/>
```

这样生成的fo中就是带缩进的了，格式就很清晰了，就容易找错误位置，方便调试错误了。

2.7. Docbook开发过程中的一些感悟

2.7.1. Docbook更关注内容

使用Docbook越多，越加发现，Docbook的确只是更加让你关注文章的内容如何写，而对于输出到底是什么样子的，就不需要让你关注了。

好处是可以让你把精力集中在文章的内容上，坏处是有时候，我可能会关注一下具体的输出效果，某些需要特别注明的字体的颜色高亮等设置，就没法方便的设置了。

但是总体还是利大于弊的。更多关注文章内容，而非文章的显示效果，可能更加有利于文章的最终的传播。

2.7.2. Docbook与盖楼

最近在折腾docbook，其内容总体来说包括两方面，一是搭建docbook的全套环境，而是完成每个book的xml源码。

突然想到，其实这个过程，就像是在盖楼。

搭建docbook的环境，就是建摩天大楼之前的打地基，这部分是最基础，也是最重要的。

基础没有建好，楼是盖不起来的。

建好了地基，再去一层层的盖楼，就是完成每本book的xml源码的过程。

而此处完成每本book的xml源码，和建造每层楼，又有点不同。

目前已经写好的20左右本book，每本的内容都完全不一样，虽然表现形式上，都算是一样的，符合docbook的定义而已。

而且每本book的内容，大小不同，换成楼层的层高的话，有的层高是只有3,5米，有的层高可能要达到30,50米。

总的说来，完成docbook环境和这么多book的内容，就像建好了地基，建了20多层的楼。

之后对docbook环境的微调，就像是在不同的细节方面，加固和完善地基。

而再写新的book，就相当于再建一层楼了。

而对于已写好的book的内容的增添修改，就相当于对于建好的楼层的内部装修，优化布局等细节处理了。

所以，之后要继续做的事情，就是继续加固地基，优化已有楼层，建新楼了。

另外，如果之后开始折腾替换已有xsltproc为Saxon的话，就相当于又找了个新的建筑方法，从新搭建另外一个地基了。

2.8. Docbook环境进化过程简单记录

此处简单记录一下，整个docbook环境，从无到有，到后期优化的大概进化过程：

1. 参考别人教程，可以编译HTML

2. 可以编译PDF了
3. 编写了一个统一的make，支持html，htmls，pdf，txt，chm，rtf
4. 添加了make release支持
5. 以最少改动去支持home和office
6. 把system的图片统一提取出去
7. 把css统一提取出去
8. 支持make两种版本local和release的时候对应链接地址不同
9. 支持从books目录下，对全局的books进行make操作
10. 添加了对于HTML中加入js代码实现统计页面访问次数的功能
[【已解决】给Docbook生成的静态HTML页面添加统计访问次数的代码](#)¹³
11. 升级到docbook-ns-1.78.1
[【记录】将docbook的xsl-ns从当前的docbook-xsl-ns-1.77.1升级到docbook-xsl-ns-1.78.1](#)¹⁴

¹³ http://www.crifan.com/docbook_add_code_to_statistic_static_html_page_view/

¹⁴ http://www.crifan.com/docbook_xsl_ns_upgrade_from_1_77_1_to_latest_1_78_1/

第 3 章 Docbook开发过程中的常见问题及解答

3.1. ValidationException: Property ID xxx previously used

```
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException:
Property ID "ref.ltr_syntax" (found on "fo:block") previously used; ID values must be
unique within a document! (See position 1315:169)
```

很常见的问题，即源码中，某个id的值，和之前的重复了。

常出现于，从别处拷贝了某段代码，然后修改完毕后，残留部分内容，导致其中有重复的id

解决办法：将重复的id删除或者改名

3.2. ValidationException: xxx is not a valid child of xxx

```
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException:
"{http://www.w3.org/1999/XSL/Format}list-item" is not a valid child of "fo:block"! (See
position 345:1605)
```

错误的根本原因是：非正确地使用了某个标签。

比如某个元素不是另一个元素的子元素，但却由于不熟悉，而误用了。

但是具体的表现形式，那可能是多种多样的。

下面就简单列举一下，我所遇到过的错误的例子：

1. 在callout中使用了qandaentry
结果导致此错误。然后去官网查了下：[qandaentry](http://www.w3.org/1999/XSL/Format#qandaentry)¹，才得知，qandaentry的父标签，只包含三种：answer, qandadiv, qandaset。所以，此处在此callout中使用qandaentry，肯定会出错了。

解决办法：不使用该标签，或者换一个合法的位置使用该标签。

2. 在part下，title和chapter之间，使用了para
结果导致报错：

```
org.apache.fop.apps.FOException: org.apache.fop.fo.ValidationException: "{http://
www.w3.org/1999/XSL/Format}block" is not a valid child of "fo:root"!
```

后来是删除了para，才去掉此错误的。

详情参考：[【已解决】org.apache.fop.apps.FOException: org.apache.fop.fo.ValidationException: "{http://www.w3.org/1999/XSL/Format}block" is not a valid child of "fo:root"!](#)²

3.3. ValidationException: xxx is missing child elements

```
org.apache.fop.apps.FOException: org.apache.fop.fo.ValidationException: "fo:list-item-
body" is missing child elements. Required content model: marker* (%block;)+ (See
position 77:16749)
```

¹ <http://www.w3.org/1999/XSL/Format#qandaentry>

² http://www.crfan.com/fop_error_validationexception_block_not_a_valid_child_of_fo_root/

```
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException:  
"fo:list-item-body" is missing child elements. Required content model: marker*  
(%block;)+ (See position 77:16747)
```

出现类似missing child elements的问题的话，从字面上就能看出，是说明某个关键字下面，缺少了某个必要的关键字

我遇到此问题，是因为就是折腾现在在用的这个qandaentry，结果在question下面，直接写上了内容，忘了加上para或screen等关键字，导致了始终此错误。

所以遇到此类问题的话，请自行去找找，是不是也对于某个关键字，漏了其下某个必须的关键字。

关于具体是哪个关键字出的问题，就要靠你自己调试，定位找到问题所在了。

而关于是其下缺少了哪个关键字，则是可以去[DocBook: The Definitive Guide](http://www.docbook.org/tdg/en/html/docbook.html)³去找到对应关键字，看其语法的定义了。

比如对于我这里的[question](http://www.docbook.org/tdg/en/html/question.html)⁴，其语法是

```
question ::=  
(label?,  
(calloutlist|glosslist|bibliolist|itemizedlist|orderedlist|  
segmentedlist|simplelist|variablelist|caution|important|note|  
tip|warning|literallayout|programlisting|programlistingco|  
screen|screenco|screenshot|synopsis|cmdsynopsis|funcsynopsis|  
classsynopsis|fieldsynopsis|constructorsynopsis|  
destructorsynopsis|methodsynopsis|formalpara|para|simpara|  
address|blockquote|graphic|graphicco|mediaobject|mediaobjectco|  
informalequation|informalexample|informalfigure|informaltable|  
equation|example|figure|table|procedure|anchor|bridgehead|  
remark|highlights|indexterm)+)
```

该加号 '+' 表示至少有1个（或更多），所以，此处就是缺少了任何一个，才出现错误的。加上合适的，比如我此处是代码，需要一个screen，即可解决问题了。

解决办法：找到对应出错的关键字，添加必要的子（child）关键字

3.4. docbook5中指定表格table中列column的宽度

解决办法：在table的tgroup之后，添加对应的colspec，指定对应的colwidth，即可达到自定义表格中列的宽度。示例代码如下：

```
<table> <title id="mpeg.header_format"> MPEG音频的帧头的格式</title>  
  <tgroup cols="4">  
    <colspec colnum="1" colname="col1" colwidth="1*"/>  
    <colspec colnum="2" colname="col2" colwidth="1*"/>  
    <colspec colnum="3" colname="col3" colwidth="6*"/>  
    <colspec colnum="4" colname="col4" colwidth="2*"/>  
  
  <thead>
```

³ <http://www.docbook.org/tdg/en/html/docbook.html>

⁴ <http://www.docbook.org/tdg/en/html/question.html>

```
<row><entry>位置(bit)</entry><entry>长度(bit)</entry><entry>含义</entry><entry>示例</entry></row>
</thead>

<tbody>
  <row><entry>0</entry><entry>11</entry><entry>用于同步帧，找到此帧头
  ( 所有位均置1 ) </entry><entry>1111 1111 111</entry></row>
  .....
</tbody>
</tgroup>
</table>
```

详情参考[【已解决】docbook5中指定表格table中列column的宽度](#)⁵

3.5. 除了章 (Chapter) 有编号之外，其他不同层级的小节 (section) 都是没有编号的，想要给各小节添加索引编号

解决办法：设置section.autolabel为1和section.label.includes.component.label为1，即可。

即给xsltproc添加参数：

```
--stringparam section.autolabel 1 --stringparam section.label.includes.component.label 1
```

或者通过给xsl文件中添加对应参数：

```
<xsl:param name="section.autolabel" select="1"/>
<xsl:param name="section.label.includes.component.label" select="1"/>
```

也可以达到同样效果。

详情参考[【已解决】Docbook中给章节添加索引编号](#)⁶

3.6. 已经include了xsl配置文件了，但是自己添加的参数配置，不起效果

原因：很可能是xsl的import和include没搞清楚，使用了include，导致后续设置的参数，没有起效

解决办法：将<xsl:include href="xxx.xsl"/>改改为<xsl:import href="docbook.xsl"/>即可。

详情参考[【整理】xsl的import和include，即xsl:import与xsl:include的区别和联系](#)⁷

3.7. xsltproc出错：Element xxx in namespace xxx encountered in book, but no template matches

此错误有两种类型：

1. Element include in namespace 'http://www.w3.org/2001/XInclude' encountered in book, but no template matches.

原因：多数情况下，主要原因都是，没有给xsltproc添加--xinclude参数

⁵ http://www.crifan.com/designate_the_width_of_column_in_docbook_table/

⁶ http://www.crifan.com/add_chapter_section_index_in_docbook/

⁷ http://www.crifan.com/difference_and_same_point_between_xsl_import_and_include/

另外，还要注意，不要出现，xml源码中使用了xi:include，但是却同时使用了ENTITY定义，以及不要把xi:include的位置放错了

解决办法：同时保证如下条件：

- a. 确保只是使用了xi:include，没有同时再使用ENTITY定义。
- b. 确保是xi:include是放在book之内，info之外
- c. 给xsltproc添加--xininclude参数

2. Element xxx in namespace encountered in book, but no template matches.

xxx为preface/chapter等docbook的元素。

原因：没有添加对应的默认的docbook的命名空间，和XInclude的命名空间

解决办法：将

```
<chapter>
```

改为：

```
<chapter
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"

  xml:id="mpeg_related"
>
```

即可。

解决办法：综合两种类型，给出可以使用的示例代码：

主文件，比如docbook_dev_note.xml中，如下：

```
<?xml version='1.0' encoding="utf-8"?>

<book version="5.0"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"
  xml:lang="zh_CN"
>

<info>
  <title>Docbook开发手记</title>
  <subtitle></subtitle>
```

```
<revhistory>...</revhistory>
<pubdate>2012-06-01</pubdate>
<copyright>
  <year>2012</year>
  <holder>Crifan</holder>
  <holder><link xl:href="http://crifan.com" /></holder>
</copyright>
<legalnotice>...</legalnotice>
</info>

<xi:include href="preface.xml" />
<xi:include href="ch01_build_env.xml" />
<xi:include href="appendix.xml" />

</book>
```

每个单独的分割的xml文件中，如下：

```
<?xml version='1.0' encoding="utf-8"?>

<chapter
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"

  xml:id="ch01_build_env"
  >

  <title>xxx</title>
  <sect1><title>xxx</title>
    <sect2><title>yyy</title>
      <para>xxx</para>
    </sect2>
  ...
  </sect1>

  <para>xxx</para>
</chapter>
```

然后在使用xsltproc处理xml的时候，记得加上参数--xinclude，即可。

详情参考【已解决】docbook中使用xsltproc出错：Element xxx in namespace xxx encountered in book, but no template matches⁸

3.8. pdf中没有出现/显示revhistory所对应的历史版本

原因：对应的stylesheet配置文件titlepage.templates.xml中没有添加revhistory，导致编出来的pdf中没有显示对应历史版本部分的内容。

解决办法：编辑titlepage.templates.xml，添加revhistory，变成：

⁸ http://www.crifan.com/docbook_element_in_namespace_encountered_in_book_but_no_template_matches/

```
<t:titlepage t:element="book" t:wrapper="fo:block">
  <t:titlepage-content t:side="recto">
...
    <author font-size="&hsize3;"
      space-before="&hsize2space;"
      keep-with-next.within-column="always"/>
    <!-- If you add editor, include this t:predicate attribute
      because only the first editor generates the list of editors.
    <editor t:predicate="[position() = 1]"/>
    -->
    <abstract text-align="left"
      space-before="&hsize5space;"/>
    <revhistory❶ text-align="left"
      space-before="&hsize5space;"/>❷
  </t:titlepage-content>
...
</t:titlepage>
```

- ❶ 添加对应的revhistory部分，可以根据自己的需要，放到合适的位置
- ❷ text-align, space-before等相关的设置，可以参考其他部分的设置，照葫芦画瓢，写出自己需要的配置。

然后将其另存为titlepage.templates_crl.xml，放在当前文件夹下。

再用如下命令去编译出相应的xsl文件：

```
xsltproc --output titlepage.templates_crls.xsl❶ D:/tmp/tmp_dev_root/cgwin/
home/CLi/develop/docbook/tools/docbook-xsl-ns-1.76.1/template/titlepage.xsl❷
titlepage.templates_crl.xml❸
```

- ❶ 输出xsl的文件名
- ❷ xsl-ns自带的titlepage的template
- ❸ 上述的，经过自己修改之后的xml模板文件

这样，就可以生成相应的titlepage.templates_crls.xsl了，然后去把此xsl文件添加到自己的配置中去：

```
<xsl:import href="titlepage.templates_crl.xsl"/>
```

用此配置编出来的pdf，就可以保护对应的revhistory部分的内容了。

详情参考[【已解决】docbook中pdf中没有出现/显示revhistory所对应的历史版本⁹](#)

关于这个问题，多说几句：

revhistory问题，同样也是releaseinfo所遇到的，默认pdf中都没有包含。

在mailing list中看到别人在2005年的时候，就讨论到此问题，默认pdf中没有releaseinfo：

[<releaseinfo> missing from fo book titlepage template?](#)¹⁰

⁹ http://www.crfan.com/docbook_revhistory_ont_show_display_in_pdf/

¹⁰ <http://lists.oasis-open.org/archives/docbook-apps/200510/msg00051.html>

而相关的开发人员Michael Smith也回复说，以后会添加此功能，结果我现在2012年了，用的fop 1.0 + docbook-xsl-ns-1.76.1，其中默认配置所输出的pdf中，也还是没把releaseinfo添加进来的。

而同样的问题，也出现在revhistory，默认输出的pdf中，是没有的。而默认输出的HTML中就有。

对此问题，即便是为了官网在解释如何定制化titlepage的教程中，把revhistory当做例子来解释，但是如此常用的功能，也应该在pdf输出中，像HTML输出一样，把其加入进来，否则对于多数使用者，在源码中写了revhistory和releaseinfo，结果输出的HTML中有，PDF中却没有，肯定会造成现在所已经出现的问题，就是大家都以为这个问题是个bug，所以google也能搜到一堆这类问题。

最新的结果是，新出的docbook-xsl-ns-1.77.0中，也还是没有把revhistory和releaseinfo包含进来，真是无语。。。

3.9. 生成pdf的时候出错：严重: Image not found. URI: xxx.png. (See position xxx)

原因：从问题本质上说，在图片存在的前提下，都是图片路径不正确。

但是对于为何图片路径不正确，至少有两种类型错误：

1. 本身图片路径设置有误
原因：对于图片本身所在的路径，没有正确指定对应的图片的地址，包括自己文章相关的图片的地址，和docbook系统相关的callout/admon/navig等相关的callout.graphics.path，admon.graphics.path，navig.graphics.path等路径。

解决办法：自行查找自己所设置的路径正确与否。

2. 运行环境不同，导致当前路径所相对应的图片路径有误，而找不到图片
之前由于某次改变了运行环境，换到了别的路径下，执行相应的命令去生成html和pdf，结果出现找不到图片的错误。

原因：换了运行脚本的路径，导致之前所设置的图片的相对路径，变得无效了。

解决办法：确保运行的环境所处的路径，是自己所期望的。

详情参考【已解决】docbook最后生成pdf的时候出错：严重: Image not found. URI: xxx.png. (See position xxx)¹¹

3.10. 想要给book中的标题下面添加一个版本号

没有找到专门的元素。暂时先借用releaseinfo来实现放此版本号等内容。

详情参考【已解决】给docbook中的book添加最新的版本号¹²

3.11. 输出的pdf中的revhistory历史版本的内容中的表格无边框，想要添加边框

解决办法：添加如下配置：

```
<!--<xsl:import href="titlepage_crl.xsl"/> -->

<xsl:attribute-set name="revhistory.title.properties">
  <xsl:attribute name="font-size">12pt</xsl:attribute>
  <xsl:attribute name="font-family">
    <xsl:value-of select="$title.fontset"/>
  </xsl:attribute>
</xsl:attribute-set>
```

¹¹ http://www.crifan.com/docbook_image_not_found/

¹² http://www.crifan.com/add_latest_version_number_for_docbook_book/


```
</xsl:attribute>
<xsl:attribute name="text-align">left</xsl:attribute>
</xsl:attribute-set>

<xsl:attribute-set name="revhistory.table.properties">
  <xsl:attribute name="border">0.5pt solid black</xsl:attribute>
  <xsl:attribute name="width">50%</xsl:attribute>
</xsl:attribute-set>

<xsl:attribute-set name="revhistory.table.cell.properties">
  <xsl:attribute name="border">0.5pt solid black</xsl:attribute>
  <xsl:attribute name="font-size">9pt</xsl:attribute>
  <xsl:attribute name="padding">4pt</xsl:attribute>
</xsl:attribute-set>
```

详情参考[【已解决】给docbook所输出的pdf中的revhistory历史版本的表格，添加边框](#)¹³

3.12. 如何去掉docbook中输出的pdf中，除了带链接的文字之外的那个链接地址

原因：PDF配置中，默认开启了ulink.show，所以会导致链接的地址，也同链接的文字同步显示出来。

解决办法：设置ulink.show为0即可。

给xsltproc传递参数

```
--stringparam ulink.show 0
```

或者添加如下配置：

```
<xsl:param name="ulink.show" select="0"/>
```

详情参考[【已解决】去掉docbook中输出的pdf中，除了带链接的文字之外的那个链接地址](#)¹⁴

3.13. 给docbook生成的pdf中的带链接的文字,添加下划线和设置字体颜色为蓝色

解决办法：添加如下配置：

```
<xsl:attribute-set name="xref.properties">
  <xsl:attribute name="color">blue</xsl:attribute>
  <xsl:attribute name="text-decoration">underline</xsl:attribute>
</xsl:attribute-set>
```

详情参考[【已解决】给docbook生成的pdf中的带链接的文字,添加下划线和设置字体颜色为蓝色](#)¹⁵

3.14. 如何给docbook的pdf添加bookmark书签的功能

¹³ http://www.crifan.com/add_table_board_line_for_revhistory_in_pdf_of_docbook/

¹⁴ http://www.crifan.com/make_http_link_address_not_show_in_pdf_of_docbook/

¹⁵ http://www.crifan.com/add_underline_underscore_for_link_in_docbook_pdf/

解决办法：

添加如下配置：

```
<xsl:param name="fop1.extensions">1</xsl:param>
```

详情参考【[顺带实现](#)】给docbook的pdf添加bookmark书签的功能¹⁶

更多的关于fop1.extensions的解释，可以去看：[Installing an XSL-FO processor](#)¹⁷和[fop1.extensions](#)¹⁸

3.15. Docbook的pdf，去除正文对于标题的缩进indent

解决办法：

添加如下配置：

```
<xsl:param name="body.start.indent">0pt</xsl:param>
```

详情参考【[已解决](#)】Docbook的pdf，去除正文对于标题的缩进indent¹⁹

3.16. 生成的pdf中emphasis无效果，想要实现给pdf中emphasis的文字加粗或斜体

原因：由于当前使用的是中文字体，中文字体没有对应的粗体和斜体，所以无效果。

与此相对的是，很多英文字体，是有对应的粗体或斜体的，所以会有效果。

而其他word等文字处理软件中，中文字体也有粗体和斜体的效果，只是相应软件处理生成出来的效果。

解决办法：此处想要给emphasis的文字的颜色设置为brown

主要就是修改inline.italicseq和inline.boldseq的部分，把相应的font-style="italic"和font-weight="bold"都改为color="brown"即可。

即添加如下配置：

```
<!--=====
=====
emphasis setting
=====
=====-->

<!-- copy from docbook-xsl-ns-1.77.0\fo\inline.xsl -->
<xsl:template name="inline.italicseq">
  <xsl:param name="content">
    <xsl:call-template name="simple.xlink">
      <xsl:with-param name="content">
```

¹⁶ http://www.crifan.com/remove_indent_of_content_and_add_bookmark_for_docbook_pdf/

¹⁷ <http://www.sagehill.net/docbookxsl/InstallingAnFO.html>

¹⁸ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/fo/fop1.extensions.html>

¹⁹ http://www.crifan.com/remove_indent_of_content_and_add_bookmark_for_docbook_pdf/

```
<xsl:apply-templates/>
</xsl:with-param>
</xsl:call-template>
</xsl:param>

<!-- changed by crifan start -->
<!-- <fo:inline font-style="italic"> -->
<fo:inline color="brown">
<!-- changed by crifan end -->
<xsl:call-template name="anchor"/>
<xsl:if test="@dir">
  <xsl:attribute name="direction">
    <xsl:choose>
      <xsl:when test="@dir = 'ltr' or @dir = 'lro'">ltr</xsl:when>
      <xsl:otherwise>rtl</xsl:otherwise>
    </xsl:choose>
  </xsl:attribute>
</xsl:if>
<xsl:copy-of select="$content"/>
</fo:inline>
</xsl:template>

<xsl:template name="inline.boldseq">
  <xsl:param name="content">
    <xsl:call-template name="simple.xlink">
      <xsl:with-param name="content">
        <xsl:apply-templates/>
      </xsl:with-param>
    </xsl:call-template>
  </xsl:param>

  <!-- changed by crifan start -->
  <!-- <fo:inline font-weight="bold"> -->
  <fo:inline color="brown">
  <!-- changed by crifan end -->
  <xsl:if test="@dir">
    <xsl:attribute name="direction">
      <xsl:choose>
        <xsl:when test="@dir = 'ltr' or @dir = 'lro'">ltr</xsl:when>
        <xsl:otherwise>rtl</xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
  </xsl:if>
  <xsl:copy-of select="$content"/>
</fo:inline>
</xsl:template>
```

详情参考[【已解决】docbook中生成的pdf的（中文字体的）emphasis无效果，没有加粗或斜体等效果](http://www.crifan.com/docbook_pdf_emphasis_no_effect_for_chinese_font/)²⁰

3.17. 如何给（docbook中的）xsl中定义实体Entity

解决办法：参考如下中的xsl_ns_base_cygwin如何定义的，即可：

²⁰ http://www.crifan.com/docbook_pdf_emphasis_no_effect_for_chinese_font/

```
<?xml version='1.0'?>
<!DOCTYPE stylesheet
[
<!ENTITY xsl_ns_base_cygwin "/home/develop/docbook/tools_root/docbook-xsl-
ns-1.77.0">
]
>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version='1.0'>
<xsl:import href="&xsl_ns_base_cygwin;/html/docbook.xsl"/>
<xsl:import href="../common_html.xsl"/>
</xsl:stylesheet>
```

详情参考[【已解决】如何给 \(docbook中的 \) xsl中定义实体Entity](#)²¹

3.18. 给docbook的pdf中的等式(equation)设置背景色

解决办法：

添加如下配置：

```
<xsl:attribute-set name="equation.properties" use-attribute-
sets="formal.object.properties">
  <xsl:attribute name="border-style">solid</xsl:attribute>
  <xsl:attribute name="border-width">0.5pt</xsl:attribute>
  <xsl:attribute name="background-color">silver</xsl:attribute>
</xsl:attribute-set>
```

详情参考[【已解决】给docbook的pdf中的等式\(equation\)设置背景色](#)²²

3.19. 如何将默认的输出单个的HTML页面，改为输入多个HTML页面？

解决办法：将之前引用html的docbook.xsl换为chunk.xsl即可。

详情参考[【已解决】把Docbook默认输出的单个HTML文件，改为输出多个的HTML文件](#)²³

3.20. 如何给docbook中的表格 (table) 的第一行 (firstrow或thead) 添加背景色？

解决办法：

添加如下配置：

```
<!--=====
=====
table setting
=====
----->

<!-- copy from docbook-xsl-ns-1.76.1\fo\table.xsl -->
```

²¹ http://www.crifan.com/define_entity_in_xsl_file_in_docbook/

²² http://www.crifan.com/docbook_pdf_set_equation_background_color/

²³ http://www.crifan.com/make_output_multiple_html_file_for_docbook/

```
<!-- Expand this template to add properties to any fo:table-cell -->
<xsl:template name="table.cell.properties">
  <xsl:param name="bgcolor.pi" select=""/>
  <xsl:param name="rowsep.inherit" select="1"/>
  <xsl:param name="colsep.inherit" select="1"/>
  <xsl:param name="col" select="1"/>
  <xsl:param name="valign.inherit" select=""/>
  <xsl:param name="align.inherit" select=""/>
  <xsl:param name="char.inherit" select=""/>

  <xsl:choose>
    <xsl:when test="ancestor::d:tgroup">

      <!-- added by crifan start -->
      <xsl:if test="ancestor::d:thead">
        <xsl:attribute name="background-color">antiquewhite</xsl:attribute>
      </xsl:if>
      <!-- added by crifan end -->

      <xsl:if test="$bgcolor.pi != ''">
        <xsl:attribute name="background-color">
          <xsl:value-of select="$bgcolor.pi"/>
        </xsl:attribute>
      </xsl:if>

      <xsl:if test="$rowsep.inherit > 0">
        <xsl:call-template name="border">
          <xsl:with-param name="side" select="'bottom'"/>
        </xsl:call-template>
      </xsl:if>

      <xsl:if test="$colsep.inherit > 0 and
        $col <= (ancestor::d:tgroup/@cols|ancestor::d:entrytbl/@cols)[last()]">
        <xsl:call-template name="border">
          <xsl:with-param name="side" select="'end'"/>
        </xsl:call-template>
      </xsl:if>

      <xsl:if test="$valign.inherit != ''">
        <xsl:attribute name="display-align">
          <xsl:choose>
            <xsl:when test="$valign.inherit='top'">before</xsl:when>
            <xsl:when test="$valign.inherit='middle'">center</xsl:when>
            <xsl:when test="$valign.inherit='bottom'">after</xsl:when>
            <xsl:otherwise>
              <xsl:message>
                <xsl:text>Unexpected valign value: </xsl:text>
                <xsl:value-of select="$valign.inherit"/>
                <xsl:text>, center used.</xsl:text>
              </xsl:message>
              <xsl:text>center</xsl:text>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:attribute>
      </xsl:if>

      <xsl:choose>
        <xsl:when test="$align.inherit = 'char' and $char.inherit != ''">
```

```

        <xsl:attribute name="text-align">
          <xsl:value-of select="$char.inherit"/>
        </xsl:attribute>
      </xsl:when>
      <xsl:when test="$align.inherit != ''">
        <xsl:attribute name="text-align">
          <xsl:value-of select="$align.inherit"/>
        </xsl:attribute>
      </xsl:when>
    </xsl:choose>

  </xsl:when>
  <xsl:otherwise>
    <!-- HTML table -->
    <xsl:if test="$bgcolor.pi != ''">
      <xsl:attribute name="background-color">
        <xsl:value-of select="$bgcolor.pi"/>
      </xsl:attribute>
    </xsl:if>

    <xsl:if test="$align.inherit != ''">
      <xsl:attribute name="text-align">
        <xsl:value-of select="$align.inherit"/>
      </xsl:attribute>
    </xsl:if>

    <xsl:if test="$valign.inherit != ''">
      <xsl:attribute name="display-align">
        <xsl:choose>
          <xsl:when test="$valign.inherit='top'">before</xsl:when>
          <xsl:when test="$valign.inherit='middle'">center</xsl:when>
          <xsl:when test="$valign.inherit='bottom'">after</xsl:when>
          <xsl:otherwise>
            <xsl:message>
              <xsl:text>Unexpected valign value: </xsl:text>
              <xsl:value-of select="$valign.inherit"/>
              <xsl:text>, center used.</xsl:text>
            </xsl:message>
            <xsl:text>center</xsl:text>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:attribute>
    </xsl:if>

    <xsl:call-template name="html.table.cell.rules"/>

  </xsl:otherwise>

</xsl:choose>

</xsl:template>

```

详情参考[【已解决】给docbook中的表格（table）的第一行（firstrow或thead）添加背景色](#)²⁴

²⁴ http://www.crifan.com/set_background_color_for_first_row_of_table_for_docbook/

3.21. 如何用docbook生成chm（微软帮助文件）？

解决办法：

总的逻辑是：使用htmlhelp.xml生成相关的HTML文件和一个总的配置文件htmlhelp.hhp，再用微软的工具hhc去将htmlhelp.hhp转换为对应的chm文件。

当然，使用htmlhelp.xml的时候，别忘了加上自己的配置，我的配置如下：

```
<?xml version='1.0'?>

<!DOCTYPE stylesheet
[
<!ENTITY xsl_ns_base_cygwin "/home/CLi/develop/docbook/tools/docbook-xsl-
ns-1.76.1">
]
>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version='1.0'>

<xsl:import href="&xsl_ns_base_cygwin;/htmlhelp/htmlhelp.xml"/>

<xsl:output
    encoding="UTF-8"
    indent="no"/>

<xsl:param name="title.font.family">msyhbd</xsl:param>
<xsl:param name="body.font.family">msyh</xsl:param>
<xsl:param name="monospace.font.family">msyh</xsl:param>

</xsl:stylesheet>
```

然后用：

```
xsltproc --xinclude -o MPEG_VBR.html docbook_htmlhelp_crl.xml MPEG_VBR.xml
```

生成对应的HTML文件和htmlhelp.hhp

然后再去下载[HTML Help Wordshop](#)²⁵并安装。

然后用

```
hhc htmlhelp.hhp
```

去生成对应的chm

详情参考[【全部解决】用Docbook生成htmlhelp](#)²⁶

3.22. docbook生成的chm的左边目录等内容都是乱码

原因：主要是由于微软的hhc对UTF-8编码支持不好，导致会出现乱码。

解决办法：确保本身输出的HTML都是UTF-8编码：

²⁵ <http://download.microsoft.com/download/0/a/9/0a939ef6-e31c-430f-a3df-dfae7960d564/htmlhelp.exe>

²⁶ http://www.crifan.com/use_docbook_output_htmlhelp_fix_messy_cn_char_of_chm_title_and_toc_index/

```
<xsl:param name="htmlhelp.encoding">UTF-8</xsl:param>
<xsl:param name="chunker.output.encoding">UTF-8</xsl:param>
```

然后再用iconv将UTF-8转换为GB18030（因为我的本地系统是中文的GBK，而GB18030是GBK的超集，使用GB18030比GBK更好，可以支持更多的字符）

```
iconv -f UTF-8 -t GB18030 < ../output/htmlhelp/htmlhelp.hhp > ../output/htmlhelp/
htmlhelp_gb18030.hhp
mv ../output/htmlhelp/htmlhelp_gb18030.hhp ../output/htmlhelp/htmlhelp.hhp

iconv -f UTF-8 -t GB18030 < ../output/htmlhelp/toc.hhc > ../output/htmlhelp/
toc_gb18030.hhc
mv ../output/htmlhelp/toc_gb18030.hhc ../output/htmlhelp/toc.hhc
```

最后再用hhc去处理，就可以了：

```
hhc htmlhelp.hhp
```

详情参考[【完全解决】生成的chm中标题和左边的索引目录是乱码的问题](#)²⁷

3.23. 如何将qanda的标题添加到TOC目录中去？

解决办法：设置qanda.in.toc为1即可：

```
<xsl:param name="qanda.in.toc">1</xsl:param>
```

更多的解释，请参考：[Q and A in table of contents](#)²⁸

3.24. 如何将docbook转换为（和word兼容的）RTF文档？

解决办法：先用xsltproc转换xml为fo，然后再用fop将fo转换为rtf：

```
fop -c fop.xconf MPEG_VBR.fo -rtf MPEG_VBR.rtf
```

详情参考[【记录】将docbook的xml源码，通过xsltproc和FOP生成（可用word打开的）RTF（Word兼容）格式](#)²⁹



目前已知的没有非常好的生成RTF的办法

目前折腾的几种方法，包括xsltproc+fop，openjade，XFC等，生成的rtf，效果都不是很好，所以貌似除非有空自己折腾一下，添加对应的xsl的配置，以使得生成的rtf效果更好些，目前暂无别的好的办法。

²⁷ http://www.crifan.com/use_docbook_output_htmlhelp_fix_messy_cn_char_of_chm_title_and_toc_index/

²⁸ <http://www.sagehill.net/docbookxsl/QandaInToc.html>

²⁹ http://www.crifan.com/docbook_output_rtf_word_compatible_format/

详情可以去看偶的折腾过程：[【已放弃解决】将docbook的xml，用xsltproc和fop生成rtf时出错：WARNING: Only simple-page-masters are supported on page-sequences. Using default simple-page-master from page-sequence-master "titlepage" "front" "body" "back",WARNING: RTF: fo:leader reference-area not supported + 【记录】折腾openjade和XFC去生成RTF](#)³⁰

3.25. 如何用Docbook生成Webhelp？

解决办法：使用webhelp.xml即可。不过偶目前暂时还没折腾出来。有空再去弄。

详情参考[【记录】将Docbook生成Webhelp](#)³¹

3.26. 用fop生成pdf过程中出现警告：warning: Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400"

解决办法：

如果只是想要去除此警告，那么可以加上此配置即可：

```
<xsl:param name="symbol.font.family"></xsl:param>
```

详情参考[【已解决】Docbook中用fop生成pdf过程中出现警告：warning: Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400"](#)³²



注意

实际上，此symbol.font.family对应着的是“符号”所采用哪种字体

而我们的目的，其实应该是支持最全的字符的显示，而不是关注，如何取消此无关紧要的警告。

而据说Cambria Math字体包含的字符算是最全的了，所以，建议设置为：

```
<xsl:param name="symbol.font.family"></xsl:param>
```

以此实现各种特殊字符，都能可以很好的显示出来。（因为之前用微软雅黑，好像个别特殊字符显示不出来，所以才会如此设置。）

3.27. Docbook中使用fop过程中出错：严重: Couldn't find hyphenation pattern zh_cn

解决办法：虽然有此提示，但是实际上中文的断字，基本上已经很好了。至少我这里这么多的book，都没有出现之前别人遇到的，中文的标点符号，偶尔会在句首出现的问题。

但是的确好像对于中文的hyphenation，没有官方的支持的。只有en等英语的支持。

总的来说，就一句话，暂时可以忽略此“严重”型的警告，生成的pdf中的中文的显示，都还是正常的。

³⁰ http://www.crifan.com/docbook_rtf_fop_warning_only_simple_page_masters_are_supported_on_page_sequences_rtf_fo_leader_reference_area_not_supported/

³¹ http://www.crifan.com/docbook_output_webhelp/

³² http://www.crifan.com/docbook_fop_pdf_warning_font_symbol_not_found_substituting_with/

详情参考：[【部分解决】Docbook中使用fop过程中出错：严重: Couldn' t find hyphenation pattern zh_cn](http://www.crifan.com/docbook_fop_could_not_find_hyphenation_pattern_zh_cn/)³³

3.28. cygwin/linux下的fop运行出错：Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/xmlgraphics/image/loader/ImageContext

原因：fop中所用到的cygpath有bug，导致不很好的支持windows中关于CLASS_PATH的定义，最终导致转换出来的路径出错，无法找到相应的各种的java的库

解决办法：修改fop文件，将（大概是189行的）：

```
LCP_TEMP=`cygpath --path --unix "$LOCALCLASSPATH"`
```

改为：

```
#LCP_TEMP=`cygpath --path --unix "$LOCALCLASSPATH"`  
# for cygpath has bug, so use follow workaround  
# edit by admin AT crifan DOT com  
LCP_TEMP=`cygpath --path --$format "$LOCALCLASSPATH"`  
LCP_TEMP=`cygpath --path --unix "$LCP_TEMP`
```

详情参考[【终极解决】fop错误：Exception in thread "main" java.lang.NoClassDefFoundError:org/apache/xmlgraphics/image/loader/ImageContext的终极解决办法，即cygpath有bug，转换路径出错，导致部分路径被截断](http://www.crifan.com/docbook_fop_error_exception_in_thread_main_java_lang_no_class_def_found_error_org_apache_xmlgraphics_image_loader_image_context_the_final_solution_that_cygpath_has_bug_path_conversion_error_causes_part_of_path_to_be_truncated/)³⁴

3.29. Docbook中用xsltproc生成HTML过程中出错：runtime error xxx/chunker.xml line 226 element document, invalid value for indent,xsltRunStylesheet : run failed

原因：有些参数的值，是字符串，所以如果没有用单引号括起来，则会导致参数赋值出错。

解决办法：将原先的参数赋值中的字符串类型的值，加上单引号。即，把：

```
<xsl:param name="chunker.output.indent" select="yes"/>
```

改为：

```
<xsl:param name="chunker.output.indent" select="'yes'"/>
```

详情参考[【已解决】Docbook中用xsltproc生成HTML过程中出错：runtime error xxx/chunker.xml line 226 element document, invalid value for indent,xsltRunStylesheet : run failed](http://www.crifan.com/docbook_xsltproc_generate_html_runtime_error_invalid_value_for_indent_xslt_run_stylesheet_run_failed/)³⁵

3.30. docbook的xsltproc生成的HTML中Charset的值是空的，不是所设置的UTF-8

³³ http://www.crifan.com/docbook_fop_could_not_find_hyphenation_pattern_zh_cn/

³⁴ http://www.crifan.com/fop_java_lang_noclassdeffoundererror_xmlgraphics_cygpath_bug_path_truncated/

³⁵ http://www.crifan.com/docbook_xsltproc_generate_html_runtime_error_invalid_value_for_indent_xslt_run_stylesheet_run_failed/

原因：同问：[3.29](#)

解决办法：把

```
<xsl:param name="chunker.output.encoding" select="UTF-8"/>
```

改为：

```
<xsl:param name="chunker.output.encoding" select="'UTF-8'"/>
```

详情参考[【顺带解决】docbook的xsltproc生成的HTML中Charset的值是空的，不是所设置的UTF-8](#)³⁶

3.31. 如何给Docbook中添加abbrev缩略词一项，且放置在目录之后，正文之前？

解决办法：单独写一个glossary.xml，其中内容如下：

```
<?xml version='1.0' encoding="utf-8"?>

<glossary xmlns="http://docbook.org/ns/docbook"
  xml:id="glossary"
  xmlns:xi="http://www.w3.org/2001/XInclude">
<title>缩略词</title>

<glossentry id="ASE"><glossterm>ASE</glossterm>
  <acronym>ASE</acronym>
  <glossdef>
    <para>Application Specific Extension</para>
    <para>（根据应用的）专用扩展</para>
  </glossdef>
</glossentry>

<glossentry id="xxx"><glossterm>xxx</glossterm>
  <acronym>xxx</acronym>
  <glossdef>
    <para>xxx</para>
    <para>xxx</para>
  </glossdef>
</glossentry>

</glossary>
```

然后将把glossary部分源码放到bookinfo之后，XInclude其他各个章节之前：

³⁶

http://www.crifan.com/docbook_xsltproc_generate_html_runtime_error_invalid_value_for_indent_xsltrunstylesheet_run_failed/

```
<?xml version='1.0' encoding="utf-8"?>

<book version="5.0" ... >

<bookinfo>
  <title>ARM与MIPS的详细对比</title>
  <subtitle></subtitle>
...
</bookinfo>
<xi:include href="glossary.xml"/>
<xi:include href="ch01_reduced_instruction_set.xml"/>
...
</book>
```

详情参考[【已解决】给Docbook中添加缩略词 + 自动链接到缩略词表项 + 将缩略词单独放置一页，且位于目录之后，正文之前](#)³⁷

3.32. 给Docbook的HTML的等式（equation），程序源码（programlisting）等添加背景色

解决办法：添加相应的css文件，其配置如下：

```
/*
 * about html color and corresponding name can refer:
 * http://www.w3schools.com/html/html_colornames.asp
 */

/* programlisting */
pre.programlisting {
/* background-color: #F4F4F4 ; */
  background-color: Lavender ;
  border: 1px solid #006600 ;
}

/* screen */
pre.screen {
/* background-color: #F4F4F4 ; */
  background-color: Lavender ;
  border: 1px solid #006600 ;
}

/* equation */
div.equation {
/* background-color: #F4F4F4 ; */
  background-color: Lavender ;
  border: 1px solid #006600 ;
}

/* table */
/* thead=table header */
thead {
```

³⁷ http://www.crifan.com/docbook_generate_acronym_abbrev_auto_link_to_glossary_item/

```
background-color: antiquewhite ;
}

/* QandA: Question and Answer */
tr.question
{
background-color: antiquewhite ;
}
```

详情参考【已解决】给Docbook的HTML的等式添加背景色³⁸

3.33. PDF中和HTML中的图片显示，始终不能统一，要么HTML中是放大的，要么PDF中不能完全显示，所以希望PDF和HTML中的图片显示可以用同一套代码来统一

解决办法：参考如下代码：

对于多数的，尺寸大于pdf中页面宽度（大概是500左右的像素）的图片，则使用pdf中设置为自动缩放：

```
<figure> <title>Nand Flash的结构图</title>
  <mediaobject>
    <imageobject role="html">
      <imagedata fileref="images/nand_flash_layout.png" align="center" scalefit="0"
width="100%" />
    </imageobject>
    <imageobject role="fo">
      <imagedata fileref="images/nand_flash_layout.png" align="center" scalefit="1"
width="100%" />
    </imageobject>
  </mediaobject>
</figure>
```

对于多数的，尺寸小于pdf中页面宽度的图片，则使用pdf中设置为禁止自动缩放，并且自行调节显示的比例：

```
<figure> <title>典型的Flash内存单元的物理结构</title>
  <mediaobject>
    <imageobject role="html">
      <imagedata fileref="images/flash_cell_structure.jpg" align="center" scalefit="0"
width="100%" />
    </imageobject>
    <imageobject role="fo">
      <imagedata fileref="images/flash_cell_structure.jpg" align="center" scalefit="0"
width="80%" />
    </imageobject>
  </mediaobject>
</figure>
```

³⁸ http://www.crifan.com/docbook_html_equation_background_color/

详情参考【[基本解决](#)】[Docbook生成的PDF和HTML中图片缩放不统一](#)³⁹

3.34. 想要设置HTML和PDF中的图片的标题都是居中对齐

解决办法：

1. FOP

对于FOP添加设置：

```
<!--=====
=====
formal(figure/table/equation/example/...) setting
=====
=====-->

<!-- copy from docbook-xsl-ns-1.76.1\fo\formal.xsl -->
<!-- http://www.sagehill.net/docbookxsl/TitleFontSizes.html#FormalTitleProperties
-->
<xsl:attribute-set name="formal.title.properties"
    use-attribute-sets="normal.para.spacing">
  <xsl:attribute name="font-weight">bold</xsl:attribute>
  <xsl:attribute name="font-size">12pt</xsl:attribute>
  <xsl:attribute name="hyphenate">>false</xsl:attribute>
  <xsl:attribute name="space-after.minimum">0.4em</xsl:attribute>
  <xsl:attribute name="space-after.optimum">0.6em</xsl:attribute>
  <xsl:attribute name="space-after.maximum">0.8em</xsl:attribute>

  <xsl:attribute name="text-align">center</xsl:attribute>
</xsl:attribute-set>
```

2. HTML

对于HTML添加设置：

```
<!--=====
=====
formal(figure/table/equation/example/...) setting
=====
=====-->

<!-- copy from docbook-xsl-ns-1.76.1\html\formal.xsl -->
<xsl:template name="formal.object.heading">
  <xsl:param name="object" select="."/>
  <xsl:param name="title">
    <xsl:apply-templates select="$object" mode="object.title.markup">
      <xsl:with-param name="allow-anchors" select="1"/>
    </xsl:apply-templates>
  </xsl:param>

  <xsl:choose>
    <xsl:when test="$make.clean.html != 0">
      <xsl:variable name="html.class" select="concat(local-name($object),'-title')"/>
      <div class="{ $html.class }">
```

³⁹ http://www.crifan.com/try_to_unify_the_scale_of_figure_in_html_and_pdf_of_docbook/

```
<xsl:copy-of select="$title"/>
</div>
</xsl:when>
<xsl:otherwise>
  <p class="title" align="center">
    <b>
      <xsl:copy-of select="$title"/>
    </b>
  </p>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

详情参考[【已解决】如何设置Docbook中的图片的标题为中间对齐](#)⁴⁰

3.35. 生成PDF中，程序代码programlisting中单行代码太长，超出页面宽度，无法显示

解决办法：添加设置

```
<xsl:attribute-set name="monospace.verbatim.properties">
  <xsl:attribute name="wrap-option">wrap</xsl:attribute>
</xsl:attribute-set>
```

详情参考[【已解决】Docbook生成PDF中，程序代码programlisting中单行代码太长，超出页面宽度，无法显示](#)⁴¹

3.36. Docbook生成的pdf中，当表格在quote中时，表格头部却在底端显示，而且如果内容是多行的，还会显示出多行

原因：是自己误用了quote所导致的。

解决办法：对于block类型的内容，如table等，应该使用blockquote而不是quote。

详情参考[【已解决】Docbook生成的pdf中，当表格在quote中时，表格头部却在底端显示，而且如果内容是多行的，还会显示出多行](#)⁴²

3.37. Docbook中的callout图片在programlisting中不显示

原因：参考的代码是官方的，其用的是areaspec，但是xsltproc不支持areaspec，所以programlisting中不显示callout图片

解决办法：使用co，放到代码中对应位置即可。可参考如下代码：

```
<programlistingco>
...
  chip->cmdfunc(mtd, NAND_CMD_READ0, 0x00, page); <co id="co.cmdfunc"
linkends="co.note.cmdfunc"/>
.....
</programlisting>
```

⁴⁰ http://www.crifan.com/docbook_figure_formal_title_align_center/

⁴¹ http://www.crifan.com/docbook_programlisting_single_line_code_too_long_exceed_page_width_not_show/

⁴² http://www.crifan.com/docbook_when_in_quote_table_header_in_bottom_and_show_multiple_line/

```
<calloutlist>
  <callout id="co.note.cmdfunc" arearefs="co.cmdfunc" >
    <para>要读取数据，肯定是要先发送对应的读页（ read page ）的命令</para>
  </callout>
</calloutlist>
</programlistingco>
```

详情参考 [【已解决】Docbook中的callout图片在programlisting中不显示 -> xsltproc不支持areaspec](http://www.crifan.com/docbook_callout_bug_not_appear_in_programlisting/)⁴³

3.38. An fo:block is wider than the available room in inline-progression-dimension

```
29, 2012 10:36:07 org.apache.fop.events.LoggingEventListener processEvent
An fo:block (See position 668:2859) is wider than the available room in inline-
progression-dimension. Adjusting end-indent based on overconstrained geometry rules
(XSL 1.1, ch. 5.3.4)
29, 2012 10:36:07 org.apache.fop.events.LoggingEventListener processEvent
Line 1 of a paragraph overflows the available area by 4390 millipoints. (See position
668:2859)
29, 2012 10:36:08 org.apache.fop.layoutmgr.table.TableContentLayoutManager
addAreas
tablePositions empty. Please send your FO file to fop-users@xmlgraphics.apache.org
```

出现类似fo:block is wider than the available room in inline-progression-dimension的问题的话，从字面上就能看出，是block比inline宽，超过界限了

我之前遇到此问题，是在不熟悉quote的用法时，将属于block类型的table等内容，放到了属于inline类型的quote中，导致上面的问题的。

详情请参考：[【已解决】Docbook生成的pdf中，当表格在quote中时，表格头部却在底端显示，而且如果内容是多行的，还会显示出多行](http://www.crifan.com/docbook_when_in_quote_table_header_in_bottom_and_show_multiple_line/)⁴⁴

所以遇到此类问题的话，请自行去找找，是不是也误用了某些关键字，把属于block的东西，放到了属于inline中去了。

解决办法：找到误用的关键字，换成适当的關鍵字

3.39. TransformerException: The column-number or number of cells in the row overflows the number of fo:table-columns specified for the table

```
javax.xml.transform.TransformerException: org.apache.fop.fo.ValidationException:
The column-number or number of cells in the row overflows the number of fo:table-
columns specified for the table. (See position 12:37135)
```

从字面上就能看出，是说明某个列数或者是单元格数目，超过了表格所定义的

比如我此处不小心写成的：

```
<table> <title>USB 3.0的引脚定义</title>
```

⁴³ http://www.crifan.com/docbook_callout_bug_not_appear_in_programlisting/

⁴⁴ http://www.crifan.com/docbook_when_in_quote_table_header_in_bottom_and_show_multiple_line/


```
<tgroup cols="4">
...
  <tbody>
...
    <row><entry>Shell</entry><entry>Shell</entry><entry name="col3"
nameend="col5">Shield</entry></row>
  </tbody>
</tgroup>
</table>
```

即在tgroup中定义了cols列数为4，结果不小心在tbody中的某行，错写了某个entry，nameend为col5，超过了本身所定义的最多4列，因此才出现此错误。

解决办法：去对应表格中，仔细找找，肯定是有某处，不小心，写错了，导致列的书目或者是单元格的数目，超过了table的tgroup中所定义了列数了。

3.40. 如何通过xref（在别处）引用某个段落（para）？

可以给para添加一个包裹层，比如用tip包裹起来para，然后给tip添加id属性，然后就可以在别处引用此para了。

后来又知道了Docbook中还有个formalpara，即正式的段落，带标题的段落

如果想要实现段落引用，个人觉得使用formalpara效果会更好，这样，段落还可以带标题。

比如我后来就遇到一个情况，希望引用别处的某段代码，其是在某个para里面的，之前的做法是专门用tip给这段programlisting包裹起来，然后给tip添加id属性，然后在别处引用此源码，现在就可以把tip换成formalpara，然后给formalpara添加id属性，这样就方便别处引用此formalpara了

解决办法：使用[formalpara](http://www.docbook.org/tdg/en/html/formalpara.html)⁴⁵，然后给formalpara添加id属性，即可在别处通过xref引用

3.41. WARNING: Content overflows the viewport of the fo:region-body on page 9 in block-progression direction by 99828 millipoints. (See position 2:5700)

警告说是，有些的内容显示，溢出实际所定义的范围了

出现溢出的东西，有多种，有的是表格内的文字溢出单元格，有的是图片显示超过了当前页面等等，而使得显示出来的效果，超过了其可视范围（viewport）

此溢出问题，有些是无关紧要的，在显示出来的效果上，也看不出有啥异常。

有些则是显示出来的效果，的确有重叠，覆盖等问题。所以，要根据实际情况而定。

解决办法：当显示效果没啥异常的话，可以忽略此警告。如果显示效果的确有问题了，则需要去找到对应的代码的位置，进行微调，比如修改表格中单元格的宽度设置等，以避免内容溢出而导致的内容显示异常问题。

3.42. PDF中callout不能点击跳转（而HTML中却可以）

原因是fo的xsl中co的配置，对于arearef没有正确添加对应的fo:basic-link的internal-destination，导致了生成的pdf中，callout图片不带链接，无法点击。而对应的HTML的xsl配置中，有对于arearef的处理，会生成对应的href，所以HTML中是callout是可以点击的。

解决方法：给fo的xsl中添加如下配置即可：

⁴⁵ <http://www.docbook.org/tdg/en/html/formalpara.html>

```
<!--=====
=====
callout setting
=====
=====-->

<!-- from docbook-xsl-ns-1.76.1\fo\callout.xsl -->
<xsl:template match="d:co">
  <!-- added by crifan start -->
  <fo:inline>
    <fo:basic-link>
      <xsl:if test="@linkends">
        <xsl:attribute name="internal-destination">
          <xsl:value-of select="@linkends"/>
        </xsl:attribute>
      </xsl:if>
    <!-- added by crifan end -->

    <xsl:call-template name="anchor"/>
    <xsl:apply-templates select="." mode="callout-bug"/>
    <!-- added by crifan start -->
  </fo:basic-link>
  <!-- added by crifan end -->
</fo:inline>
</xsl:template>

<xsl:template match="d:coref">
  <!-- tricky; this relies on the fact that we can process the "co" that's -->
  <!-- "over there" as if it were "right here" -->

  <xsl:variable name="co" select="key('id', @linkend)"/>
  <xsl:choose>
    <xsl:when test="not($co)">
      <xsl:message>
        <xsl:text>Error: coref link is broken: </xsl:text>
        <xsl:value-of select="@linkend"/>
      </xsl:message>
    </xsl:when>
    <xsl:when test="local-name($co) != 'co'">
      <xsl:message>
        <xsl:text>Error: coref doesn't point to a co: </xsl:text>
        <xsl:value-of select="@linkend"/>
      </xsl:message>
    </xsl:when>
    <xsl:otherwise>
      <fo:inline>
        <!-- added by crifan start -->
        <fo:basic-link>
          <xsl:if test="@linkend">
            <xsl:attribute name="internal-destination">
              <xsl:value-of select="@linkend"/>
            </xsl:attribute>
          </xsl:if>
        <!-- added by crifan end -->
        <xsl:call-template name="anchor"/>
        <xsl:apply-templates select="$co" mode="callout-bug"/>
        <!-- added by crifan start -->
      </fo:basic-link>
```

```

    <!-- added by crifan end -->
  </fo:inline>
</xsl:otherwise>
</xsl:choose>
</xsl:template>

<!-- from docbook-xsl-ns-1.76.1\fo\lists.xsl -->
<xsl:template name="callout.arearef">
  <xsl:param name="arearef"></xsl:param>
  <xsl:variable name="targets" select="key('id',$arearef)"/>
  <xsl:variable name="target" select="$targets[1]"/>

  <xsl:choose>
    <xsl:when test="count($target)=0">
      <xsl:value-of select="$arearef"/>
      <xsl:text> ???</xsl:text>
    </xsl:when>
    <xsl:when test="local-name($target)='co'">
      <!-- added by crifan start -->
      <fo:basic-link>
        <xsl:attribute name="internal-destination">
          <xsl:value-of select="$arearef"/>
        </xsl:attribute>
        <!-- added by crifan end -->

        <xsl:apply-templates select="$target" mode="callout-bug"/>
        <!-- added by crifan start -->
      </fo:basic-link>
      <!-- added by crifan end -->
    </xsl:when>
    <xsl:when test="local-name($target)='areaset'">
      <xsl:call-template name="callout-bug">
        <xsl:with-param name="conum">
          <xsl:apply-templates select="$target" mode="conumber"/>
        </xsl:with-param>
      </xsl:call-template>
    </xsl:when>
    <xsl:when test="local-name($target)='area'">
      <xsl:choose>
        <xsl:when test="$target/parent::d:areaset">
          <xsl:call-template name="callout-bug">
            <xsl:with-param name="conum">
              <xsl:apply-templates select="$target/parent::d:areaset"
                mode="conumber"/>
            </xsl:with-param>
          </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
          <xsl:call-template name="callout-bug">
            <xsl:with-param name="conum">
              <xsl:apply-templates select="$target" mode="conumber"/>
            </xsl:with-param>
          </xsl:call-template>
        </xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text> ???</xsl:text>
    </xsl:otherwise>
  </xsl:choose>

```

```
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

详情参考：[【全部解决】Docbook生成的PDF中callout不能点击跳转（而HTML中却可以）](#)⁴⁶



提示

相关的支持callout和co的internal-link，以便可以互相点击实现跳转的功能的针对docbook-xsl-ns-1.77.0版本的patch，我已经提交到docbook了：
[add internal-link support for callout and co - ID: 3531945](#)⁴⁷

不知道官网会不会采纳偶的patch，会不会添加到下一个版本中呢，呵呵，期待中...

后记：对于上述我写的代码，已经被Robert Stayton修改增强后，提交了。

此处已确认，截止2013-10-04，最新版本的docbook-ns是1.78.1，已经包含了此部分的代码，从而：

不论在html或pdf中，点击callout，都可以互相跳转了。

3.43. callout图片在programlisting的源码中不显示

callout图片，虽然可以在callout注释中显示，但是却在programlisting的源码中不显示。

其原因在于，xsltproc不支持areaspec，而官网对于programlistingco的例子中，却用的是areaspec，结果就是按照例子写出来的代码，用xsltproc生成的结果，callout图片在源码中不显示。

解决方法：不用arespec，改用co，放在programlisting的代码中具体的位置，即可。示例代码如下：

例 3.1. co用法示例

```
<programlistingco>
  <programlisting language="c">
...
    chip->cmdfunc(mtd, NAND_CMD_READ0, 0x00, page); <co id="co.cmdfunc"
    linkends="co.note.cmdfunc"/>
...
  </programlisting>
  <calloutlist>
    <callout id="co.note.cmdfunc" arearefs="co.cmdfunc" >
      <para>要读取数据，肯定是要先发送对应的读页（read page）的命令</para>
    </callout>
  </calloutlist>
</programlistingco>
```

详情参考：[【已解决】Docbook中的callout图片在programlisting中不显示 -> xsltproc不支持areaspec](#)⁴⁸

⁴⁶ http://www.crifan.com/docbook_callout_in_pdf_can_not_click_to_jump/

⁴⁷ http://sourceforge.net/tracker/?func=detail&aid=3531945&group_id=21935&atid=373749

⁴⁸ http://www.crifan.com/docbook_callout_bug_not_appear_in_programlisting/

3.44. 如何在（只使用co而没有使用areaspec的）源码中多个位置，指向同一个co（callout）？

之前已经实现了源码中的callout，详见：[【已解决】Docbook中的callout图片在programlisting中不显示](#)⁴⁹

但是后来遇到的需求是，希望在源码中的多个位置，都设置同一个co

之前的临时性的办法是，使用linkends指向另一个co：

```
.suspend<co id="co.suspend" linkends="co.note.suspend" /> =  
s3c24xx_nand_suspend,  
.resume<co linkends="co.note.suspend" /> = s3c24xx_nand_resume,
```

很明显，效果很不好。

后来无意间发现有个[coref](#)⁵⁰，发现正是我想要的，对应的用法如下：

例 3.2. coref用法示例

```
.suspend<co id="co.suspend" linkends="co.note.suspend" /> =  
s3c24xx_nand_suspend,  
.resume<coref linkend="co.suspend" /> = s3c24xx_nand_resume,
```

生成的HTML中的效果，如下：

⁴⁹ http://www.crifan.com/docbook_callout_bug_not_appear_in_programlisting/

⁵⁰ <http://www.docbook.org/tdg/en/html/coref.html>

图 3.1. coref在HTML中的效果

其实，要了解，关于驱动框架部分，你所要做的事情的话，

```
static struct platform_driver s3c2410_nand_driver = {
    .probe①          = s3c2410_nand_probe,
    .remove②       = s3c2410_nand_remove,
    .suspend③      = s3c24xx_nand_suspend,
    .resume③       = s3c24xx_nand_resume,
    .driver③       = {
        .name        = "s3c2410-nand",
        .owner       = THIS_MODULE,
    },
};
```

- ① probe就是系统“探测”，就是前面解释的整个过
- ② remove，就是和probe对应的，“反初始化”相
- ③ suspend和resume，对于很多没用到电源管理的I

解决办法：使用coref，设置linkend属性值为某共同的co的id，即可

3.45. 新的docbook-xsl-ns-1.77.0生成的html中的revhistory中单元格无边框

原因：新版的docbook-xsl-ns-1.77.0相对于旧版的docbook-xsl-ns-1.76.1，改变了

```
<xsl:template match="d:revhistory" mode="titlepage.mode">
```

部分的设置，导致新的边框的属性的设置，只应用于整个表格，而单元格的边框没有设置。

解决办法：添加如下配置：

```
<!--=====
=====
revhistory table setting
=====
=====-->

<!--
from docbook-xsl-ns-1.77.0\html\titlepage.xsl
has refer: http://www.w3school.com.cn/css/css_table.asp, but 'solid' not work
-->
<xsl:template match="d:revhistory" mode="titlepage.mode">
  <xsl:variable name="numcols">
    <xsl:choose>
      <xsl:when test="./d:authorinitials|./d:author">3</xsl:when>
      <xsl:otherwise>2</xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
```

```

<xsl:variable name="id"> <xsl:call-template name="object.id"/> </xsl:variable>

<xsl:variable name="title">
  <xsl:call-template name="gentext">
    <xsl:with-param name="key">RevHistory</xsl:with-param>
  </xsl:call-template>
</xsl:variable>

<xsl:variable name="contents">
  <div>
    <xsl:apply-templates select="." mode="common.html.attributes"/>
    <xsl:call-template name="id.attribute"/>
    <table>
      <xsl:if test="$css.decoration != 0">
        <!-- changed by crifan start -->
        <!--
        <xsl:attribute name="style">
          <xsl:text>border-style:solid; width:100%;</xsl:text>
        </xsl:attribute>
        -->
        <xsl:attribute name="border">
          <xsl:text>1px solid black</xsl:text>
        </xsl:attribute>
        <xsl:attribute name="width">
          <xsl:text>100%</xsl:text>
        </xsl:attribute>
        <!-- changed by crifan end -->
      </xsl:if>
      <!-- include summary attribute if not HTML5 -->
      <xsl:if test="$div.element != 'section'">
        <xsl:attribute name="summary">
          <xsl:call-template name="gentext">
            <xsl:with-param name="key">revhistory</xsl:with-param>
          </xsl:call-template>
        </xsl:attribute>
      </xsl:if>
      <tr>
        <th align="{ $direction.align.start}" valign="top" colspan="{ $numcols}">
          <b>
            <xsl:call-template name="gentext">
              <xsl:with-param name="key" select="RevHistory"/>
            </xsl:call-template>
          </b>
        </th>
      </tr>
      <xsl:apply-templates mode="titlepage.mode">
        <xsl:with-param name="numcols" select="$numcols"/>
      </xsl:apply-templates>
    </table>
  </div>
</xsl:variable>

<xsl:choose>
  <xsl:when test="$generate.revhistory.link != 0">

    <!-- Compute name of revhistory file -->
    <xsl:variable name="file">

```

```
<xsl:call-template name="ln.or.rh.filename">
  <xsl:with-param name="is.ln" select="false()"/>
</xsl:call-template>
</xsl:variable>

<xsl:variable name="filename">
  <xsl:call-template name="make-relative-filename">
    <xsl:with-param name="base.dir" select="$chunk.base.dir"/>
    <xsl:with-param name="base.name" select="$file"/>
  </xsl:call-template>
</xsl:variable>

<a href="{ $file}">
  <xsl:copy-of select="$title"/>
</a>

<xsl:call-template name="write.chunk">
  <xsl:with-param name="filename" select="$filename"/>
  <xsl:with-param name="quiet" select="$chunk.quietly"/>
  <xsl:with-param name="content">
    <xsl:call-template name="user.preroot"/>
    <html>
      <head>
        <xsl:call-template name="system.head.content"/>
        <xsl:call-template name="head.content">
          <xsl:with-param name="title">
            <xsl:value-of select="$title"/>
            <xsl:if test="..../d:title">
              <xsl:value-of select="concat(' ', ../d:title, '')"/>
            </xsl:if>
          </xsl:with-param>
        </xsl:call-template>
        <xsl:call-template name="user.head.content"/>
      </head>
      <body>
        <xsl:call-template name="body.attributes"/>
        <xsl:copy-of select="$contents"/>
      </body>
    </html>
    <xsl:text>&#x0a;</xsl:text>
  </xsl:with-param>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
  <xsl:copy-of select="$contents"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

详情参考：[【已解决】新的docbook-xsl-ns-1.77.0生成的html中的revhistory中单元格无边框](http://www.crifan.com/new_77_0_stylesheet_revhistory_table_cell_no_border/)⁵¹

3.46. WARNING: Glyph "?" (0x21b5, carriagereturn) not available in font "MicrosoftYaHei".

⁵¹ http://www.crifan.com/new_77_0_stylesheet_revhistory_table_cell_no_border/

原因：某些个人的特殊字符，此处为Unicode值为0x21b5的特殊的回车键形状的字符：↵，对应的字体无法显示，所以报错。

解决办法：给该特殊字符添加role="xxx",其中xxx为别的某种类型的字体。比如，将原先的源码：

```
<para>回车 ( &crarr; ) 后再输入：</para>
```

改为：

```
<para>回车 ( <phrase role="symbol">&crarr;</phrase> ) 后再输入：</para>
```

详情参考：[【已解决】Docbook用fop生成pdf过程中出现警告：WARNING: Glyph "?" \(0x21b5, carriagereturn\) not available in font "MicrosoftYaHei".](#)⁵²

3.47. 如何实现给table中（多个）特定位置添加相应的注释？

有两种办法：

1. xreflabel + note/tip

通过使用xreflabel的方法，加入到table中特定位置。然后注释部分的内容放在table之外的tip或note中。

缺点是，很不方便用户使用，不直观

参考代码：

```
<table>
  <row>...<entry> <xref linkend="channel_mode"/> (两个单声道)</entry>...</row>
.....
</table>
<tip id="channel_mode" xreflabel="双声道">
  <title>双声道</title>
  双声道文件由两个独立的单声道所组成。大多数解码器把双声道输出成立体声，但是实际上，不是所有的双声道都是立体声的。
</tip>
```

关于xreflabel，更详细的解释，请参考官网的[Linking from other elements](#)⁵³

2. co + calloutlist + callout

在table的对应位置使用co，然后在table之外的calloutlist和callout中添加对应的注释内容。

优点是，可以支持自动编号，且可互相点击跳转。

⁵² http://www.crifan.com/docbook_fop_glyph_not_avaliable_in_font/

⁵³ <http://www.sagehill.net/docbookxsl/CrossRefs.html#LinkingFromOther>

参考代码：

```
<table> <title>xxx</title>
  <tgroup cols="6">
...
    <thead>
      <row>...<entry>控制字符<co id="co.ctrl_char" linkends="co.note.ctrl_char" /
> </entry>...</row>
    </thead>

    <tbody>
...
    </tbody>
  </tgroup>
</table>
<calloutlist>
  <callout id="co.note.ctrl_char" arearefs="co.ctrl_char">
    <para>即在C语言中或其他地方如何表示。</para>
  </callout>
</calloutlist>
```

详情参考：[【已解决】找到更好的方法，给Docbook中的表格（table）添加注释（note）-> 用co（和calloutlist，callout）](http://www.crifan.com/docbook_good_method_add_note_for_table/)⁵⁴

⁵⁴ http://www.crifan.com/docbook_good_method_add_note_for_table/

第 4 章 Docbook参考资料

4.1. Docbook相关的资源和工具下载

- docbook的样式表stylesheet
[Docbook Sourceforge](#)¹

想要下载Docbook的相关的stylesheet，即各种xsl配置文件的话，可以去此docbook官网下载页面去下载

包括各种版本，及最新版本的，docbook-xsl-ns

- 搭建windows下的docbook开发环境所需要的各种工具
之前最早的找到的是zlatkovic的下载页面：<ftp://ftp.zlatkovic.com/pub/libxml/>

后来发现其还有个64bit的子页面，包含更多的，更新版本的，编译好的，win32的，二进制包，供下载：

<ftp://ftp.zlatkovic.com/pub/libxml/64bit/>

之后，又发现，其中部分的内容，xml官网也已经提供下载了：

<http://xmlsoft.org/sources/win32/>

4.2. 关于对于Docbook的介绍和解释方面的资料

1. Docbook的HelloWord
[单个HTML页面的版本](#)²

[多个HTML页面的版本](#)³

[mindmap版本](#)⁴

特点：入门参考经典资料

算是在Docbook的中文方面，最早的，解释的最全的资料了。

2. [Ubuntu Docbook Wiki](#) ⁵

特点:中文,解释的很清晰易懂.

3. [Docbook简介 by 熊伟](#) ⁶



警告

此网站会被赛门铁克检测出病毒

特点：内容不多，但是解释的深入浅出

4. [开源世界旅行手册 之 Docbook指南](#) ⁷

¹ <http://sourceforge.net/projects/docbook/files/>

² http://www.worldhello.net/doc/docbook_howto/docbook_howto.htm

³ http://www.worldhello.net/doc/docbook_howto/index.html

⁴ http://www.worldhello.net/doc/docbook_howto/mindmap/docbook.mm.htm

⁵ <http://wiki.ubuntu.org.cn/DocBookGuide>

⁶ <http://www.turbolinux.com.cn/turbo/wiki/doku.php?id=docbook>

⁷ <http://i.linuxtoy.org/docs/guide/ch27.html>

特点：清晰明了。东西概括得很好，尤其是很多基本概念，解释的到位。

5. [Docbook Beginning](#) ⁸

特点：解释的简洁易懂

6. [DocBook - Tutorial by Lars Vogel](#) ⁹

特点：老外写的，基于Eclipse搭建的环境，内容很新，更新很及时。目前最近的更新是2012.07.02

而且还有国人帮忙翻译了一下：[DocBook与Eclipse - 教程](#) ¹⁰

4.3. 关于对于Docbook的开发过程中可以参考和查阅的资料

1. [DocBook XSL: The Complete Guide – Fourth Edition – by Bob Stayton](#) ¹¹

特点：最权威解释的。

关于docbook的最权威的，官网的教程。

有任何问题，上去找，一般都可以找到答案或参考。

2. [DocBook XSL Stylesheets: Reference Documentation](#) ¹²

特点：参数大全

关于各种输出格式，HTML，FO（PDF），manpage，website等等的参数方面，有哪些参数，具体如何配置，都可以去这里找到。

3. [Linux C编程一站式学习](#) ¹³

特点：（除了本身教程写的好之外，）docbook的xml源码可下载供参考

关于C编程的概念和知识，解释的很透彻，值得一看。

除了教程写的很有价值，其还把docbook的xml源码也公开了，可下载做参考。

相应html也可打包下载。

所以很适合，最开始写docbook源码的时候，做为参考代码。

4. [DocBook 5: The Definitive Guide by Norman Walsh](#) ¹⁴

特点：Norman Walsh写的，最权威的关于docbook5的参考资料

放在后面介绍，并不代表不重要。只是因为英文写的，不适合做入门资料。

适合当参考书，有需要的时候去找相关内容再细看。

5. [DocBook 5 快速起步教程 by tanghw](#) ¹⁵

⁸ <http://www.blogjava.net/hengheng123456789/archive/2007/09/03/142345.html>

⁹ <http://www.vogella.com/articles/DocBook/article.html#xinclude>

¹⁰ <http://www.blogjava.net/antony9/archive/2011/08/07/355908.html>

¹¹ <http://www.sagehill.net/docbookxsl/index.html>

¹² <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/index.html>

¹³ <http://learn.akae.cn/media/index.html>

¹⁴ <http://www.docbook.org/tdg5/en/html/docbook.html>

¹⁵ http://blog.csdn.net/tanghw/article/details/fop_pdf_normal

特点：入门的参考教程

解释的很详细，适合入门者。东西解释的够全，很多知识点都值得参考参考。

只不过其搭建环境是Mac，不是很多人常用的Windows。

6. [DocBook 文件寫作入門 by 老貢生](#)¹⁶

特点：部分内容可做参考

台湾人写的。解释的比较详细。等遇到相关知识，也许可以做参考。

7. [Installing And Using An XML/SGML DocBook Editing Suite by Ashley J.S Mills](#)¹⁷

特点：很多例子，还是可以参考参考的。而且内容也很全面，从环境搭建到例子，全部都有。还介绍了用各种工具，转换成各种不同的格式的用法。

8. [DocBook – Tutorial by Lars Vogel](#)¹⁸

简单介绍了如何写docbook，其中有关于用Eclipse平台的方法。

9. [一张图片总结了Docbook生成各种格式的途经](#)¹⁹

这里一张图片，就总结了，关于Docbook的各种方法，生成各种格式的文件。

10. [DOCBOOK编译环境参考手册 by 廖佳亮](#)²⁰

中文的docbook的教程，写的还不错，有空可以参考参考。

11. [Docbook Frequently Asked Questions](#)²¹

Docbook的FAQ

里面详细介绍了，各种常见问题。

个人觉得比较有帮助的有：

[Docbook tools](#)²²

其中记录了，关于Docbook相关的一些工具，比如xsltproc等相关的问题和解决办法

12. [The Docbook Project](#)²³

Docbook的主页

算是Docbook的入口了，想要找一切关于Docbook的东西，都可以先去那里看看，说不定会有意外发现。

13. 关于Docbook中Table的一些资料

¹⁶ 尤其是DocBook的表格GALS和HTML表格的比较，很值得看看：

¹⁷ <https://supportweb.cs.bham.ac.uk/documentation/tutorials/docsystem/build/tutorials/docbooksys/segmentedhtml/index.html>

¹⁸ <http://www.vogella.com/articles/DocBook/article.html>

¹⁹ <http://nwalsh.com/docbook/procdiagram/xmlsgmlpub.png>

²⁰ <http://www.liaojl.com/docbook/programming/docbook/docbook-build-reference.html>

²¹ <http://www.dpawson.co.uk/docbook/>

²² <http://www.dpawson.co.uk/docbook/tools.html>

²³ <http://docbook.sourceforge.net/>

[Comparison of CALS and HTML tables](#)²⁴

其中，也有关于表格中列的宽度的介绍：

[Column widths](#)²⁵

14. 关于如何添加新的字体
之前别的教程中，最开始部分介绍的关于添加字体的内容，估计就是参考官网的这部分的解释：

[Chapter 13. Print customizations – Adding a font](#)²⁶

15. 关于在生成对应的HTML或者pdf的时候，需要按照自己的需要，配置输出的效果
主要分两种方法，一种是写对应的xsl配置内容，放到自己的xsl文件中去。

另外一种就是使用相应的参数控制。

对应的这些参数有哪些，以及具体含义，在这里可以找到（注：此处是用的xsl是docbook-xsl-ns-1.76.1）：

[DocBook XSL Stylesheets: Reference Documentation](#)²⁷

（也可以下载其[pdf版本](#)²⁸，留着当做参考书）

其中，

对于HTML的参数都在这里：[1. HTML Parameter Reference](#)²⁹

对于（用fo生成的）PDF的参数控制，都在这里：[2. FO Parameter Reference](#)³⁰

关于如何使用这些参数，可以参考一些相关的帖子：

ulink.show：[【已解决】去掉docbook中输出的pdf中，除了带链接的文字之外的那个链接地址](#)³¹

fop1.extensions：[【已解决】Docbook的pdf，去除正文对于标题的缩进indent + 【顺带实现】给docbook的pdf添加bookmark书签的功能](#)³²

16. 想要给pdf添加书签bookmark，以及支持更多种图片的话。
那么可以在使用xsltproc的时候，添加对应的参数即可。

对于使用fop的话：

如果是用的0.20.5或更早的版本，参考这里：[Processor Extensions – fop.extensions](#)³³添加这个参数：

```
--stringparam fop.extensions 1
```

如果是用的0.90及更新的版本，参考这里：[Processor Extensions – fop1.extensions](#)³⁴添加这个参数：

```
--stringparam fop1.extensions 1
```

²⁴ <http://www.sagehill.net/docbookxsl/Tables.html>

²⁵ <http://www.sagehill.net/docbookxsl/ColumnWidths.html>

²⁶ <http://www.sagehill.net/docbookxsl/AddFont.html>

²⁷ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/index.html>

²⁸ <http://docbook.sourceforge.net/release/xsl/current/doc/reference.pdf>

²⁹ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/html/index.html>

³⁰ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/fo/index.html>

³¹ http://www.crifan.com/make_http_link_address_not_show_in_pdf_of_docbook/

³² http://www.crifan.com/remove_indent_of_content_and_add_bookmark_for_docbook_pdf/

³³ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/fo/fop.extensions.html>

³⁴ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/fo/fop1.extensions.html>

我现在所用的fop是1.0的，所以用的是fop1.extensions。

17. 参数配置，可以通过传递--stringparam给xsltproc，也可以通过xsl:param写入到自己的xsl配置文件中
折腾其他参数的过程中，无意间看到：

[使用FOP将中文DocBook xml转换成pdf的实现记录](http://blog.donews.com/limodou/archive/2004/04/01/9917.aspx)³⁵

中是将参数写成这样的：

```
<xsl:param name=" body.font.family" >simsun</xsl:param>
<xsl:param name=" monospace.font.family" >simsun</xsl:param>
<xsl:param name=" title.font.family" >simhei</xsl:param>
<xsl:param name=" saxon.character.representation" select=" 'native' " />
<xsl:param name=" admon.graphics" select=" 1"/>
<xsl:param name=" section.autolabel" select=" 1"/>
<xsl:param name=" section.label.includes.component.label" select=" 1"/>
<xsl:param name=" table.borders.with.css" select=" 1"/>
<xsl:param name=" use.extensions" select=" 1"/>
<xsl:param name=" tablecolumns.extension" select=" 0"/>
<xsl:param name=" callout.unicode" select=" 1"/>
<xsl:param name=" callout.unicode.start.character" select=" 10102"> </xsl:param>
<xsl:param name=" variablelist.as.blocks" select=" 1"> </xsl:param>
<xsl:param name=" callout.graphics" select=" 0"/>
<xsl:param name=" fop.extensions" select=" 1"/>
<xsl:param name=" hyphenate" >false</xsl:param>
<xsl:param name=" l10n.gentext.default.language" select=" 'zh' " />
<xsl:param name=" paper.type" select=" 'A4' " />
<xsl:param name=" draft.mode" select=" 'no' " />
<xsl:param name=" generate.toc" >
```

而与此相对，我之前都是写到--stringparam中的，所以，将原先的命令：

```
xsltproc.exe --stringparam section.autolabel 1 --stringparam
section.label.includes.component.label 1 --stringparam bibliography.numbered
1 --stringparam ulink.show 1 --stringparam ulink.footnotes 1 --stringparam
table.cell.border.color green --stringparam table.frame.border.thickness 2pt --
stringparam fop1.extensions 1 --stringparam body.start.indent 0pt --xinclude -o ../
output/fo/MPEG_VBR.fo /home/CLi/develop/docbook/config/docbook-xsl-ns-1.76.1/fo/
docbook_crl.xsl MPEG_VBR.xml
```

转化为：

```
xsltproc.exe --xinclude -o ../output/fo/MPEG_VBR.fo /home/CLi/develop/docbook/config/
docbook-xsl-ns-1.76.1/fo/docbook_crl.xsl MPEG_VBR.xml
```

加上docbook_crl.xsl中的相关配置：

```
<xsl:param name="admon.graphics" select="1"/>
<xsl:param name="section.autolabel" select="1"/>
```

³⁵ <http://blog.donews.com/limodou/archive/2004/04/01/9917.aspx>

```
<xsl:param name="section.label.includes.component.label" select="1"/>
<xsl:param name="bibliography.numbered" select="1"/>
<xsl:param name="ulink.show" select="1"/>
<xsl:param name="ulink.footnotes" select="1"/>
<xsl:param name="table.cell.border.color">green</xsl:param>
<xsl:param name="table.frame.border.thickness">2pt</xsl:param>
<xsl:param name="fop1.extensions">1</xsl:param>
<xsl:param name="body.start.indent">0pt</xsl:param>
```

这样，每次就不用输入那么多的配置参数了。

而需要额外的参数配置，也就都可以写到xsl文件中了。

此时，也才理解，原来[DocBook XSL Stylesheets: Reference Documentation](#)³⁶

中的HTML和FO (PDF) 的那么多参数，每个都有个语法解释，比如：[table.cell.border.color](#)³⁷中的：

```
<xsl:param name="table.cell.border.color"></xsl:param>
```

指的就是，将此配置，写入到xsl文件中。而之前一直都不知道，都傻傻地加入到xsltproc的参数中去了。

18. 对于catalog相关的知识，这一章有详细介绍：[Part I. Setting up the tools – Chapter 5. XML catalogs](#)³⁸

包括xsltproc中如何使用catalog的内容：
[Using catalogs with xsltproc](#)³⁹

19. 关于pdf实现一定输入效果，会涉及到xsl文件参数配置。
这里：[Docbook Stylesheet Customisation](#)⁴⁰

列出了常见的一些配置参数，包括：章节编号的深度控制，给tip和warning加图片，给programlisting加背景色等。

20. 使用Docbook建网站
如果想要通过Docbook建立网站，可以参考此文：[How to use the Docbook Website system](#)⁴¹

21. 基于Ant的Docbook开发环境：ant4docbook
这里有专门的别人已经帮你搭建好的，基于ant的docbook的开发环境：[ant4docbook](#)⁴²

目前看到的已支持多种格式：[html](#)⁴³，[pdf](#)⁴⁴，[epub](#)⁴⁵，[docx](#)⁴⁶，[rtf](#)⁴⁷，

22. 用Ant编译Docbook

³⁶ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/index.html>

³⁷ <http://docbook.sourceforge.net/release/xsl/1.76.1/doc/html/table.cell.border.color.html>

³⁸ <http://www.sagehill.net/docbookxsl/Catalogs.html>

³⁹ <http://www.sagehill.net/docbookxsl/UseCatalog.html>

⁴⁰ <http://www.dpawson.co.uk/docbook/styling/params.html>

⁴¹ <http://www.dpawson.co.uk/docbook/website/v5/>

⁴² <http://ant4docbook.sourceforge.net/index.html>

⁴³ <http://ant4docbook.sourceforge.net/docbook-html.html>

⁴⁴ <http://ant4docbook.sourceforge.net/docbook-pdf.html>

⁴⁵ <http://ant4docbook.sourceforge.net/docbook-epub.html>

⁴⁶ <http://ant4docbook.sourceforge.net/docbook-docx.html>

⁴⁷ <http://ant4docbook.sourceforge.net/docbook-rtf.html>

ant的官网中，也有介绍：[Create DocBook XML based documentation with Ant](http://wiki.apache.org/ant/CreateDocBookXMLbaseddocumentationwithAnt)⁴⁸

⁴⁸ <http://wiki.apache.org/ant/CreateDocBookDocumentation>

第 5 章 未解决的问题

此处记录Docbook中的一些还没有解决的问题

有人知道解决办法的话，希望可以[联系并告知我](#)¹，谢谢。

5.1. 支持各种语言的代码的语法高亮

折腾过程参见：[【已放弃】给Docbook中代码programlisting实现高亮显示](#)²

5.2. 中文hypertext断字

折腾过程参见：[【部分解决】docbook的programlisting中，单行的连续字符太长，超出pdf边界](#)³和[【已解决】Docbook生成PDF中，程序代码programlisting中单行代码太长，超出页面宽度，无法显示](#)⁴

5.3. 给callout的co重新编号

[【未解决】docbook中给callout的co重新编号](#)⁵

¹ <http://www.crifan.com/about/me/>

² http://www.crifan.com/docbook_highlight_code_for_programlisting_using_xsltproc_fop/

³ http://www.crifan.com/single_line_long_char_exceed_margin_in_programlisting/

⁴ http://www.crifan.com/docbook_programlisting_single_line_code_too_long_exceed_page_width_not_show/

⁵ http://www.crifan.com/docbook_callout_co_renumber_reindex/

第 6 章 Docbook各种元素用法举例

记录Docbook中，各种元素(element)的当前最新版本的示例用法

以下代码示例中，暂定均不含注解（callout），以方便借鉴者，直接拷贝代码，就可以参考使用。

此处，尽量添加实际的例子，以便快速掌握用法。

其中，贴出来的xml源码，部分可能被删减，以防内容太长或者是和此处CDATA等内容冲突。

6.1. chapter, section, sect1-sect5

例 6.1. chapter,section,sect1-sect5

```
<chapter xml:id="crosscompiler_intro"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xl="http://www.w3.org/1999/xlink"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:ns5="http://www.w3.org/1998/Math/MathML"
  xmlns:ns4="http://www.w3.org/1999/xhtml"
  xmlns:ns3="http://www.w3.org/2000/svg"
  xmlns:ns="http://docbook.org/ns/docbook">
<title>交叉编译器简介</title>
<para></para>
<para>相关旧帖：<link xl:href="http://www.crifan.com/
embedded_development_cross_compiler_and_toolchain/">嵌入式开发之交叉编译器</link></
para>
<para></para>
<para></para>
<sect1 xml:id="crosscompiler_naming_rule"><title>交叉编译器的名字的命名规则</title>
  <para>在折腾嵌入式开发，用到交叉编译器的时候，常常会看到这样的名字：</para>
  <para>arm-xscale-linux-gnueabi-gcc</para>
  <para>arm-liunx-gnu-gcc</para>
  <para>等等</para>
  <para>其中，对应的交叉编译器的前缀为：</para>
  <para>arm-xscale-linux-gnueabi-</para>
  <para>arm-liunx-gnu-</para>
  <para>而关于这些名字，我之前也是没注意，或者说很模糊，不是很清楚这些名字是从何而来
的。</para>
  <para>后来，经过折腾了crosstool-ng后，基本上明白了这些名字，是如何生成的。</para>
  <para>其中，貌似此交叉编译器命名的规则，应该是通用的，至少记得是Buildroot中，好像也是
这样命名的。</para>
  <para>下面，就以crosstool-ng为例，参考我之前折腾crosstool-ng期间：</para>
  <para><link xl:href="http://www.crifan.com/
crosstool_tuple_vendor_string_ct_target_vendor/">【整理】crosstool中如何设置xscales的
Tuple' s vendor string(CT_TARGET_VENDOR)</link></para>
  <para>所了解到的内容，来解释解释，这些名字的含义。</para>

  <sect2 xml:id="crosstool_ng_crosscompiler_naming_rule"><title>crosstool-ng中交叉编译前
缀的命名规则</title>
  <para>crosstool-ng中，交叉编译器的（前缀）的名字的命名规则是：</para>
  <screen>arch-vendor-kernel-system</screen>
  <para>对应分别是：</para>
```

```

<sect3 xml:id="crosscompiler_system_part"><title>交叉编译器名字中的system部分</title>
<para>system，直译为，系统</para>
<para>其实主要表示的，交叉编译器所选择的库函数和目标系统</para>
<para>最常见的一些值有，gnu，gnueabi，uclibcgnueabi等等。</para>
<para>其中，此处有几方面的值，表示了几方面的含义：</para>

<sect4 xml:id="system_uclibc"><title>system中的uclibc</title>
<para>uclibc，是c库中的一种</para>
<para>crosstool-ng中，目前支持三种：glibc, eglibc, uclibc</para>
<para>关于三种的关系，详见：</para>
<para><link xl:href="http://www.crifan.com/relation_between_uclibc_glibc_eglibc/">【整理】uclibc, eglibc, glibc之间的区别和联系</link></para>
<para></para>
<sect5 xml:id="system_uclibc_in_crosstool_ng"><title>crosstool-ng中system为uclibc的情况</title>
<para>crosstool-ng中，和system中uclibc对应的值，应该就是"C-library"中的"C library"的值设置为"uClibc"了。</para>
<screen>
C-library
C library (uClibc) --->
</screen>
<para>对应的配置参数是：CT_LIBC_uClibc</para>
<para>对应的help的解释是：</para>
</sect5>
</sect4>
<para></para>
</sect3>
</sect2>
<para></para>
</sect1>

</chapter>

```

6.2. 段落

例 6.2. 普通的段落:para

```

<para>在折腾嵌入式开发，用到交叉编译器的时候，常常会看到这样的名字：</para>
<para><link xl:href="http://www.crifan.com/crosstool_tuple_vendor_string_ct_target_vendor/">【整理】crosstool中如何设置xscales的Tuple's vendor string(CT_TARGET_VENDOR)</link></para>

```

例 6.3. 带标题的正式的段落:formalpara

```

<formalpara><title>段落的标题</title>

```

```
<para>段落的内容</para>
</formalpara>
```

6.3. 图片figure

例 6.4. 带标题的图片:figure

```
<figure xml:id="fg.figure_id"><title>图片的标题</title>
  <mediaobject>
    <imageobject role="html"><imagedata fileref="images/figure_id.png" align="left"
scalefit="0" width="100%" /></imageobject>
    <imageobject role="fo"> <imagedata fileref="images/figure_id.png" align="center"
scalefit="1" width="100%" /></imageobject>
  </mediaobject>
</figure>
```

例 6.5. figure内包含2张mediaobject图片

```
<figure id="fg.hid_use_page"><title>HID Usage page summary</title>
  <mediaobject>
    <imageobject role="html">
      <imagedata fileref="images/hid_use_page.gif" align="left" scalefit="0"
width="100%" />
    </imageobject>
    <imageobject role="fo">
      <imagedata fileref="images/hid_use_page.gif" align="center" scalefit="1"
width="100%" />
    </imageobject>
  </mediaobject>
  <mediaobject>
    <imageobject role="html">
      <imagedata fileref="images/hid_use_page_2.gif" align="left" scalefit="0"
width="100%" />
    </imageobject>
    <imageobject role="fo">
      <imagedata fileref="images/hid_use_page_2.gif" align="center" scalefit="1"
width="100%" />
    </imageobject>
  </mediaobject>
</figure>
```

例 6.6. 不带标题的非正式的图片:informalfigure

```
<informalfigure>
```

```

<mediaobject>
  <imageobject role="html"> <imagedata fileref="images/no_hl_fo_source.jpg"
scalefit="0" width="100%" /> </imageobject>
  <imageobject role="fo"> <imagedata fileref="images/no_hl_fo_source.jpg"
scalefit="1" width="100%" /> </imageobject>
</mediaobject>
</informalfigure>

```

在同一行内显示图片，之前docbook 4使用inlinegraphic

而根据[Inline graphics](http://www.sagehill.net/docbookxsl/InlineGraphics.html)¹，而现在docbook 5已废弃使用inlinegraphic，改用inlinemediaobject。

例 6.7. 行内图片:inlinemediaobject

```

<para>右击cmd的左上角的图标
  <inlinemediaobject>
    <imageobject> <imagedata fileref="images/windows/cmd/cmd_icon.png"/> </
imageobject>
    <textobject> <phrase>cmd的图标</phrase> </textobject>
  </inlinemediaobject>
  , 会出现对应的菜单 :
</para>

```

6.4. 表格，表格嵌套，表格单元格合并

例 6.8. 表格:table

```

<table xml:id="tbl.usb_class"> <title>USB Class表</title>
  <tgroup cols="3">
    <colspec colnum="1" colname="col1" colwidth="1*" />
    <colspec colnum="2" colname="col2" colwidth="1*" />
    <colspec colnum="3" colname="col3" colwidth="4*" />

    <thead>
      <row> <entry>Base Class</entry> <entry>Descriptor Usage</
entry> <entry>Description</entry> </row>
    </thead>

    <tbody>
      <row> <entry>00h</entry> <entry>Device </entry> <entry>Use class information in
the Interface Descriptors</entry> </row>
      <row> <entry>01h</entry> <entry>Interface</entry> <entry>Audio</entry> </row>
      <row> <entry>02h</entry> <entry>Both</entry> <entry>Communications and CDC
Control</entry> </row>
      <row> <entry>03h</entry> <entry>Interface</entry> <entry>HID (Human Interface
Device)</entry> </row>
      <row> <entry>05h</entry> <entry>Interface</entry> <entry>Physical</entry> </
row>

```

¹ <http://www.sagehill.net/docbookxsl/InlineGraphics.html>

```

        <row> <entry>06h</entry> <entry>Interface</entry> <entry>Image</entry> </row>
        <row> <entry>07h</entry> <entry>Interface</entry> <entry>Printer</entry> </row>
        <row> <entry>08h</entry> <entry>Interface</entry> <entry>Mass Storage</
entry> </row>
        <row> <entry>09h</entry> <entry>Device</entry> <entry>Hub</entry> </row>
        <row> <entry>0Ah</entry> <entry>Interface</entry> <entry>CDC-Data</entry> </
row>
        <row> <entry>0Bh</entry> <entry>Interface</entry> <entry>Smart Card</entry> </
row>
        <row> <entry>0Dh</entry> <entry>Interface</entry> <entry>Content Security</
entry> </row>
        <row> <entry>0Eh</entry> <entry>Interface</entry> <entry>Video</entry> </row>
        <row> <entry>0Fh</entry> <entry>Interface</entry> <entry>Personal Healthcare</
entry> </row>
        <row> <entry>DCh</entry> <entry>Both</entry> <entry>Diagnostic Device</
entry> </row>
        <row> <entry>E0h</entry> <entry>Interface</entry> <entry>Wireless Controller</
entry> </row>
        <row> <entry>EFh</entry> <entry>Both</entry> <entry>Miscellaneous</entry> </
row>
        <row> <entry>FEh</entry> <entry>Interface</entry> <entry>Application Specific</
entry> </row>
        <row> <entry>FFh</entry> <entry>Both</entry> <entry>Vendor Specific</entry> </
row>
    </tbody>
</tgroup>
</table>

```

例 6.9. 带嵌套 (embedded) 表格:informaltable

```

<table id="tbl.mpeg_xing_header_format" align="left"> <title>XING 头的格式及含义</title>
  <tgroup cols="4">
    <colspec colnum="1" colname="col1" colwidth="2*" />
    <colspec colnum="2" colname="col2" colwidth="2*" />
    <colspec colnum="3" colname="col3" colwidth="8*" />
    <colspec colnum="4" colname="col4" colwidth="3*" />
    <thead>
      <row> <entry>位置 ( 字节 ) </entry> <entry>长度 ( 字节 ) </entry> <entry>含义</
entry> <entry>示例</entry> </row>
    </thead>
    <tbody>
      <row> <entry>0</entry> <entry>4</entry> <entry>4个ASCII字符的VBR头 ID, 要么是Xing, 要么是Info, 无NULL结尾 ( 普通字符串都以NULL,即\0结尾 ) </entry> <entry>'Xing'</
entry> </row>
      <row> <entry>4</entry> <entry>4</entry>
        <entry> <para>存放一个标志, 用于表示接下来存在哪些域/字段,各字段逻辑或的结果 :
          <informaltable>
            <tgroup cols="2" >
              <colspec colnum="1" colwidth="1*" />
              <colspec colnum="2" colwidth="2*" />
              <tbody>
                <row> <entry>0x0001</entry> <entry>存在总帧数 ( Frames ) 字段</
entry> </row>

```



```

entry> </row>
        <row> <entry>0x0002</entry> <entry>存在文件大小 ( Bytes ) 字段</
        <row> <entry>0x0004</entry> <entry>存在TOC字段</entry> </row>
        <row> <entry>0x0008</entry> <entry>存在音频质量指示字段</entry> </
row>
        </tbody>
    </tgroup>
</informaltable> </para>
</entry>
<entry> <para>0x0007 就表示下面存在 :
    <simplelist type='vert' columns='1'>
        <member>总帧数</member>
        <member>文件大小总字节数</member>
        <member>TOC表</member>
    </simplelist> </para>
</entry> </row>
<row> <entry>8</entry> <entry>4</entry> <entry> <emphasis>总帧数(Frames)</
emphasis> , 大端[可选]</entry> <entry>7344</entry> </row>
<row> <entry>8或12</entry> <entry>4</entry> <entry>文件总大小, 单位字节, 大端
[可选]</entry> <entry>45000</entry> </row>
<row> <entry>8, 12, 16</entry> <entry>100</entry> <entry>TOC表, 大端[可选]</
entry> <entry> </entry> </row>
<row> <entry>8或12, 16, 108, 112 , 116</entry> <entry>4</entry> <entry>音频质量
指示, 最差0, 最好100, 大端[可选]</entry> <entry>0</entry> </row>
    </tbody>
</tgroup>
</table>

```

6.5. 注释callout

例 6.10. programlisting的callout:programlistingco

```

<programlistingco>
  <programlisting language="c">
static struct platform_driver s3c2410_nand_driver = {
  .probe<co id="co.probe" linkends="co.note.probe" /> = s3c2410_nand_probe,
  .remove<co id="co.remove" linkends="co.note.remove" /> = s3c2410_nand_remove,
  .suspend<co id="co.suspend" linkends="co.note.suspend" /> = s3c24xx_nand_suspend,
  .resume<coref linkend="co.suspend" /> = s3c24xx_nand_resume,
  .driver = {
    .name = "s3c2410-nand",
    .owner = THIS_MODULE,
  },
};
  </programlisting>
  <calloutlist>
    <callout id="co.note.probe" arearefs="co.probe" >
      <para>probe就是系统“探测”，就是前面解释的整个过程，这个过程的多数步骤，都是和你自己的Nand Flash相关的，尤其是那些硬件初始化部分，是你必须要自己实现的。</para>
    </callout>
    <callout id="co.note.remove" arearefs="co.remove" >
      <para>remove，就是和probe对应的，“反初始化”相关的动作。主要是释放系统相关资源和关闭硬件的时钟等常见操作了。</para>
    </callout>
  </calloutlist>
</programlistingco>

```

```

</callout>
<callout id="co.note.suspend" arearefs="co.suspend" >
  <para>suspend和resume，对于很多没用到电源管理的情况下，至少对于我们刚开始写基本的驱动的时候，可以不用关心，放个空函数即可。</para>
</callout>
</calloutlist>
</programlistingco>

```

对应的screenco，就是把programlistingco和programlisting换为screenco和screen即可。

screenco举例如下：

例 6.11. screen的callout:screenco

```

<screenco>
  <screen>
crosstool-ng-1.18.0<co id="co.ctng_src_folder" linkends="co.note.ctng_src_folder" />
crosstool-ng-1.18.0_build<co id="co.ctng_build_folder"
linkends="co.note.ctng_build_folder" />
src<co id="co.download_src_folder" linkends="co.note.download_src_folder" />
x-tools<co id="co.x_tools_folder" linkends="co.note.x_tools_folder" />
crosstool-ng-1.18.0.tar.bz2<co id="co.ctng_tar_file" linkends="co.note.ctng_tar_file" />
  </screen>
  <calloutlist>
    <callout id="co.note.ctng_src_folder" arearefs="co.ctng_src_folder" >
      <para>crosstool-ng的源码包：<xref linkend="co.ctng_tar_file" />，解压后的文件夹</para>
    </callout>
    <callout id="co.note.ctng_build_folder" arearefs="co.ctng_build_folder" >
      <para>专门为后期使用crosstool-ng去建立交叉编译器，的编译（ build ），而专门建立的文件夹</para>
    </callout>
    <callout id="co.note.download_src_folder" arearefs="co.download_src_folder" >
      <para>为crosstool-ng中，后续需要下载各种软件的源码包，而准备的，</para>
    </callout>
    <callout id="co.note.x_tools_folder" arearefs="co.x_tools_folder" >
      <para>这个文件夹，是，用crosstool-ng所生成的交叉编译器，所在的路径。</para>
    </callout>
  </calloutlist>
  <screen>(&HOME;/develop/crosstool-ng/x-tools/&CT_TARGET) Prefix directory</screen>
  <para></para>
  <para>的配置，用来指定生成的交叉编译器，存放在何处。</para>
  <para>此时，就是去设置为此处对应的路径即可。</para>
  <callout id="co.note.ctng_tar_file" arearefs="co.ctng_tar_file" >
    <para>很明显，这个就是之前我在折腾crosstool-ng-1.18.0时，去下载的源码包了。</para>
  </callout>
</screenco>

```

```

    <para>对应的上面的<xref linkend="co.ctng_src_folder" />，就是此源码包解压后的路
径。</para>
    <para></para>
  </callout>
</calloutlist>
</screenco>

```

对应的table中添加callout，就是在table的任意位置使用co，然后再table之外使用calloutlist+callout即可。

下面这个例子，是自己摸索出来的，通过在任意位置使用co，然后别处直接可以使用coref引用该co，且不一定需要对应的callout和calloutlist：

例 6.12. 没有callout的co实现在任意位置使用co然后在别处使用coref去引用

```

<orderedlist>
  <listitem><emphasis>Evacuation</emphasis> <co id="co.evacuation"
linkends="co.note.evacuation" /> procedures are implemented whenever the fire alarms have
been activated.</listitem>
  <listitem>The following rules and policies are binding for every employee who works at the
<emphasis>premises</emphasis> <co id="co.premises" linkends="co.note.premises" /> of
M&M Software (Suzhou) GmbH</listitem>
</orderedlist>

<table xml:id="tbl.eng_words_1_20"><title>1-20单词的含义</title>
  <tgroup cols="4">
    <colspec colnum="1" colname="col1" colwidth="1*" />
    <colspec colnum="2" colname="col2" colwidth="2*" />
    <colspec colnum="3" colname="col3" colwidth="2*" />
    <colspec colnum="4" colname="col4" colwidth="4*" />

    <thead align="center">
      <row><entry>序号</entry><entry>英文单词</entry><entry>中文含义</
entry><entry>说明注释</entry></row>
    </thead>

    <tbody>
      <row><entry><coref linkend="co.evacuation" id="co.note.evacuation" /></
entry><entry>evacuation</entry><entry>疏散，撤离</entry><entry>指失火了，要逃离的情
况</entry></row>
      <row><entry><coref linkend="co.premises" id="co.note.premises" /></
entry><entry>premises</entry><entry>经营场所</entry><entry>此处是公司地址，场所的意
思</entry></row>
    </tbody>
  </tgroup>
</table>

```

不过目前此种用法，有个小bug，那就是生成的HTML中鼠标虽可点击，但是无法跳转，不过生成的PDF中是正常的支持鼠标点击co和coref互相跳转的。

6.6. 链接，引用

链接，使用link代替旧的ulink：

注意，是在顶层的chapter，sect1等中，定义对应xlink的域名的

例 6.13. 带xlink域名的link

```
<sect1
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"

  xml:id="lan.python">
<title>Python</title>
.....
  <sect2 xml:id="lan.python.crifanlib"><title>crifan的Python库：crifanLib.py</title>
    <para>之前在折腾<link xl:href="http://www.crifan.com/crifan_released_all/website/
python/blogstowordpress/">BlogsToWordPress：将百度空间，网易163，新浪sina，QQ空间，人
人网，CSDN，搜狐Sohu等博客搬家到WordPress </link>的过程中，先后遇到很多个问题，然后基
本上也都自己解决了。对应的也写了相应的代码和函数。</para>
    .....
  </sect2>
</sect1>
```

引用，很简单，就是用xref而已：

例 6.14. xref引用

```
<tip>
  <para>上述gif动画演示，只支持HTML在浏览器中的显示。其他格式输出中，比如PDF中，不支持
此gif动画。所以下面再用文字解释一下大概流程：<xref linkend="text_explain_run_py_in_cmd" /
></para>...
</tip>
<para>用文字简述就是：</para>
<formalpara xml:id="text_explain_run_py_in_cmd"><title>用文字解释如何在Windows的cmd中
运行Python脚本BlogsToWordpress.py</title>
  <orderedlist>
    .....
  </orderedlist>
</formalpara>
```

6.7. 等式

例 6.15. 等式equation

```
<equation xml:id="eq.cbr_duration"><title>CBR播放时间 ( CBR Duration ) </title>
  <mathphrase>
    <para>CBR Duration</para>
    <para>= File Size(Byte) × 8 bit/Byte ÷ ( Bitrate ( K bit/s ) × 1000 bit/Kbit ) </para>
    <para>CBR播放时间</para>
    <para>= 文件大小 ( 字节 ) × 8比特/字节 ÷ ( 比特率 千比特/秒 ×1000 比特/千比特 ) </para>
  </mathphrase>
</equation>
```

6.8. list列表系列

例 6.16. itemizedlist

```
<itemizedlist>
  <listitem>
    <para>xx</para>
    <para>xx</para>
    <para>xx</para>
  </listitem>
  <listitem>
    <para>yy</para>
    <para>yy</para>
    <para>yy</para>
  </listitem>
</itemizedlist>

<itemizedlist>
  <listitem>xxxxxx
    <para>xxx</para>
    <para>xxx</para>
    <para>xxx</para>
  </listitem>
  <listitem>xxxxxx
    <para>yyy</para>
    <para>yyy</para>
    <para>yyy</para>
  </listitem>
</itemizedlist>
```

例 6.17. 带emphasis的itemizedlist

```
<itemizedlist>
  <listitem><emphasis>xxx</emphasis>
    <para>xx</para>
    <para>xx</para>
    <para>xx</para>
  </listitem>
```

```

<listitem> <emphasis>yyy</emphasis>
  <para>yy</para>
  <para>yy</para>
  <para>yy</para>
</listitem>
</itemizedlist>

```

例 6.18. 带指定编号类型的orderedlist

```

<orderedlist numeration="arabic">
  <listitem>总的帧数：<para>VBR中的总的帧的数目。</para></listitem>
  <listitem>帧的采样个数：<para>对于MP3（MPEG1，Layer III）来说，是固定的1152个采样。</para></listitem>
  <listitem>帧的采样率：<para>通过解析第一帧，即可得知帧采样率索引，查表，即可得此采样率。</para></listitem>
</orderedlist>

```

6.9. 简单列表simplelist , literallayout

例 6.19. 简单列表simplelist

```

<table id="mp3_content_struct">
  <title>MP3文件的内容组织结构</title>
  <tgroup cols="1">
    <colspec colnum="1" colname="col1" colwidth="1*"/>
    <tbody>
      <row><entry><para>第二帧，格式如下：
        <simplelist type='horiz' columns='2'>
          <member>[ID3.....]</member><member>ID3 V2的头，大多数最新的MP3，都有这个头</member>
          <member>[APE 头]</member><member>用于APE格式的头，现在也用于MPEG</member>
        </simplelist></para></entry></row>
      .....
      <row><entry>[TAG .....] 128字节的ID3 V1信息，如果没有前面的ID3 V2，多数都有这个ID3 V1的头</entry></row>
    </tbody>
  </tgroup>
</table>

```

例 6.20. 简单列举literallayout

```

<table id="tbl.nand_third_id"> <title>Nand Flash第3个ID的含义</title>
  <tgroup cols="7">
    <colspec colnum="1" colname="col1" colwidth="3*" />
    .....
    <thead>
      <row> <entry></entry> <entry>Description</entry> <entry>I/O7</entry> <entry>I/
O6</entry> <entry>I/O5 I/O4</entry> <entry>I/O3 I/O2</entry> <entry>I/O1 I/O0</
entry></row>
    </thead>
    <tbody>
      <row> <entry>Internal Chip Number</entry> <entry> <emphasis> <literallayout>1
2
4
8
</literallayout> </emphasis> </entry> <entry></entry> <entry></entry> <entry></
entry> <entry></entry> <entry><literallayout>0    0
0    1
1    0
1    1
</literallayout> </entry> </row>
    .....
    </tbody>
  </tgroup>
</table>

```

6.10. xinclude文件，xinclude文件中的其中一部分的内容

在主文件docbook_dev_note.xml中：

例 6.21. xinclude主文件中使用xinclude

```

<?xml version='1.0' encoding="utf-8"?>
.....
<book version="5.0"
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"

  xml:lang="zh_CN"
  >
<info>
  <title>Docbook开发手记</title>
  <subtitle></subtitle>
.....
</info>

<xi:include href="preface.xml" />
<xi:include href="ch01_build_env.xml" />
.....
</book>

```


然后被包含的文件中，是正常的xml文件。

比如preface.xml：

例 6.22. 被xinclude包含的子文件：preface.xml

```
<?xml version='1.0' encoding="utf-8"?>

<!DOCTYPE preface>

<preface
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"

  xml:id="preface">
<title>Docbook介绍</title>
<sect1 xml:id="what_is_docbook"> <title>什么是Docbook</title>
.....
</sect1>
</preface>
```

比如ch01_build_env.xml：

例 6.23. 被xinclude包含的子文件：ch01_build_env.xml

```
<?xml version='1.0' encoding="utf-8"?>

<!DOCTYPE chapter
[
.....
]>

<chapter
  xmlns="http://docbook.org/ns/docbook"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xmlns:xl="http://www.w3.org/1999/xlink"

  xml:id="ch01_build_env">

<title>Windows下的Docbook的环境搭建</title>
<abstract> </abstract>
  <sect1> <title>搭建Docbook之前需要知道的最基本的事情</title>
.....
  </sect1>
</chapter>
```

6.11. filename : 文件 , 文件夹 , 后缀名 ,

当去表示一个普通文件 , 文件夹 , 符号链接 , 分区 , 库文件 , 头文件 , 设备文件等等内容时 , 可以用filename这个关键字。

关于filename的详细语法 , 可参考 : [docbook element: filename](http://www.docbook.org/tdg/en/html/filename.html)²

例 6.24. filename是文件 , 文件夹的例子

<para>此处 , 只是去修改对应的<filename>.build/src/linux-custom/scripts/unifdef.c/</filename> , 即可。</para>

<para>比如 : 我之前 , 除了 , 解压crosstool所得的文件夹<filename class="directory">crosstool-ng-1.18.0</filename>之外 , 去建立了一个对应的文件夹 : <filename class="directory">crosstool-ng-1.18.0_build</filename></para>

<para>同时 , 为了后续crosstool-ng下载对应的各个包 , 也建立了对应的<filename class="directory">src</filename>和<filename class="directory">x-tools</filename>两个文件夹。</para>

例 6.25. 示例 : filename是后缀名

<para>提示 : 其中可以看到<link xl:href="http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html">Notepad++</link>自动帮你写好了<filename class="extension">.py</filename>后缀 , 那是因为你之前设置了Python语法高亮。</para>

6.12. admonitions : note , caution , warning , tip和important

Docbook中有个一系列的术语叫做admonitions , 包含了note , caution , warning , tip和important

6.12.1. note

例 6.26. note举例

```
<note xml:id="note.install_java_first"> <title>如果没有安装java , 请先安装java</title>
  <para>fop的运行 , 依赖于java运行环境。</para>
  <para>如果你的windows中 , 还没有安装java , 请先安装java运行环境 ( JRE=java runtime environment ) 。</para>
  <para>关于java的基本知识和如何安装 , 不了解的请参考 : <link xl:href="http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#java">Java</link></para>
```

² <http://www.docbook.org/tdg/en/html/filename.html>

```
<para>java已经安装好前提下，另外记得确保JAVA_HOME环境变量已经正确设置好了。</para>
<para>比如，我的系统中的是：</para>
<screen>JAVA_HOME=D:\Program Files\Java\jre7</screen>
</note>
```

6.12.2. caution

例 6.27. caution举例

```
<caution xml:id="not_make_ldflags_lintl_lcurses"><title>此处不要通过make时加LDFLAGS参数
去添加-lintl和-lcurses</title>
<para>其实，上面贴出来的，修改makefile的办法，的确已经解决了此处的，缺少intl和缺少
ncurses的库的两个问题了。</para>
<para>只是，后来，在另外一个Cygwin环境下去折腾同样的make编译crosstool-ng的时候：</
para>
<para><link xl:href="http://www.crifan.com/
cygwin_crosstool_ng_configure_make_make_install">【记录】Cygwin下配置编译和安装
crosstool-ng</link></para>
<para>由于之前解决上面这个问题的时候：</para>
<para><link xl:href="http://www.crifan.com/
crosstool_ng_cygwin_zconf_tab_o_zconf_tab_c_text_0x162a_undefined_reference_to_libintl_gettext-2/">【已
解决】Cygwin 1.7.17下make编译crosstool-ng出错：zconf.tab.o:zconf.tab.c:(.text+0x162a):
undefined reference to `_libintl_gettext'</link></para>
<para>用的解决办法是：</para>
<para>不去修改makefile，而直接给make加上LDFLAGS参数加上-lintl：</para>
<screen>make LDFLAGS="-lintl"</screen>
<para>当时也是可以解决此问题的。</para>
</caution>
```

6.12.3. warning

6.12.4. tip

例 6.28. tip举例

```
<tip xml:id="tip.how_set_win_env"><title>如何设置windows环境变量</title>
<para>关于如何添加相应路径到windows的环境变量中，不熟悉的可以参考：</para>
<para><link xl:href="http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/
html/soft_dev_basic.html#win_env_var">Windows的环境变量</link></para>
</tip>
```

6.12.5. important

6.13. 一些inline元素： keycap , guibutton , keycombo

参考：[Inline Docbook Elements](#)³

而看到，Docbook中有个一系列的术语叫做内嵌元素(inline elements)，比如，keycap，guibutton，keycombo

这些元素，可以通过css配置而达到3D显示的效果：

6.13.1. keycap

例 6.29. keycap举例

```
<para>按<keycap>PrtScn</keycap>键可以截屏。</para>
```

此处，就用上面的代码，加上合适的css去配置，在浏览器中打开编译输出的网页，就能看到下面的这个PrtScn可以显示出3D的效果：

按键可以截屏。

6.14. programlisting代码和screen屏幕输出

docbook中用于代码和屏幕输出等内容，对应的是programlisting和screen



XML中的实体和CDATA

对于docbook中写programlisting和screen等方面的东西

会涉及到CDATA方面的事情，其和XML的中的实体有关。

对于XML的实体不了解的，可以去参考：[XML中的实体引用](#)⁴

6.14.1. programlisting代码

6.14.1.1. 带CDATA的programinglist示例

例 6.30. 举例：带CDATA的programinglist

³ <http://www.badgers-in-foil.co.uk/projects/docbook-css/tests/inlines-gen.xml>

⁴ http://www.w3school.com.cn/xml/xml_syntax.asp

```
<programlisting language="c">
<![CDATA[
for (setno = 0; setno < nr_sets; setno++, nmtd++) {
pr_debug("initialising set %d (%p, info %p)\n", setno, nmtd, info);
s3c2410_nand_init_chip(info, nmtd, sets);
nmtd->scan_res = nand_scan_ident(&nmtd->mtd,(sets) ? sets->nr_chips : 1);

if (nmtd->scan_res == 0) {
s3c2410_nand_update_chip(info, nmtd);
nand_scan_tail(&nmtd->mtd);
s3c2410_nand_add_partition(info, nmtd, sets);
}
if (sets != NULL)
sets++;
}
]]>

</programlisting>
```

上述，就是由于`programlisting`的内容中出现了`&`，`<`，`>`，才用`<![CDATA[和]]>`括起来的。

6.14.1.2. 不带CDATA的programinglist示例

例 6.31. 举例：不带CDATA的programinglist

```
<programlisting language="c">
for (setno = 0; setno &lt; nr_sets; setno++, nmtd++) {
pr_debug("initialising set %d (%p, info %p)\n", setno, nmtd, info);
s3c2410_nand_init_chip(info, nmtd, sets);
nmtd-&gt;scan_res = nand_scan_ident(&nmtd-&gt;mtd,(sets) ? sets-&gt;nr_chips : 1);

if (nmtd-&gt;scan_res == 0) {
s3c2410_nand_update_chip(info, nmtd);
nand_scan_tail(&nmtd-&gt;mtd);
s3c2410_nand_add_partition(info, nmtd, sets);
}
if (sets != NULL)
sets++;
}
</programlisting>
```

上述，就是由于`programlisting`的内容中出现了：`&`，`<`，`>`

所以才分别用：`&`，`<`，`>`，去替换的。

6.14.2. screen屏幕输出

`screen`适合用于表现：对于软件开发等过程中，屏幕输出信息之类的内容

6.14.2.1. 带CDATA的屏幕输出screen

例 6.32. 举例：带CDATA的screen

```
<para>查看单个的某个示例配置的核心参数，用：</para>  
<screen><![CDATA[ct-ng show-<sample>]]></screen>
```

上述，就是由于screen的内容中出现了<，>，才用<![CDATA[和]]>括起来的。

6.14.2.2. 不带CDATA的屏幕输出screen

例 6.33. 举例：不带CDATA的screen

```
<para>查看单个的某个示例配置的核心参数，用：</para>  
<screen>ct-ng show-&lt;sample&gt;</screen>
```

上述，就是由于screen的内容中出现了：<，>

所以才分别用：<，>，去替换的。

第 7 章 Docbook相关的知识和概念

7.1. 输出类型：html和打印输出print out

在将docbook的xml源码，转换输出为其他格式的时候，最常见的两大类是：HTML和fo

然后基于html，有多种对应的可选的输出类型，比如单个html，多个html，htmlhelp（即chm），webhelp等等。

而基于fo，则可以再转换为其他的格式，比如pdf，ps等等。

此处，在docbook中，html和fo，则分别对应着：html和print out打印输出，算是两种不同的输出类型。

7.2. docbook中的条件编译：profiling

Docbook中有个类似于一般编程中的条件编译的东西，叫做：

[Profiling](http://www.sagehill.net/docbookxsl/Profiling.html)¹

即，你在xml源码中，加上一些参数，然后在输出生成HTML或FO的时候，会根据你所设置的条件，去执行相应的动作。

比如：

例 7.1. 添加role参数实现条件编译

```
<figure id="fg.nand_read_op_flow_2"><title>Nand Flash数据读取操作的时序图</title>
  <mediaobject>
    <imageobject role="html">
      <imagedata fileref="images/nand_read_op_flow.png" align="center" scalefit="0"
width="100%" />
    </imageobject>
    <imageobject role="fo">
      <imagedata fileref="images/nand_read_op_flow.png" align="center" scalefit="1"
width="100%" />
    </imageobject>
  </mediaobject>
</figure>
```

然后编译的时候，就会根据你所输出的是HTML，就调用上面的那个fileref中scalefit="0"的配置，而生成FO的时候，fileref的配置中的，就使用scalefit="1"了。

7.3. 关于docbook中的图片方面的内容

一般常见模式是，在xml源码同路径下，存在一个images文件夹，然后把自己文章相关的图片放进去。

此类图片的地址，在xml中都是很简单的images/xxx.jpg之类的。

¹ <http://www.sagehill.net/docbookxsl/Profiling.html>

另一些图片，就是和docbook系统相关的了，其随着docbook的xsl文件一同发布的，在docbook-xsl-ns-1.76.1\images中可以找到。

共分好几类，包括：

1. callout
1,2,。。。, 30
2. admon
caution, important, note, tip, warning
3. navi
home, prev, next, up

基本上，每种类型，都有对应的矢量的svg图片，和普通的bitmap类型的gif/jpg/png图片。

7.4. docbook的一些变体

7.4.1. 简化版的docbook

首先我们要知道的是，对于docbook本身，这个规范

其包含了很多元素，比如book, title, para, sect1, sect2, 等等等等，

参考：[Writing Documentation Using DocBook](#)²

一共有超过400个元素（还不包括元素的各种属性）

而后来，由于大家在实际使用中，发现貌似完整版的docbook，有点太复杂了。

而很多时候，比如只将docbook用于写个article，技术手册(white paper)等等情况时，只需要用到其中部分的元素，就够用了。

所以，后来就又弄出一个docbook的简化版：[Simplified DocBook](#)

简化版的docbook，只包含了大约119个元素（还有其他相关的555多个实体，29个符号）

很明显，相对于400多个元素的，完整版的docbook，还是要简化不少。

注意，此处的简化版的docbook，并非官网标准，而是所谓的community project，即由docbook社区发起的，逐渐丰富和制定出的一个标准。

简化版的docbook的主页：

[Simplified DocBook DTD](#)³

相关页面：

[Simplified DocBook Home](#)⁴

[Simplified DocBook](#)⁵

7.4.2. docbook(制作)幻灯片

类似于上面的简化版的docbook，这个docbook slides，也是个社区标准，而非官方标准。

² <http://xml.web.cern.ch/XML/goossens/dbatcern/dbatcern.html>

³ <http://docbook.org/schemas/sdocbook/>

⁴ <http://www.oasis-open.org/docbook/xml/simple/4.1.2.5/>

⁵ <http://docbook.org/schemas/5x-custom#simple>

其中，最新版本的docbook-xsl-ns中，可以找到对应的样式表stylesheet，比如：docbook-xsl-ns-1.78.1/slides

相关页面：

[DocBook "Slides"](#) ⁶

7.4.3. docbook(制作)网站

类似于上面的简化版的docbook，这个docbook websites，也是个社区标准，而非官方标准。

其中，最新版本的docbook-xsl-ns中，可以找到对应的样式表stylesheet，比如：docbook-xsl-ns-1.78.1/website

相关页面：

[DocBook "Website"](#) ⁷

⁶ <http://docbook.org/schemas/5x-custom.html#slides>

⁷ <http://docbook.org/schemas/5x-custom.html#website>

附录 A. 已通过Docbook发布的资料

A.1. 基础知识相关

1. [字符编码详解](#)¹
2. [软件开发基础知识](#)²
3. [各种计算机语言简介和总结](#)³
4. [JSON详解](#)⁴
5. [正则表达式学习心得](#)⁵
6. [计算机编程语言基础知识](#)⁶
7. [详解抓取网站，模拟登陆，抓取动态网页的原理和实现（Python，C#等）](#)⁷

A.2. 嵌入式相关

1. [【详解】如何编写Linux下Nand Flash驱动](#)⁸
2. [Uboot中start.S源码的指令级的详尽解析](#)⁹
3. [MPEG简介 + 如何计算CBR和VBR的MP3的播放时间](#)¹⁰
4. [Linux MTD下获取Nand flash各个参数的过程的详细解析](#)¹¹
5. [如何在Linux下写无线网卡的驱动](#)¹²
6. [【详解】嵌入式开发中固件的烧录方式](#)¹³
7. [在Linux运行期间升级Linux系统（Uboot+kernel+Rootfs）](#)¹⁴
8. [GNU Binutils详解](#)¹⁵
9. [交叉编译详解](#)¹⁶
10. [crosstool-ng详解](#)¹⁷
11. [USB基础知识概论](#)¹⁸

¹ http://www.crifan.com/files/doc/docbook/char_encoding/release/html/char_encoding.html

² http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html

³ http://www.crifan.com/files/doc/docbook/language_summary/release/html/language_summary.html

⁴ http://www.crifan.com/files/doc/docbook/json_tutorial/release/html/json_tutorial.html

⁵ http://www.crifan.com/files/doc/docbook/regular_expression/release/html/regular_expression.html

⁶ http://www.crifan.com/files/doc/docbook/programming_language_basic/release/html/programming_language_basic.html

⁷ http://www.crifan.com/files/doc/docbook/web_scrape_emulate_login/release/html/web_scrape_emulate_login.html

⁸ http://www.crifan.com/files/doc/docbook/linux_nand_driver/release/html/linux_nand_driver.html

⁹ http://www.crifan.com/files/doc/docbook/uboot_starts_analysis/release/html/uboot_starts_analysis.html

¹⁰ http://www.crifan.com/files/doc/docbook/mpeg_vbr/release/html/mpeg_vbr.html

¹¹ http://www.crifan.com/files/doc/docbook/nand_get_type/release/html/nand_get_type.html

¹² http://www.crifan.com/files/doc/docbook/linux_wireless/release/html/linux_wireless.html

¹³ http://www.crifan.com/files/doc/docbook/firmware_download/release/html/firmware_download.html

¹⁴ http://www.crifan.com/files/doc/docbook/runtime_upgrade_linux/release/html/runtime_upgrade_linux.html

¹⁵ http://www.crifan.com/files/doc/docbook/binutils_intro/release/html/binutils_intro.html

¹⁶ http://www.crifan.com/files/doc/docbook/cross_compile/release/html/cross_compile.html

¹⁷ http://www.crifan.com/files/doc/docbook/crosstool_ng/release/html/crosstool_ng.html

¹⁸ http://www.crifan.com/files/doc/docbook/usb_basic/release/html/usb_basic.html

12. [如何实现Linux下的U盘 \(USB Mass Storage \) 驱动](#) ¹⁹
13. [USB HID Learning Record](#) ²⁰
14. [详解ARM的AMBA设备中的DMA设备PL08X的Linux驱动](#) ²¹
15. [硬件电路基础知识](#) ²²
16. [ARM与MIPS的详细对比](#) ²³
17. [【详解】中断相关的知识](#) ²⁴
18. [现场总线Field Bus简析](#) ²⁵
19. [Code 128 Symbology Introduction](#) ²⁶
20. [GS1-128条形码和相关的AI及FNC1的详解](#) ²⁷
21. [Plessey & MSI Symbology Introduction](#) ²⁸
22. [UPC/UPC-A/UPC-E & EAN Barcode Symbology](#) ²⁹

A.3. 上层软件相关

1. [Docbook开发手记](#) ³⁰
2. [建设网站详细教程](#) ³¹
3. [网站搬家详解](#) ³²
4. [详解crifan的C#库：crifanLib.cs](#) ³³
5. [C#学习心得](#) ³⁴
6. [Python语言总结](#) ³⁵
7. [python初级教程：入门详解](#) ³⁶
8. [python中级教程：开发总结](#) ³⁷
9. [Python专题教程：字符串和字符编码](#) ³⁸
10. [Python专题教程：抓取网站，模拟登陆，抓取动态网页](#) ³⁹

- ¹⁹ http://www.crifan.com/files/doc/docbook/usb_disk_driver/release/html/usb_disk_driver.html
- ²⁰ http://www.crifan.com/files/doc/docbook/usb_hid/release/html/usb_hid.html
- ²¹ http://www.crifan.com/files/doc/docbook/dma_pl08x_analysis/release/html/dma_pl08x_analysis.html
- ²² http://www.crifan.com/files/doc/docbook/hardware_basic/release/html/hardware_basic.html
- ²³ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/html/arm_vs_mips.html
- ²⁴ http://www.crifan.com/files/doc/docbook/interrupt_related/release/html/interrupt_related.html
- ²⁵ http://www.crifan.com/files/doc/docbook/fieldbus_intro/release/html/fieldbus_intro.html
- ²⁶ http://www.crifan.com/files/doc/docbook/symbology_code128/release/html/symbology_code128.html
- ²⁷ http://www.crifan.com/files/doc/docbook/symbology_gs128/release/html/symbology_gs128.html
- ²⁸ http://www.crifan.com/files/doc/docbook/symbology_plessey/release/html/symbology_plessey.html
- ²⁹ http://www.crifan.com/files/doc/docbook/symbology_upc/release/html/symbology_upc.html
- ³⁰ http://www.crifan.com/files/doc/docbook/docbook_dev_note/release/html/docbook_dev_note.html
- ³¹ http://www.crifan.com/files/doc/docbook/build_website/release/html/build_website.html
- ³² http://www.crifan.com/files/doc/docbook/website_transfer/release/html/website_transfer.html
- ³³ http://www.crifan.com/files/doc/docbook/crifanlib_csharp/release/html/crifanlib_csharp.html
- ³⁴ http://www.crifan.com/files/doc/docbook/csharp_summary/release/html/csharp_summary.html
- ³⁵ http://www.crifan.com/files/doc/docbook/python_summary/release/html/python_summary.html
- ³⁶ http://www.crifan.com/files/doc/docbook/python_beginner_tutorial/release/html/python_beginner_tutorial.html
- ³⁷ http://www.crifan.com/files/doc/docbook/python_intermediate_tutorial/release/html/python_intermediate_tutorial.html
- ³⁸ http://www.crifan.com/files/doc/docbook/python_topic_str_encoding/release/html/python_topic_str_encoding.html
- ³⁹ http://www.crifan.com/files/doc/docbook/python_topic_web_scrape/release/html/python_topic_web_scrape.html

A.4. 其他

1. [crifan推荐软件](#) ⁴⁰
2. [【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++](#) ⁴¹
3. [英语学习笔记](#) ⁴²
4. [买房详细教程](#) ⁴³
5. [电脑基础知识教程](#) ⁴⁴
6. [虚拟机教程](#) ⁴⁵
7. [VirtualBox教程](#) ⁴⁶
8. [VMWare教程](#) ⁴⁷

⁴⁰ http://www.crifan.com/files/doc/docbook/crifan_rec_soft/release/html/crifan_rec_soft.html

⁴¹ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/html/rec_soft_npp.html

⁴² http://www.crifan.com/files/doc/docbook/english_learn/release/html/english_learn.html

⁴³ http://www.crifan.com/files/doc/docbook/buy_house/release/html/buy_house.html

⁴⁴ http://www.crifan.com/files/doc/docbook/compute_basic/release/html/compute_basic.html

⁴⁵ http://www.crifan.com/files/doc/docbook/virutal_machine_tutorial/release/html/virutal_machine_tutorial.html

⁴⁶ http://www.crifan.com/files/doc/docbook/virtualbox_tutorial/release/html/virtualbox_tutorial.html

⁴⁷ http://www.crifan.com/files/doc/docbook/vmware_tutorial/release/html/vmware_tutorial.html

参考书目

- [1] [理解docbook](#)¹
- [2] [XML中的实体引用](#)²
- [3] [DTD vs XML Schema](#)³
- [4] [xml的命名空间namespace](#)⁴
- [5] [XML Entity Declarations for Characters](#)⁵
- [6] [RELAX NG - XML syntax](#)⁶
- [7] [一个简单的Docbook 5.0例子](#)⁷
- [8] [在Windows上安装配置Apache FOP](#)⁸
- [9] [DocBook 5 differences](#)⁹
- [10] [Profiling](#)¹⁰
- [11] [Graphics](#)¹¹
- [12] [Using XIncludes with DocBook 5](#)¹²
- [13] [Selecting part of a file](#)¹³
- [14] [qandaset](#)¹⁴
- [15] [docbook element: filename](#)¹⁵
- [16] [Cygwin](#)¹⁶
- [17] [Notepad++的多种编码支持](#)¹⁷
- [18] [【已解决】给Docbook生成的静态HTML页面添加统计访问次数的代码](#)¹⁸
- [19] [【记录】将docbook的xsl-ns从当前的docbook-xsl-ns-1.77.1升级到docbook-xsl-ns-1.78.1](#)¹⁹
- [20] [【整理】制作pdf文件后，保持原先word文档中的页内链接和超链接+常见pdf打印机安装使用方法简介](#)²⁰

¹ <http://hi.baidu.com/zighouse/item/a288301637c2fedbbf9042dc>

² http://www.w3school.com.cn/xml/xml_syntax.asp

³ http://www.w3school.com.cn/schema/schema_howto.asp

⁴ http://www.w3school.com.cn/xml/xml_namespaces.asp

⁵ <http://www.w3.org/2003/entities/>

⁶ http://en.wikipedia.org/wiki/RELAX_NG#XML_syntax

⁷ <http://easwy.com/blog/archives/a-simple-docbook-5-example/>

⁸ <http://easwy.com/blog/archives/install-apache-fop-on-windows/>

⁹ <http://www.sagehill.net/docbookxsl/Db5Tools.html#Db5Diffs>

¹⁰ <http://www.sagehill.net/docbookxsl/Profiling.html>

¹¹ <http://www.sagehill.net/docbookxsl/Graphics.html>

¹² <http://www.sagehill.net/docbookxsl/ProcesingDb5.html>

¹³ <http://www.sagehill.net/docbookxsl/ModularDoc.html#UsingXinclude>

¹⁴ <http://www.docbook.org/tdg/en/html/qandaset.html>

¹⁵ <http://www.docbook.org/tdg/en/html/filename.html>

¹⁶ http://www.crifan.com/files/doc/docbook/cygwin_intro/release/html/cygwin_intro.html

¹⁷ http://www.crifan.com/files/doc/docbook/rec_soft_npp/release/htmls/npp_func_multi_enc.html

¹⁸ http://www.crifan.com/docbook_add_code_to_statistic_static_html_page_view/

¹⁹ http://www.crifan.com/docbook_xsl_ns_upgrade_from_1_77_1_to_latest_1_78_1/

²⁰ http://www.crifan.com/order_make_pdf_file_to_keep_the_original_word_document_page_links_and_hyperlinks_within_the_common_pdf_printer_installation_method_introduction/

- [21] [【终极解决】fop错误：Exception in thread "main" java.lang.NoClassDefFoundError:org/apache/xmlgraphics/image/loader/ImageContext 的终极解决办法，即cygpath有bug，转换路径出错，导致部分路径被截断²¹](#)

²¹ http://www.crifan.com/fop_java_lang_noclassdeffounderor_xmlgraphics_cygpath_bug_path_truncated/