The DocBook Schema

Working Draft V5.0b5, 12 April 2006

Document identifier:
docbook-5.0b5-spec-wd-01

Location:
http://www.oasis-open.org/docbook/specs

Editor:
Norman Walsh, Sun Microsystems, Inc. <Norman.Walsh@Sun.COM>

Abstract:

DocBook is a general purpose [XML] schema particularly well suited to books and papers about computer hardware and software (though it is by no means limited to these applications).

The Version 5.0 release is a complete rewrite of DocBook in RELAX NG. The intent of this rewrite is to produce a schema that is true to the spirit of DocBook while simultaneously removing inconsistencies that have arisen as a natural consequence of DocBook's long, slow evolution. The Technical Committee has taken this opportunity to simplify a number of content models and tighten constraints where RELAX NG makes that possible.

The Technical Committee provides the DocBook 5.0 schema in other schema languages, including W3C XML Schema and an XML DTD, but the RELAX NG Schema is now the normative schema.

Status:

This is a Working Draft. It does not necessarily represent the consensus of the committee.

Please send comments on this specification to the <docbook@lists.oasis-open.org> list. To subscribe, please use the OASIS Subscription Manager.

The errata page for this specification is at http://www.oasis-open.org/docbook/specs/docbook5-errata.html.

Table of Contents

1. Introduction 2
2. Terminology 2
3. The DocBook RELAX NG Schema V5.0 2
    3.1. Changes in DocBook V5.0b5 2
    3.2. Changes in DocBook V5.0b4 3
    3.3. Changes in DocBook V5.0b3 3

3.4. Changes in DocBook V5.0b2 4
3.5. Changes in DocBook V5.0 4
4. Release Notes 9

Appendixes

A. The DocBook Media Type 9
1. Registration of MIME media type application/docbook+xml 9
2. Fragment Identifiers 10
B. OASIS DocBook Technical Committee (Non-Normative) 10
C. Notices 10
D. Intellectual Property Rights 11
E. Revision History 11
References 11

1. Introduction

DocBook is general purpose XML schema particularly well suited to books and papers about computer hardware and software (though it is by no means limited to these applications).

The DocBook Technical Committee maintains the DocBook schema. Starting with V5.0, DocBook is normatively available as a [RELAX NG] Schema (with some additional Schematron assertions). W3C XML Schema and Document Type Definition (DTD) versions are also available.

The Version 5.0 release is a complete rewrite. In programming-language terms, think of it as a code refactoring.

This rewrite introduces a large number of backwards-incompatible changes. Essentially all DocBook V4.x documents will have to be modified to validate against DocBook V5.0. An XSLT 1.0 stylesheet is provided to ease this transition.

The DocBook Technical Committee welcomes bug reports and requests for enhancement (RFEs) from the user community. The current list of outstanding requests is available through the SourceForge tracker interface. This is also the preferred mechanism for submitting new requests. Old RFEs, from a previous legacy tracking system, are archived for reference.

2. Terminology

The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and *optional* in this Working Draft are to be interpreted as described in [RFC 2119]. Note that for reasons of style, these words are not capitalized in this document.

3. The DocBook RELAX NG Schema V5.0

The DocBook RELAX NG Schema is distributed from the DocBook site at OASIS. DocBook is also available from the mirror on http://docbook.org/.

3.1. Changes in DocBook V5.0b5

This release contains several improvements over V5.0b4.

1. Restored the `class` attribute on `refmiscinfo` (removing the `type` attribute introduced in V5.0b4). The `class` attribute is now an enumerated list with the standard `otherclass` extension point.

2. Added `parameter` to `db.technical.inlines`. This allows `parameter` to occur in places like `userinput` and `computeroutput`.

3. Allow XInclude elements in `info` elements (in the **docbookxi** schemas).

4. Fixed bugs in the build process that resulted in broken DTD versions of beta 4 and earlier betas.

3.2. Changes in DocBook V5.0b4

This release contains several improvements over V5.0b3.

1. Fixed RFE 1416903: Added a `cover` element to hold additional material for document covers. Updated reference documentation.

2. Corrected a typo in the list of values allowed on the `class` attribute of `biblioid`: changed "pubnumber" to "pubsnumber" (note the "s"). This is consistent with its use as a replacement for the `pubsnumber` tag that has been removed in DocBook V5.0.

3. Fixed a bug in the content model of the various "`info`" elements. In previous beta releases, the title-related elements (`title`, `titleabbrev`, and `subtitle`) were erroneously required to appear first. The requirement is only that they appear exactly or at most once, depending on the context.

4. Renamed the "`sgmlcomment`" attribute value of the `class` attribute of `tag`. There's no significant difference between XML and SGML comments and the "SGML" name implies that there ought to be an "`xmlcomment`" value, which there is not. The new value is simply "`comment`".

5. Renamed the "`class`" attribute of `refmiscinfo`. The DocBook semantics of class attributes is that they have enumerated values. This attribute should always have been called "`type`" as it is now.

6. Updated `renderas` on `bridgehead` and `class` on `othercredit` to have "attribute/otherattribute" co-constraints. (In other words, if you select "other" for `renderas` on `bridgehead` or `class` on `othercredit`, you have to also provide a value for `otherrenderas` or `othercredit`, respectively.

7. Changed `width` attribute in media objects to be "text" instead of "`xs:integer`".

8. Fixed bug in the build process that resulted in unusable XML Schema versions of beta 2 and beta 3.

9. Improved reference documentation for attributes on many elements.

3.3. Changes in DocBook V5.0b3

This release contains several small improvements over V5.0b2.

1. Fixed RFE 1358844: allow multiple `imageobject`s inside an `imageobjectco`. Updated reference documentation.

2. Restored default values to the `type` attribute on `simplelist` and the `choice` and `rep` attributes on `methodparam`, `arg`, and `group`. Fixed a bug in `paramdef` where `plain` was accidentally allowed as a `choice`. These defaults are reflected in the generated XML DTD as well.

3. Reduced the content model of `blockquote` which seemed way too broad.

4. Improved reference documentation for attributes on many elements.

3.4. Changes in DocBook V5.0b2

This release addresses several bugs identified in V5.0b1.

1. When SVG or MathML are used, allow more than one element from the respective namespace to be used in the appropriate location.

2. Fixed RFE 1356238: the `xrefstyle` attribute on `olink` is now "`text`" rather than "`xsd:IDREF`".

3. Fixed RFE 1380477: Make `xml:id` optional on `area`s within `areaset`; allow linking attributes on `areaset`; establish the semantics that an `area` inside an `areaset` inherits its linking attributes from the `areaset` if it doesn't have linking attributes of its own.

4. Allow `alt` inside `equation`, `informalequation`, and `inlineequation`.

5. Fixed RFE 1356254: `dbforms.rnc` schema now supports the HTML form elements.

3.5. Changes in DocBook V5.0

In V5.0, DocBook has been rewritten as a native RELAX NG grammar. The goals of this redesign were to produce a schema that:

1. "feels like" DocBook. Most existing documents should still be valid or it should be possible to transform them in simple, mechanical ways into valid documents.

2. enforces as many constraints as possible in the schema. Some additional constraints are expressed with Schematron rules.

3. cleans up the content models.

4. gives users the flexibility to extend or subset the schema in an easy and straightforward way.

5. can be used to generate XML DTD and W3C XML Schema versions of DocBook.

Under the ordinary operating rules of DocBook evolution, the only backwards incompatible changes that could be made in DocBook V5.0 were those announced in DocBook V4.0. In light of the fact that this is a complete rewrite, the Technical Committee gave itself the freedom to make "unannounced" backwards-incompatible changes for this one release.

3.5.1. Removing Legacy Elements

A number of elements have been removed from DocBook. Many of these have been replaced by simpler, more versatile alternatives. Others have simply been removed because they are not believed to be widely used.

Table 1. DocBook Element Changes

Element(s)	Explanation
`articleinfo`, `bookinfoinfo`, ..., `*info`	Replaced by `info`, see Section 3.5.3, "Uniform Info Elements".
`authorblurb`	Replaced by `personblurb`. This more general name better reflects the fact that it is available in elements other than `author` (e.g., `editor`).
`collabname`, `corpauthor`, `corpcredit`, `corpname`	Replaced by `orgname` and the updated content models of `author`, `editor`, and `othercredit`.
`graphic`, `graphicco`, `inlinegraphic`, `mediaobjectco`	Removed in favor of `mediaobject` and `inlinemediaobject`.
`isbn`, `issn`, `pubsnumber`	Replaced by `biblioid`.
`lot`, `lotentry`, `tocback`, `tocchap`, `tocfront`, `toclevel1`, `toclevel2`, `toclevel3`, `toclevel4`, `toclevel5`, `tocpart`	Replaced by simpler `tocdiv` element.
`ulink`	Replaced by ubiquitous linking, see Section 3.5.9, "Universal Linking".
`sgmltag`	Replaced by `tag`.
`action`, `beginpage`, `highlights`, `interface`, `invpartnumber`, `medialabel`, `modespec`, `structfield`, `structname`	Removed.

3.5.2. Smaller Content Models

The content models of many inlines have been reduced, sometimes drastically. The parameter entity customization of DocBook V4.x and previous versions resulted in very broad content models for some inlines.

Consider, for example, `command` in DocBook V4.4:

```
command ::=
   (#PCDATA|link|olink|ulink|action|application|classname|methodname|
   interfacename|exceptionname|ooclass|oointerface|ooexception|
   command|computeroutput|database|email|envar|errorcode|errorname|
   errortype|errortext|filename|function|guibutton|guiicon|guilabel|
   guimenu|guimenuitem|guisubmenu|hardware|interface|keycap|keycode|
   keycombo|keysym|literal|code|constant|markup|medialabel|
   menuchoice|mousebutton|option|optional|parameter|prompt|property|
   replaceable|returnvalue|sgmltag|structfield|structname|symbol|
   systemitem|uri|token|type|userinput|varname|nonterminal|anchor|
   remark|subscript|superscript|inlinegraphic|inlinemediaobject|
   indexterm|beginpage)*
```

In DocBook V5.0, `command` has a much smaller, more rational content model:

```
command ::=

  * Zero or more of:
      o text
      o alt
      o anchor
      o annotation
      o biblioref
      o indexterm
      o inlinemediaobject
      o link
      o phrase
      o remark
      o replaceable
      o subscript
      o superscript
      o xref
```

DocBook V5.0 may be overzealous in its simplification of content models. The Technical Committee expects to adjust these simplifications during user testing. Users are encouraged to report places where formally valid documents can no longer be made valid because content models have been reduced.

3.5.3. Uniform Info Elements

DocBook V4.x has `setinfo`, `bookinfo`, `chapterinfo`, `appendixinfo`, `sectioninfo`, etc. DocBook would be smaller and simpler if it had a single `info` element in all these places.

There's an historical reason for the large number of unique names: customizers might very well want to adjust the content models of info elements at different levels. For example, a copyright statement might be required at the book level, or an author forbidden at the sub-section level. In DTDs, there's only one content model allowed per element name, so in order to support independent customization, each info element must have a different name.

In RELAX NG, no such limitation exists. We can use patterns to achieve both a single `info` element while still allowing customizers to change its content model in different contexts. In light of this functionality, we've replaced all the various flavors of info with a single element name.

3.5.4. Required Titles

DocBook V5.0 enforces the constraint that titles are required on `article`s and other large structures where they are effectively optional in DocBook V4.x. (They are optional only in the sense that DTDs are unable to enforce the constraint that they be present, the documentation has always made it clear that titles were required.)

3.5.5. Required Version

In DocBook V4.x and earlier, the presence of a document type declaration served as a mechanism for identifying the DocBook version of a document. Although the declaration was not actually required, it was present in the vast majority of DocBook documents.

In RELAX NG, no similar declaration exists. Although a document type declaration might still be present, it seems likely that this will not usually be the case.

Nevertheless, downstream processors may benefit from some indication of the version of DocBook being used. As a result DocBook V5.0 adds a new `version` attribute which *must* be present on the document element of a DocBook document.

Mixing versions is explicitly allowed and the version attribute may be used on other elements as well. This might be the case, for example, in a compound document constructed from multiple documents each with its own version.

3.5.6. Co-Constraints

DocBook V5.0 enforces attribute co-constraints such as the `class`/`otherclass` attributes on `biblioid`.

3.5.7. Improved HTML and CALS Table Support

In DocBook V5.0, HTML tables and CALS tables are independently specified. Where the DTD of DocBook V4.x allows for incoherent mixing of the two models, DocBook V5.0 forbids such mixtures.

3.5.8. Data Types

DocBook V5.0 adds a few simple data types. For example, the `cols` attribute on `tgroup` must be a positive integer.

Some of these constraints, such as the requirement that elements like `pubdate` include a proper date-time type, may prove controversial. Users are encouraged to report places where formally valid documents can no longer be made valid because data types have been introduced.

3.5.9. Universal Linking

Starting with DocBook V5.0, the `linkend` and `xlink:href` attributes are available on almost all elements.

The `linkend` attribute provides an ID/IDREF link within the document. The `xlink:href` attribute provides a URI-based link.

The `ulink` element has been removed from DocBook as URI-based links can now be achieved directly from the appropriate inline (such as `productname` or `command`). For instances where no specific semantic inline is needed, `link` is still available. Where `link` used to be limited to ID/IDREF linking, it now sports an `xlink:href` attribute as well.

Support for extended links are provided through the `extendedlink`, `arc`, and `locator` elements.

3.5.10. Improved Accessibility

Accessibility is improved by allowing both inline and block annotations in most context. The `alt` element is now allowed in most places for inline annotations, the new element `annotation` supports block annotations.

3.5.11. Simplified Table of Contents Markup

The DocBook V4.x markup for Tables of Contents, or more generally for Lists of Titles, was complex and had not evolved quite in step with the rest of DocBook. In DocBook V5.0, it has all been replaced by a quite simple, recursive `toc`/`tocdiv`/`tocentry` structure.

While most Tables of Contents and Lists of Titles are generated automatically and authors never have to produce markup for them by hand, this simplified content model should make it easier for authors to generate them when necessary. One possible application of hand-authored `toc` markup is to generate custom hierarchies which can be assembled on-the-fly from a library of topics marked up in DocBook.

3.5.12. Extra-Grammatical Constraints

Grammar based validation technologies (like RELAX NG) and rule based validation technologies (like Schematron) are naturally complementary. Mixing them allows us to play to the strengths of each without stretching either to enforce constraints that they aren't readily designed to enforce.

For example, DocBook NG requires that the root element of a document have an explicit version attribute. Because there are a great many elements that can be root elements in DocBook, and because they can almost all appear as descendants of a root element as well, it would be tedious to express this constraint in RELAX NG. But it is easy in a rule-based schema language.

DocBook V5.0 uses Schematron where appropriate.

3.5.13. Customization

From the very beginning, one of the goals of DocBook has been that users should be able to produce customizations that are either subsets of extensions of DocBook.

Customization is possible in DocBook V4.x, but because of the intricacies of XML DTD syntax and the complex and highly stylized patterns of parameter entitiy usage in DocBook, it's not as easy as we would like it to be.

In DocBook V5.0, we hope to take advantage of RELAX NGs more robust design (and it's lack of pernicious determinism rules) to make customization easier.

Three schema design patterns get us most of the way there.

3.5.13.1. Logical Groupings

DocBook elements, particularly the inlines, can be divided into broad classes: general purpose, technical, error-related, operating-system related, bibliographic, publishing, etc. In DocBook V5.0, these are collected together in named patterns.

To add a new inline, `endpoint` for example, to the list of technical inlines, one need only extend the appropriate pattern. If an element should appear in several classes, they can all be extended in the same way:

```
db.technical.inlines |= endpoint
db.programming.inlines |= endpoint
db.os.inlines |= endpoint
```

Much the same concept was used in DocBook V4.x, where instead of patterns we had parameter entities. However, the constraints of DTD validation severely limit the circumstances under which an element can appear twice in a content model. That meant that adding an element to one parameter entity might make it an error to add it to another. Such constraints do not exist in RELAX NG which greatly simplifies the customization.

3.5.13.2. Element Definitions

Each element in DocBook V5.0 is defined by its own pattern. To change the content model of an element, only that pattern need be redefined. To remove an element from DocBook, that pattern can be redefined as "`notAllowed`".

3.5.13.3. Attribute Definitions

Each attribute list in DocBook V5.0 is defined by its own pattern. To change the list of attributes available on an element, only that pattern need be redefined. To remove all the attributes, that pattern can be redefined as "`empty`".

3.5.14. Conversion

There's an XSLT 1.0 stylesheet for performing conversion from DocBook V4.x to DocBook V5.0. Presented with a valid DocBook V4.x document, it attempts to produce a valid DocBook V5.0 document.

It succeeds entirely automatically for the most part, though human intervention is suggested for constructs that might have multiple interpretations (and therefore multiple possible transformations).

Users are encouraged to report documents that are not successfully transformed by the stylesheet, especially those which do have valid DocBook V5.0 representations.

4. Release Notes

See http://www.relaxng.org/ for a list of tools that can validate an XML document using RELAX NG. Note that not all products are capable of evaluating the Schematron assertions in the schema.

A. The DocBook Media Type

This appendix registers a new MIME media type, "`application/docbook+xml`".

1. Registration of MIME media type application/docbook+xml

MIME media type name:	`application`
MIME subtype name:	`docbook+xml`
Required parameters:	None.
Optional parameters:	`charset` This parameter has identical semantics to the `charset` parameter of the `application/xml` media type as specified in [RFC 3023] or its successors.
Encoding considerations:	By virtue of DocBook XML content being XML, it has the same considerations when sent as "`application/docbook+xml`" as does XML. See [RFC 3023], Section 3.2.
Security considerations:	Several DocBook elements may refer to arbitrary URIs. In this case, the security issues of RFC 2396, section 7, should be considered.
Interoperability considerations:	None.
Published specification:	This media type registration is for DocBook documents as described by [DocBook: TDG5].
Applications which use this media type:	There is no experimental, vendor specific, or personal tree predecessor to "`application/docbook+xml`", reflecting the fact that no applications currently recognize it. This new type is being registered in order to allow for the deployment of DocBook on the World Wide Web, as a first class XML application.

Additional information:	Magic number(s):	There is no single initial octet sequence that is always present in DocBook documents.
	File extension(s):	DocBook documents are most often identified with the extension "**.xml**".
	Macintosh File Type Code(s):	TEXT
Person & email address to contact for further information:	Norman Walsh, `<ndw@nwalsh.com>`.	
Intended usage:	COMMON	
Author/Change controller:	The DocBook specification is a work product of the DocBook Technical Committee at OASIS.	

2. Fragment Identifiers

For documents labeled as "`application/docbook+xml`", the fragment identifier notation is exactly that for "`application/xml`", as specified in [RFC 3023] or its successors.

B. OASIS DocBook Technical Committee (Non-Normative)

The following individuals were members of the committee during the formulation of this Working Draft:

- Steve Cogorno, Sun Microsystems
- Gary Cornelius, Individual
- Adam Di Carlo, Debian
- Paul Grosso, Arbortext
- Dick Hamilton, Hewlett-Packard
- Nancy Harrison, IBM
- Scott Hudson, Individual
- Mark Johnson, Debian
- Jirka Kosek, Individual
- Larry Rowland, Hewlett-Packard
- Michael Smith, Individual
- Robert Stayton, Individual (Secretary)
- Norman Walsh, Sun Microsystems (Chair, Editor)

C. Notices

D. Intellectual Property Rights

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the DocBook web page (http://www.oasis-open.org/committees/docbook/)

E. Revision History

Working Draft “Beta 4”	9 March 2006
Working Draft “Beta 3”	1 February 2006
Working Draft “Beta 2”	12 January 2006
Working Draft “Beta 1”	27 October 2005
Working Draft “Alpha 1”	26 June 2005

References

Normative

[RELAX NG] James Clark, editor. *RELAX NG Specification (Committee Specification)*. OASIS. 2001.

[XML] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, *et. al.*, editors. *Extensible Markup Language (XML) 1.0 (Third Edition)*. World Wide Web Consortium, 04 Feb 2004.

[XLink11] Steven DeRose, Eve Maler, David Orchard, Norman Walsh, editors. *XML Linking Language (XLink) Version 1.1*. World Wide Web Consortium, 2005.

[RFC 2119] IETF (Internet Engineering Task Force). *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels*. S. Bradner. 1997.

[RFC 3023] IETF (Internet Engineering Task Force). *RFC 3023: XML Media Types*. M. Murata, S. St. Laurent, D. Kohn. 2001.

[DocBook: TDG5] Norman Walsh and Leonard Meullner. *DocBook 5.0: The Definitive Guide.*

Non-Normative

[SGML] JTC 1, SC 34. *ISO 8879:1986 Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)*. 1986.

[W3C XML Schema] Henry S. Thompson, David Beech, Murray Maloney, et. al., editors. *XML Schema Part 1: Structures*. World Wide Web Consortium, 2000.

[W3C XML Datatypes] Paul V. Biron and Ashok Malhotra, editors. *XML Schema Part 2: Datatypes*. World Wide Web Consortium, 2000.

[Schematron] Rick Jelliffe, editor. *The Schematron Assertion Language 1.5*. Rick Jelliffe and Acedemia Sinica Computing Centre. 2001, 2001.