DocBook

The Definitive Guide

Norman Walsh Leonard Muellner

May 19, 2003

Legal Notice

Back Cover Text: Visit O'Reilly on the Web at www.oreilly.com http://www.oreilly.com/.

Preface

DocBook provides a system for writing structured documents using SGML or XML. It is particularly well-suited to books and papers about computer hardware and software, though it is by no means limited to them. DocBook is a document type definition (DTD). Because it is a large and robust DTD, and because its main structures correspond to the general notion of what constitutes a book, DocBook has been adopted by a large and growing community of authors. DocBook is supported "out of the box" by a number of commercial tools, and support for it is rapidly growing in a number of free software environments. In short, DocBook is an easy-to-understand and widely used DTD. Dozens of organizations use DocBook for millions of pages of documentation, in various print and online formats, worldwide.

This book is designed to be the clear, concise, normative reference to the DocBook DTD. This book is the official documentation for the DocBook DTD.

We hope to answer, definitively, all the questions you might have about all the elements and entities in DocBook. In particular, we cover the following subjects:

- The general nature of DocBook. With over 300 elements, DocBook can be a bit overwhelming at first. We quickly get you up to speed on how the pieces fit together.
- How to write DocBook documents. Where should you start and what should you do?
- Parsing and validation. After you've written a document, how can you tell if it really conforms to the DocBook DTD?
- How to publish DocBook documents. After you've written one, what do you do with it? We provide a guide to using some popular free tools to publish DocBook documents both in print and on the Web.
- Customizing the DTD. Many individuals and corporations have standardized on the DocBook DTD. Whether your subject matter is computer software documentation or not, we explain how you can write a "customization layer" to tailor DocBook explicitly for your information.
- Understanding all of the elements. Each element is extensively documented, including the intended semantics and the purpose of all its attributes. An example of proper usage is given for every element. The parameter entities and character entities are also described.
- Stylesheets. Several standard stylesheet languages are briefly described.
- XML compatability. We outline all of the points that you'll need to consider as you or your organization contemplate XML for authoring, publishing, or both.
- Additional resources and a CD-ROM. Finally, we direct you to other places you can go for all the latest info, and offer a complete set of online documentation on the CD-ROM.

We expect that most readers will have some familiarity with SGML or XML. Even if your experience goes no farther than writing a few HTML pages, you're probably in good shape. Although we provide an introduction to SGML, XML, and structured markup, this book may not suffice as your only tutorial about SGML and XML. This depends, naturally, on your needs and experience. For a list of some other good resources, consult Appendix D.

Some sections of this book describe tools and applications. For the most part, these are Microsoft Windows or UNIX applications, although there's nothing about DocBook that makes it unsuitable for the Mac or VM/CMS or any other operating system of your choice.

This book is divided into three parts. *Part I: Introduction* is an introduction to structured markup and DocBook:

Chapter 1, *Getting Startedwith SGML/XML* A quick introduction to structured markup.

Chapter 2, *Creating DocBook Documents* How to make DocBook documents.

Chapter 3, *Parsing DocBook Documents* Parsing and validating DocBook documents.

Chapter 4, *Publishing DocBook Documents* How to publish DocBook documents.

Chapter 5, *Customizing DocBook* How to customize DocBook.

Part II: Reference
is a complete reference to every element in the DocBook V4.1.2 DTD and provides a concise summary of the parameter entities. For a detailed reference to the parameter entities, consult the online version available either on the CD-ROM or the web site `http://docbook.org/`.

Reference 6 A reference guide to the DocBook elements.

Reference 7 A reference guide to the DocBook parameter entities.

Reference 8 A reference guide to the character entities declared in DocBook.

Part III: Appendixes discusses other resources:

Appendix A How to install DocBook, Jade, and the stylesheets.

Appendix B DocBook as XML.

Appendix C A guide to DocBook versions, including a summary of the features expected in future releases.

Appendix D Other resources.

Appendix E What's on the CD?

Appendix F An interchange checklist. Things to consider when you're sharing DocBook documents with others.

Appendix G A Quick Reference to the elements in DocBook.

At the end of this book you'll find a Glossary and an
Index.

- `Garamond Book` is used for element and attribute names.
- `Constant Willison` is used for program examples, attribute value literals, start- and end-tags, and source code example text.
- *`Constant Willison Oblique`* is used for "replaceable" text or variables. Replaceable text is text that describes something you're supposed to type, like a *`filename`*, in which the word "filename" is a placeholder for the actual filename.
- `Garamond Italic` is used for filenames and (in the print version of the book) URLs.
- URLs `http://docbook.org/` are presented in parentheses after the name of the resource they describe in the print version of the book.

If you want to hold this book in your hand and flip through its pages, you have to buy it as you would any other book. You can also get this book in electronic form, as a DocBook SGML document, and in HTML, either on the CD that accompanies the bound book or from this book's web site: `http://docbook.org/`.

All of the examples are included on the CD-ROM and online at the book's web site. You can get the most up-to-date information about this book from the web site: `http://docbook.org/`.

The DocBook DTD is included on the CD-ROM. You can get the most up-to-date version and information about DocBook from the DocBook web page: `http://www.oasis-open.org/docbook/`.

Please help us improve future editions of this book by reporting any errors, inaccuracies, bugs, misleading or confusing statements, and plain old typos that you find. An online errata list is maintained at `http://docbook.org/tdg/errata.html`. Email your bug reports and comments to us at <`mailto:bookcomments@docbook.org`>.

This book has been in the works for a long time. It could not have been completed without the help and encouragement of a lot of people, most especially my wife, Deborah, who supported me through the long hours and the late nights.

I also want to thank Lenny for collaborating with me and developing real prose out of my rough outlines, cryptic email messages, and scribbled notes.

A number of people contributed technical feedback as this book was being written, in particular Terry Allen and Eve Maler. I owe most of what I know about SGML to them, and to the other members of the Davenport Group who answered all my questions so many years ago, especially Jon Bosak, Eduardo Guttentag, and Murray Maloney. Paul Prescod, Mark Galassi, and Dave Pawson also provided invaluable feedback on the technical review draft. It's a better book because of them.

My gratitude goes back to Dale Dougherty and Terry Allen, who long ago encouraged me and the production department at O'Reilly to learn SGML; and to Lar Kaufman, who also made large contributions to my knowledge and appreciation of SGML. But my greatest debt of thanks goes to Norm for all that he patiently taught me about DocBook, and for his supreme graciousness in keeping me a part of this project.

Thanks finally to the great people at O'Reilly who encouraged us to write it (Frank Willison and Sheryl Avruch), agreed to edit it (Frank), helped design it (Alicia Cech, who worked on the interior design, and Edie Freeman, who designed the cover), proofed and produced it (Chris Maden, Madeline Newell, and David Futato), and indexed it (Ellen Troutman).

Part I

Introduction

Chapter 1

Getting Started with SGML/XML

This chapter is intended to provide a quick introduction to structured markup (SGML and XML). If you're already familiar with SGML or XML, you only need to skim this chapter.

To work with DocBook, you need to understand a few basic concepts of structured editing in general, and DocBook, in particular. That's covered here. You also need some concrete experience with the way a DocBook document is structured. That's covered in the next chapter.

1.1 HTML and SGML vs. XML

This chapter doesn't assume that you know what HTML is, but if you do, you have a starting point for understanding structured markup. HTML (Hypertext Markup Language) is a way of marking up text and graphics so that the most popular web browsers can interpret them. HTML consists of a set of markup tags with specific meanings. Moreover, HTML is a very basic type of SGML markup that is easy to learn and easy for computer applications to generate. But the simplicity of HTML is both its virtue and its weakness. Because of HTML's limitations, web users and programmers have had to extend and enhance it by a series of customizations and revisions that still fall short of accommodating current, to say nothing of future, needs.

SGML, on the other hand, is an international standard that describes how markup languages are defined. SGML does not consist of particular tags or the rules for their usage. HTML is an example of a markup language defined in SGML.

XML promises an intelligent improvement over HTML, and compatibility with it is already being built into the most popular web browsers. XML is not a new markup language designed to compete with HTML, and it's not designed to create conversion headaches for people with tons of HTML documents. XML is intended to alleviate compatibility problems with browser software; it's a new, easier version of the standard rules that govern the markup itself, or, in other words, a new version of SGML. The rules of XML are designed to make it easier to write both applications that interpret its type of markup and applications that generate its markup. XML was developed by a team of SGML experts who understood and sought to correct the problems of learning and implementing SGML. XML is also *extensible* markup, which means that it is customizable. A browser or word processor that is XML-capable will be able to read any XML-based markup language that an individual user defines.

In this book, we tend to describe things in terms of SGML, but where there are differences between SGML and XML (and there are only a few), we point them out. For our purposes, it doesn't really matter whether you use SGML or XML.

During the coming months, we anticipate that XML-aware web browsers and other tools will become available. Nevertheless, it's not unreasonable to do your authoring in SGML and your online publishing in XML or HTML. By the same token, it's not unreasonable to do your authoring in XML.

1.2 Basic SGML/XML Concepts

Here are the basic SGML/XML concepts you need to grasp:

- structured, semantic markup
- elements

- attributes
- entities

1.2.1 Structured and Semantic Markup

An essential characteristic of structured markup is that it explicitly distinguishes (and accordingly "marks up" within a document) the structure and semantic content of a document. It does not mark up the way in which the document will appear to the reader, in print or otherwise.

In the days before word processors it was common for a typed manuscript to be submitted to a publisher. The manuscript identified the logical structures of the documents (chapters, section titles, and so on), but said nothing about its appearance. Working independently of the author, a designer then developed a specification for the appearance of the document, and a typesetter marked up and applied the designer's format to the document.

Because presentation or appearance is usually based on structure and content, SGML markup logically precedes and generally determines the way a document will look to a reader. If you are familiar with strict, simple HTML markup, you know that a given document that is structurally the same can also look different on different computers. That's because the markup does not specify many aspects of a document's appearance, although it does specify many aspects of a document's structure.

Many writers type their text into a word processor, line-by-line and word-for-word, italicizing technical terms, underlining words for emphasis, or setting section headers in a font complementary to the body text, and finally, setting the headers off with a few carriage returns fore and aft. The format such a writer imposes on the words on the screen imparts structure to the document by changing its appearance in ways that a reader can more or less reliably decode. The reliability depends on how consistently and unambiguously the changes in type and layout are made. By contrast, an SGML/XML markup of a section header explicitly specifies that a specific piece of text is a section header. This assertion does not specify the presentation or appearance of the section header, but it makes the fact that the text is a section header completely unambiguous.

SGML and XML use named elements, delimited by angle brackets ("<" and ">") to identify the markup in a document. In DocBook, a top-level section is `<sect1>`, so the title of a top-level section named *My First-Level Header* would be identified like this:

```
<sect1><title>My First-Level Header</title>
```

Note the following features of this markup:

Clarity A title begins with `<title>` and ends with `</title>`. The `sect1` also has an ending `</sect1>`, but we haven't shown the whole section so it's not visible.

Hierarchy "My First-Level Header" is the title of a top-level section because it occurs inside a title in a `sect1`. A `title` element occurring somewhere else, say in a `Chapter` element, would be the title of the chapter.

Plain text SGML documents can have varying character sets, but most are ASCII. XML documents use the Unicode character set. This makes SGML and XML documents highly portable across systems and tools.

In an SGML document, there is no obligatory difference between the size or face of the type in a first-level section header and the title of a book in a footnote or the first sentence of a body paragraph. All SGML files are simple text files without font changes or special characters.[1] Similarly, an SGML document does not specify the words in a text that are to be set in italic, bold, or roman type. Instead, SGML marks certain kinds of texts for their semantic content. For example, if a particular word is the name of a file, then the tags around it should specify that it is a filename:

```
Many mail programs read configuration information from the
users filename.mailrcfilename file.
```

If the meaning of a phrase is particularly audacious, it might get tagged for boldness of thought instead of appearance. An SGML document contains all the information that a typesetter needs to lay out and typeset a printed page in the most effective and consistent way, but it does not specify the layout or the type.[2]

[1] Some structured editors apply style to the document while it's being edited, using fonts and color to make the editing task easier, but this stylistic information is not stored in the actual SGML/XML document. Instead, it is provided by the editing application.

[2] The distinction between appearance or presentation and structure or content is essential to SGML, but there is a way to specify the appearance of an SGML document: attach a stylesheet to it. There are several standards for such stylesheets: CSS, XSL, FOSIs, and DSSSL. See Chapter 4, *Publishing DocBook Documents*.

Not only is the structure of an SGML/XML document explicit, but it is also carefully controlled. An SGML document makes reference to a set of declarations—a document type definition (DTD)—that contains an inventory of tag names and specifies the combination rules for the various structural and semantic features that make up a document. What the distinctive features are and how they should be combined is "arbitrary" in the sense that almost any selection of features and rules of composition is theoretically possible. The DocBook DTD chooses a particular set of features and rules for its users.

Here is a specific example of how the DocBook DTD works. DocBook specifies that a third-level section can follow a second-level section but cannot follow a first-level section without an intervening second-level section.

This is valid:	This is not:

`<sect1><title>...</title>`
`<sect2><title>...</title>`
`<sect3><title>...</title> ...`
`</sect3> </sect2> </sect1>`
	`<sect1><title>...</title>`
	`<sect3><title>...</title> ...`
	`</sect3> </sect1>`

Because an SGML/XML document has an associated DTD that describes the valid, logical structures of the document, you can test the logical structure of any particular document against the DTD. This process is performed by a *parser*. An SGML processor must begin by parsing the document and determining if it is valid, that is, if it conforms to the rules specified in the DTD. XML processors are not required to check for validity, but it's always a good idea to check for validity when authoring. Because you can test and validate the structure of an SGML/XML document with software, a DocBook document containing a first-level section followed immediately by a third-level section will be identified as invalid, meaning that it's not a valid *instance* or example of a document defined by the DocBook DTD. Presumably, a document with a logical structure won't normally jump from a first- to a third-level section, so the rule is a safeguard—but not a guarantee—of good writing, or at the very least, reasonable structure. A parser also verifies that the names of the tags are correct and that tags requiring an ending tag have them. This means that a valid document is also one that should format correctly, without runs of paragraphs incorrectly appearing in bold type or similar monstrosities that everyone has seen in print at one time or another. For more information about SGML/XML parsers, see Chapter 3, *Parsing DocBook Documents*.

In general, adherence to the explicit rules of structure and markup in a DTD is a useful and reassuring guarantee of consistency and reliability within documents, across document sets, and over time. This makes SGML/XML markup particularly desirable to corporations or governments that have large sets of documents to manage, but it is a boon to the individual writer as well.

1.2.1.1 How can this markup help you?

Semantic markup makes your documents more amenable to interpretation by software, especially publishing software. You can publish a white paper, authored as a DocBook `Article`, in the following formats:

- On the Web in HTML
- As a standalone document on $8\frac{1}{2}\times 11$ paper
- As part of a quarterly journal, in a 6×9 format
- In Braille
- In audio

You can produce each of these publications from exactly the same source document using the presentational techniques best suited to both the content of the document and the presentation medium. This versatility also frees the author to concentrate on the document content. For example, as we write this book, we don't know exactly how O'Reilly will choose to present chapter headings, bulleted lists, SGML terms, or any of the other semantic features. And we don't care. It's irrelevant; whatever presentation is chosen, the SGML sources will be transformed automatically into that style.

Semantic markup can relieve the author of other, more significant burdens as well (after all, careful use of paragraph and character styles in a word processor document theoretically allows us to change the presentation independently from the document). Using semantic markup opens up your documents to a world of possibilities. Documents become, in a loose sense, databases of information. Programs can compile, retrieve, and otherwise manipulate the documents in predictable, useful ways.

Consider the online version of this book: almost every element name (`Article`, `Book`, and so on) is a hyperlink to the reference page that describes that element. Maintaining these links by hand would be tedious and might be unreliable, as well. Instead, every element name is marked as an element using `SGMLTag`: a `Book` is a `<sgmltag>Book</sgmltag>`.

Because each element name in this book is tagged semantically, the program that produces the online version can determine which occurrences of the word "book" in the text are actually references to the `Book` element. The program can then automatically generate the appropriate hyperlink when it should.

There's one last point to make about the versatility of SGML documents: how much you have depends on the DTD. If you take a good photo with a high resolution lens, you can print it and copy it and scan it and put it on the Web, and it will look good. If you start with a low-resolution picture it will not survive those transformations so well. DocBook SGML/XML has this advantage over, say, HTML: DocBook has specific and unambiguous semantic and structural markup, because you can convert its documents with ease into other presentational forms, and search them more precisely. If you start with HTML, whose markup is at a lower resolution than DocBook's, your versatility and searchability is substantially restricted and cannot be improved.

1.2.1.2 What are the shortcomings to structural authoring?

There are a few significant shortcomings to structured authoring:

- It requires a significant change in the authoring process. Writing structured documents is very different from writing with a typical word processor, and change is difficult. In particular, authors don't like giving up control over the appearance of their words especially now that they have acquired it with the advent of word processors. But many publishing companies need authors to relinquish that control, because book design and production remains their job, not their authors'.
- Because semantics are separate from appearance, in order to publish an SGML/XML document, a stylesheet or other tool must create the presentational form from the structural form. Writing stylesheets is a skill in its own right, and though not every author among a group of authors has to learn how to write them, someone has to.
- Authoring tools for SGML documents can generally be pretty expensive. While it's not entirely unreasonable to edit SGML/XML documents with a simple text editor, it's a bit tedious to do so. However, there are a few free tools that are SGML-aware. The widespread interest in XML may well produce new, clever, and less expensive XML editing tools.

1.3 Elements and Attributes

SGML/XML markup consists primarily of *elements*, *attributes*, and *entities*. Elements are the terms we have been speaking about most, like `sect1`, that describe a document's content and structure. Most elements are represented by pairs of tags and mark the start and end of the construct they surround—for example, the SGML source for this particular paragraph begins with a `<para>` tag and ends with a `</para>` tag. Some elements are "empty" (such as DocBook's cross-reference element, `<xref>`) and require no end tag.[3]

Elements can, but don't necessarily, include one or more attributes, which are additional terms that extend the function or refine the content of a given element. For instance, in DocBook a `<sect1>` start tag can contain an identifier—an id attribute—that will ultimately allow the writer to cross-reference it or enable a reader to retrieve it. End tags cannot contain attributes. A `<sect1>` element with an id attribute looks like this:

```
<sect1 id="idvalue">
```

In SGML, the catalog of attributes that can occur on an element is predefined. You cannot add arbitrary attribute names to an element. Similarly, the values allowed for each attribute are predefined. In XML, the use of namespaces `http://www.w3.org/TR/REC-xml-names/` may allow you to add additional attributes to an element, but as of this writing, there's no way to perform validation on those attributes.

The id attribute is one half of a cross reference. An idref attribute on another element, for example `<xref linkend="idvalue">`, provides the other half. These attributes provide whatever application might process the SGML source with the data needed either to make a hypertext link or to substitute a named and/or numbered cross reference in place of the `<xref>`. Another use for attributes is to specify subclasses of certain elements. For instance, you can subdivide DocBook's `<systemitem>` into URLs and email addresses by making the content of the role attribute the distinction between them, as in `<systemitem role="URL">` versus `<systemitem role="emailaddr">`.

[3] In XML, this is written as `<xref/>`, as we'll see in the section Section 2.1.5.

1.4 Entities

Entities are a fundamental concept in SGML and XML, and can be somewhat daunting at first. They serve a number of related, but slightly different functions, and this makes them a little bit complicated.

In the most general terms, entities allow you to assign a name to some chunk of data, and use that name to refer to that data. The complexity arises because there are two different contexts in which you can use entities (in the DTD and in your documents), two types of entities (parsed and unparsed), and two or three different ways in which the entities can point to the chunk of data that they name.

In the rest of this section, we'll describe each of the commonly encountered entity types. If you find the material in this section confusing, feel free to skip over it now and come back to it later. We'll refer to the different types of entities as the need arises in our discussion of DocBook. Come back to this section when you're looking for more detail.

Entities can be divided into two broad categories, *general entities* and *parameter entities*. Parameter entities are most often used in the DTD, not in documents, so we'll describe them last. Before you can use any type of entity, it must be formally declared. This is typically done in the document prologue, as we'll explain in Chapter 2, *Creating DocBook Documents*, but we will show you how to declare each of the entities discussed here.

1.4.1 General Entities

In use, general entities are introduced with an ampersand (&) and end with a semicolon (;). Within the category of general entities, there are two types: *internal general entities* and *external general entities*.

1.4.1.1 Internal general entities

With internal entities, you can associate an essentially arbitrary piece of text (which may have other markup, including references to other entities) with a name. You can then include that text by referring to its name. For example, if your document frequently refers to, say, "O'Reilly & Associates," you might declare it as an entity:

```
<!ENTITY ora "O'Reilly &amp; Associates">
```

Then, instead of typing it out each time, you can insert it as needed in your document with the entity reference `&ora;`, simply to save time. Note that this entity declaration includes another entity reference within it. That's perfectly valid as long as the reference isn't directly or indirectly recursive.

If you find that you use a number of entities across many documents, you can add them directly to the DTD and avoid having to include the declarations in each document. See the discussion of `dbgenent.mod` in Chapter 5, *Customizing DocBook*.

1.4.1.2 External general entities

With external entities, you can reference other documents from within your document. If these entities contain document text (SGML or XML), then references to them cause the parser to insert the text of the external file directly into your document (these are called parsed entities). In this way, you can use entities to divide your single, logical document into physically distinct chunks. For example, you might break your document into four chapters and store them in separate files. At the top of your document, you would include entity declarations to reference the four files:

```
<!ENTITY ch01 SYSTEM "ch01.sgm">
<!ENTITY ch02 SYSTEM "ch02.sgm">
<!ENTITY ch03 SYSTEM "ch03.sgm">
<!ENTITY ch04 SYSTEM "ch04.sgm">
```

Your `Book` now consists simply of references to the entities:

```
<book>
&ch01;
&ch02;
&ch03;
&ch04;
</book>
```

Sometimes it's useful to reference external files that don't contain document text. For example, you might want to reference an external graphic. You can do this with entities by declaring the type of data that's in the entity using a notation (these are called unparsed entities). For example, the following declaration declares the entity `tree` as an encapsulated PostScript image:

```
<!ENTITY tree SYSTEM "tree.eps" NDATA EPS>
```

Entities declared this way cannot be inserted directly into your document. Instead, they must be used as entity attributes to elements:

```
<graphic entityref="tree"></graphic>
```

Conversely, you cannot use entities declared without a notation as the value of an entity attribute.

1.4.1.3 Special characters

In order for the parser to recognize markup in your document, it must be able to distinguish markup from content. It does this with two special characters: "<," which identifies the beginning of a start or end tag, and "&," which identifies the beginning of an entity reference.[4] If you want these characters to have their literal value, they must be encoded as entity references in your document. The entity reference `<` produces a left angle bracket; `&` produces the ampersand.[5]

If you do not encode each of these as their respective entity references, then an SGML parser or application is likely to interpret them as characters introducing elements or entities (an XML parser will always interpret them this way); consequently, they won't appear as you intended. If you wish to cite text that contains literal ampersands and less-than signs, you need to transform these two characters into entity references before they are included in a DocBook document. The only other alternative is to incorporate text that includes them in your document through some process that avoids the parser.

In SGML, character entities are frequently declared using a third entity category (one that we deliberately chose to overlook), called *data entities*. In XML, these are declared using numeric character references. Numeric character references resemble entity references, but technically aren't the same. They have the form `ϧ`, in which "999" is the numeric character number.

In XML, the numeric character number is always the Unicode character number. In addition, XML allows hexadecimal numeric character references of the form `&#xhhhh;`. In SGML, the numeric character number is a number from the document character set that's declared in the SGML declaration.

Character entities are also used to give a name to special characters that can't otherwise be typed or are not portable across applications and operating systems. You can then include these characters in your document by refering to their entity name. Instead of using the often obscure and inconsistent key combinations of your particular word processor to type, say, an uppercase letter U with an umlaut (Ü), you type in an entity for it instead. For instance, the entity for an uppercase letter U with an umlaut has been defined as the entity `Uuml`, so you would type in `Ü` to reference it instead of the actual character. The SGML application that eventually processes your document for presentation will match the entity to your platform's handling of special characters in order to render it appropriately.

1.4.2 Parameter Entities

Parameter entities are only recognized in markup declarations (in the DTD, for example). Instead of beginning with an ampersand, they begin with a percent sign. Parameter entities are most frequently used to customize the DTD. For a detailed discussion of this topic, see Chapter 5, *Customizing DocBook*. Following are some other uses for them.

1.4.2.1 Marked sections

You might use a parameter entity reference in an SGML document in a marked section. Marking sections is a mechanism for indicating that special processing should apply to a particular block of text. Marked sections are introduced by the special sequence `<![keyword[` and end with `]]>`. In SGML, marked sections can appear in both DTDs and document instances. In XML, they're only allowed in the DTD.[6]

The most common keywords are `INCLUDE`, which indicates that the text in the marked section should be included in the document; `IGNORE`, which indicates that the text in the marked section should be ignored (it completely disappears from the parsed document); and `CDATA`, which indicates that all markup characters within that section should be ignored except for the closing characters `]]>`.

[4] In XML, these characters are fixed. In SGML, it is possible to change the markup start characters, but we won't consider that case here. If you change the markup start characters, you know what you're doing. While we're on the subject, in SGML, these characters only have their special meaning if they are followed by a name character. It is, in fact, valid in an *SGML* (but not an XML) document to write "O'Reilly & Associates" because the ampersand is not followed by a name character. Don't do this, however.

[5] The sequence of characters that end a marked section (see Section 1.4.2.1), such as]]> must also be encoded with at least one entity reference if it is not being used to end a marked section. For this purpose, you can use the entity reference `>` for the final right angle bracket.

[6] Actually, CDATA marked sections are allowed in an XML document, but the keyword cannot be a parameter entity, and it must be typed literally. See the examples on this page.

In SGML, these keywords can be parameter entities. For example, you might declare the following parameter entity in your document:

```
<!ENTITY % draft "INCLUDE">
```

Then you could put the sections of the document that are only applicable in a draft within marked sections:

```
<![%draft;[
<para>
This paragraph only appears in the draft version.
</para>
]]>
```

When you're ready to print the final version, simply change the `draft` parameter entity declaration:

```
<!ENTITY % draft "IGNORE">
```

and publish the document. None of the draft sections will appear.

1.5 How Does DocBook Fit In?

DocBook is a very popular set of tags for describing books, articles, and other prose documents, particularly technical documentation. DocBook is defined using the native DTD syntax of SGML and XML. Like HTML, DocBook is an example of a markup language defined in SGML/XML.

1.5.1 A Short DocBook History

DocBook is almost 10 years old. It began in 1991 as a joint project of HaL Computer Systems and O'Reilly. Its popularity grew, and eventually it spawned its own maintenance organization, the Davenport Group. In mid-1998, it became a Technical Committee (TC) of the Organization for the Advancement of Structured Information Standards (OASIS).

1.5.1.1 The HaL and O'Reilly era

The DocBook DTD was originally designed and implemented by HaL Computer Systems and O'Reilly & Associates around 1991. It was developed primarily to facilitate the exchange of UNIX documentation originally marked up in **troff**. Its design appears to have been based partly on input from SGML interchange projects conducted by the Unix International and Open Software Foundation consortia.

When DocBook V1.1 was published, discussion about its revision and maintenance began in earnest in the Davenport Group, a forum created by O'Reilly for computer documentation producers. Version 1.2 was influenced strongly by Novell and Digital.

In 1994, the Davenport Group became an officially chartered entity responsible for DocBook's maintenance. DocBook V1.2.2 was published simultaneously. The founding sponsors of this incarnation of Davenport include the following people:

- Jon Bosak, Novell
- Dale Dougherty, O'Reilly & Associates
- Ralph Ferris, Fujitsu OSSI
- Dave Hollander, Hewlett-Packard
- Eve Maler, Digital Equipment Corporation
- Murray Maloney, SCO
- Conleth O'Connell, HaL Computer Systems
- Nancy Paisner, Hitachi Computer Products
- Mike Rogers, SunSoft
- Jean Tappan, Unisys

1.5.1.2 The Davenport era

Under the auspices of the Davenport Group, the DocBook DTD began to widen its scope. It was now being used by a much wider audience, and for new purposes, such as direct authoring with SGML-aware tools, and publishing directly to paper. As the largest users of DocBook, Novell and Sun had a heavy influence on its design.

In order to help users manage change, the new Davenport charter established the following rules for DocBook releases:

- Minor versions ("point releases" such as V2.2) could add to the markup model, but could not change it in a backward-incompatible way. For example, a new kind of list element could be added, but it would not be acceptable for the existing itemized-list model to start requiring two list items inside it instead of only one. Thus, any document conforming to version *n*.0 would also conform to *n.m*.
- Major versions (such as V3.0) could both add to the markup model and make backward-incompatible changes. However, the changes would have to be announced in the last major release.
- Major-version introductions must be separated by at least a year.

V3.0 was released in January 1997. After that time, although DocBook's audience continued to grow, many of the Davenport Group stalwarts became involved in the XML effort, and development slowed dramatically. The idea of creating an official XML-compliant version of DocBook was discussed, but not implemented. (For more detailed information about DocBook V3.0 and plans for subsequent versions, see Appendix C.)

The sponsors wanted to close out Davenport in an orderly way to ensure that DocBook users would be supported. It was suggested that OASIS become DocBook's new home. An OASIS DocBook Technical Committee was formed in July, 1998, with Eduardo Gutentag of Sun Microsystems as chair.

1.5.1.3 The OASIS era

The DocBook Technical Commitee `http://www.oasis-open.org/docbook/` is continuing the work started by the Davenport Group. The transition from Davenport to OASIS has been very smooth, in part because the core design team consists of essentially the same individuals (we all just changed hats).

DocBook V3.1, published in February 1999, was the first OASIS release. It integrated a number of changes that had been "in the wings" for some time.

In February of 2001, OASIS made DocBook SGML V4.1 and DocBook XML V4.1.2 official OASIS Specifications `http://lists.oasis-open.org/archives/members/200102/msg00000.html`.

Version 4.2 `http://www.oasis-open.org/docbook/specs/cs-docbook-docbook-4.2.html` of the DocBook DTD, for both SGML and XML, was released in July 2002.

The committee continues new DocBook development to ensure that the DTD continues to meet the needs of its users. Forthcoming and experimental work includes:

- A V5.0 DTD projected for release no earlier than the end of 2002.
- Experimental RELAX NG `http://www.oasis-open.org/committees/relax-ng/` schemas available `http://www.oasis-open.org/docbook/relaxng`.
- Experimental W3C XML Schema `http://www.w3.org/XML/Schema` versions available `http://www.oasis-open.org/docbook/xmlschema/`.
- Experimental RELAX `http://www.xml.gr.jp/relax/` schemas available `http://www.oasis-open.org/docbook/relax/`.
- Experimental TREX `http://www.thaiopensource.com/trex/` schemas available `http://www.oasis-open.org/docbook/trex/`.

Chapter 2

Creating DocBook Documents

This chapter explains in concrete, practical terms how to make DocBook documents. It's an overview of all the kinds of markup that are possible in DocBook documents. It explains how to create several kinds of DocBook documents: books, sets of books, chapters, articles, and reference manual entries. The idea is to give you enough basic information to actually start writing. The information here is intentionally skeletal; you can find "the details" in the reference section of this book.

Before we can examine DocBook markup, we have to take a look at what an SGML or XML system requires.

2.1 Making an SGML Document

SGML requires that your document have a specific prologue. The following sections describe the features of the prologue.

2.1.1 An SGML Declaration

SGML documents begin with an optional SGML Declaration. The declaration can precede the document instance, but generally it is stored in a separate file that is associated with the DTD. The SGML Declaration is a grab bag of SGML defaults. DocBook includes an SGML Declaration that is appropriate for most DocBook documents, so we won't go into a lot of detail here about the SGML Declaration.

In brief, the SGML Declaration describes, among other things, what characters are markup delimiters (the default is angle brackets), what characters can compose tag and attribute names (usually the alphabetical and numeric characters plus the dash and the period), what characters can legally occur within your document, how long SGML "names" and "numbers" can be, what sort of minimizations (abbreviation of markup) are allowed, and so on. Changing the SGML Declaration is rarely necessary, and because many tools only partially support changes to the declaration, changing it is best avoided, if possible.

Wayne Wholer has written an excellent tutorial on the SGML Declaration; if you're interested in more details, see `http://www.oasis-open.org/cover/wlw11.html`.

2.1.2 A Document Type Declaration

All SGML documents must begin with a document type declaration. This identifies the DTD that will be used by the document and what the root element of the document will be. A typical doctype declaration for a DocBook document looks like this:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
```

This declaration indicates that the *root element*, which is the first element in the hierarchical structure of the document, will be `<book>` and that the DTD used will be the one identified by the public identifier `-//OASIS//DTD DocBook V3.1//EN`. See Section 2.3.1" later in this chapter.

2.1.3 An Internal Subset

It's also possible to provide additional declarations in a document by placing them in the document type declaration:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY nwalsh "Norman Walsh">
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap2 SYSTEM "chap2.sgm">
]>
```

These declarations form what is known as the *internal subset*. The declarations stored in the file referenced by the public or system identifier in the DOCTYPE declaration is called the *external subset* and it is technically optional. It is legal to put the DTD in the internal subset and to have no external subset, but for a DTD as large as DocBook that wouldn't make much sense.

NOTE

The internal subset is parsed *first* and, if multiple declarations for an entity occur, the first declaration is used. Declarations in the internal subset override declarations in the external subset.

2.1.4 The Document (or Root) Element

Although comments and processing instructions may occur between the document type declaration and the root element, the root element usually immediately follows the document type declaration:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY nwalsh "Norman Walsh">
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap2 SYSTEM "chap2.sgm">
]>
<book>
&chap1;
&chap2;
</book>
```

You cannot place the root element of the document in an external entity.

2.1.5 Typing an SGML Document

If you are entering SGML using a text editor such as Emacs or vi, there are a few things to keep in mind.[1] Using a structured text editor designed for SGML hides most of these issues.

- DocBook element and attribute names are not case-sensitive. There's no difference between <Para> and <pArA>. Entity names are case-sensitive, however.

  If you are interested in future XML compatibility, input all element and attribute names strictly in lowercase.

- If attribute values contain spaces or punctuation characters, you must quote them. You are not required to quote attribute values if they consist of a single word or number, although it is not wrong to do so.

  When quoting attribute values, you can use either a straight single quote (’), or a straight double quote ("). Don't use the "curly" quotes (“ and ”) in your editing tool.

  If you are interested in future XML compatibility, always quote all attribute values.

- Several forms of markup minimization are allowed, including empty tags. Instead of typing the entire end tag for an element, you can type simply </>. For example:

[1] Many of these things are influenced by the SGML declaration in use. For the purpose of this discussion, we assume you are using the standard DocBook declaration.

```
<para>
This is <emphasis>important</>: never stick the tines of a fork
in an electrical outlet.
</para>
```

You can use this technique for any and every tag, but it will make your documents very hard to understand and difficult to debug if you introduce errors. It is best to use this technique only for inline elements containing a short string of text.

Empty start tags are also possible, but may be even more confusing. For the record, if you encounter an empty start tag, the SGML parser uses the element that ended last:

```
<para>
This is <emphasis>important</emphasis>.  So is <>this</emphasis>.
</para>
```

Both “important” and “this” are emphasized.

If you are interested in future XML compatibility, don’t use any of these tricks.

- The null end tag (net) minimization feature allows constructions like this:

```
<para>
This is <emphasis/important/: never stick the tines of a fork
in an electrical outlet.
</para>
```

If, instead of ending a start tag with >, you end it with a slash, then the next occurrence of a slash ends the element.

If you are interested in future XML compatibility, don’t use net tag minimization either.

If you are willing to modify both the declaration and the DTD, even more dramatic minimizations are possible, including completely omitted tags and “shortcut” markup.

REMOVING MINIMIZATIONS

Although we’ve made a point of reminding you about which of these minimization features are not valid in XML, that’s not really a sufficient reason to avoid using them. (The fact that many of the minimization features can lead to confusing, difficult-to-author documents might be.)

If you want to convert one of these documents to XML at some point in the future, you can run it through a program like **sgmlnorm**, which will remove all the minimizations and insert the correct, verbose markup. The **sgmlnorm** program is part of the SP and Jade distributions `http://www.jclark.com/`, which are on the CD-ROM.

2.2 Making an XML Document

In order to create DocBook documents in XML, you’ll need an XML version of DocBook. We’ve included one on the CD, but it hasn’t been officially adopted by the OASIS DocBook Technical Committee yet. If you’re interested

in the technical details, Appendix B, describes the specific differences between SGML and XML versions of DocBook.

XML, like SGML, requires a specific prologue in your document. The following sections describe the features of the XML prologue.

2.2.1 An XML Declaration

XML documents should begin with an XML declaration. Unlike the SGML declaration, which is a grab bag of features, the XML declaration identifies a few simple aspects of the document:

```
<?xml version="1.0" standalone="no"?>
```

Identifying the version of XML ensures that future changes to the XML specification will not alter the semantics of this document. The standalone declaration simply makes explicit the fact that this document cannot "stand alone," and that it relies on an external DTD. The complete details of the XML declaration are described in the XML specification http://www.w3.org/TR/REC-xml.

2.2.2 A Document Type Declaration

Strictly speaking, XML documents don't require a DTD. Realistically, DocBook XML documents will have one.

The document type declaration identifies the DTD that will be used by the document and what the root element of the document will be. A typical doctype declaration for a DocBook document looks like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//Norman Walsh//DTD DocBk XML V3.1.4//EN"
                      "http://nwalsh.com/docbook/xml/3.1.4/db3xml.dtd">
```

This declaration indicates that the root element will be <book> and that the DTD used will be the one indentified by the public identifier -//Norman Walsh//DTD DocBk XML V3.1.4//EN. External declarations in XML must include a system identifier (the public identifier is optional). In this example, the DTD is stored on a web server.

System identifiers in XML must be URIs. Many systems may accept filenames and interpret them locally as file: URLs, but it's always correct to fully qualify them.

2.2.3 An Internal Subset

It's also possible to provide additional declarations in a document by placing them in the document type declaration:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//Norman Walsh//DTD DocBk XML V3.1.4/EN"
                      "http://nwalsh.com/docbook/xml/3.1.4/db3xml.dtd" [
<!ENTITY nwalsh "Norman Walsh">
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap2 SYSTEM "chap2.sgm">
]>
```

These declarations form what is known as the internal subset. The declarations stored in the file referenced by the public or system identifier in the DOCTYPE declaration is called the external subset, which is technically optional. It is legal to put the DTD in the internal subset and to have no external subset, but for a DTD as large as DocBook, that would make very little sense.

NOTE

The internal subset is parsed *first* in XML and, if multiple declarations for an entity occur, the first declaration is used. Declarations in the internal subset override declarations in the external subset.

2.2.4 The Document (or Root) Element

Although comments and processing instructions may occur between the document type declaration and the root element, the root element usually immediately follows the document type declaration:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//Norman Walsh//DTD DocBk XML V3.1.4//EN"
                        "http://nwalsh.com/docbook/xml/3.1.4/db3xml.dtd" [
<!ENTITY nwalsh "Norman Walsh">
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap2 SYSTEM "chap2.sgm">
]>
<book>...</book>
```

The important point is that the root element must be physically present immediately after the document type declaration. You cannot place the root element of the document in an external entity.

2.2.5 Typing an XML Document

If you are entering SGML using a text editor such as Emacs or vi, there are a few things to keep in mind. Using a structured text editor designed for XML hides most of these issues.

- In XML, all markup is case-sensitive. In the XML version of DocBook, you must always type all element, attribute, and entity names in lowercase.
- You are required to quote all attribute values in XML.

  When quoting attribute values, you can use either a straight single quote (’), or a straight double quote (").
  Don’t use the “curly” quotes (“ and ”) in your editing tool.
- Empty elements in XML are marked with a distinctive syntax: `<xref/>`.
- Processing instructions in XML begin and end with a question mark: `<?pitarget data?>`.
- XML was designed to be served, received, and processed over the Web. Two of its most important design principles are ease of implementation and interoperability with both SGML and HTML.

  The markup minimization features in SGML documents make it more difficult to process, and harder to write a parser to interpret it; these minimization features also run counter to the XML design principles named above. As a result, XML does not support them.

  Luckily, a good authoring environment can offer all of the features of markup minimization without interfering with the interoperability of documents. And because XML tools are easier to write, it’s likely that good, inexpensive XML authoring environments will be available eventually.

2.2.6 XML and SGML Markup Considerations in This Book

Conceptually, almost everything in this book applies equally to SGML and XML. But because DocBook V3.1 is an SGML DTD, we naturally tend to use SGML conventions in our writing. If you’re primarily interested in XML, there are just a few small details to keep in mind.

- XML is case-sensitive, while the SGML version of DocBook is not. In this book, we’ve chosen to present the element names using mixed case (`Book`, `indexterm`, `XRef`, and so on), but in the DocBook XML DTD, all element, attribute, and entity names are strictly lowercase.
- Empty element start tags in XML are marked with a distinctive syntax: `<xref/>`. In SGML, the trailing slash is not present, so some of our examples need slight revisions to be valid XML elements.
- Processing instructions in XML begin and end with a question mark: `<?pitarget data?>`. In SGML, the trailing question mark is not present, so some of our examples need slight revisions to be valid XML elements.
- Generally we use public identifiers in examples, but whenever system identifiers are used, don’t forget that XML system identifiers must be Uniform Resource Indicators (URIs), in which SGML system identifiers are usually simple filenames.

For a more detailed discussion of DocBook and XML, see Appendix B.

2.3 Public Identifiers, System Identifiers, and Catalog Files

When a DTD or other external file is referenced from a document, the reference can be specified in three ways: using a *public identifier*, a *system identifier*, or both. In XML, the system identifier is *generally* required and the public identifier is optional. In SGML, neither is required, but at least one must be present.[2]

A public identifier is a globally unique, abstract name, such as the following, which is the official public identifier for DocBook V3.1:

```
-//OASIS//DTD DocBook V3.1//EN
```

The introduction of XML has added some small complications to system identifiers. In SGML, a system identifier generally points to a single, local version of a file using local system conventions. In XML, it must point with a Uniform Resource Indicator (URI). The most common URI today is the Uniform Resource Locator (URL), which is familiar to anyone who browses the Web. URLs are a lot like SGML system identifiers, because they generally point to a single version of a file on a particular machine. In the future, Uniform Resource Names (URN), another form of URI, will allow XML system identifiers to have the abstract characteristics of public identifiers.

The following filename is an example of an SGML system identifier:

```
/usr/local/sgml/docbook/3.1/docbook.dtd
```

An equivalent XML system identifier might be:

```
file:///usr/local/sgml/docbook/3.1/docbook.dtd
```

The advantage of using the public identifier is that it makes your documents more portable. For any system on which DocBook is installed, the public identifier will resolve to the appropriate local version of the DTD (if public identifiers can be resolved at all).

Public identifiers have two disadvantages:

- Because XML does not require them, and because system identifiers are required, developing XML tools may not provide adequate support for public identifiers. To work with these systems you must use system identifiers.
- Public identifiers aren't magical. They're simply a method of indirection. For them to work, there must be a resolution mechanism for public identifiers. Luckily, several years ago, SGML Open (now OASIS http://www.oasis-open.org/) described a standard mechanism for mapping public identifiers to system identifers using catalog files.

  See OASIS Technical Resolution 9401:1997 (Amendment 2 to TR 9401). http://www.oasis-open.org/html/a401.htm

2.3.1 Public Identifiers

An important characteristic of public identifiers is that they are *globally unique*. Referring to a document with a public identifier should mean that the identifier will resolve to the same actual document on any system even though the location of that document on each system may vary. As a rule, you should never reuse public identifiers, and a published revision should have a new public identifier. Not following these rules defeats one purpose of the public identifier.

A public identifier can be any string of upper- and lowercase letters, digits, any of the following symbols: "'", "(", ")", "+", ",", "-", ".", "/", ":", "=", "?", and white space, including line breaks.

2.3.1.1 Formal public identifiers

Most public identifiers conform to the ISO 8879 standard that defines *formal public identifiers*. Formal public identifiers, frequently referred to as FPI, have a prescribed format that can ensure uniqueness:[3]

```
prefix//owner-identifier//
text-class text-description//
language//display-version
```

Here are descriptions of the identifiers in this string:

[2] This is not absolutely true. SGML allows for the possibility that the reference could be implied by the application, but this is very rarely the case.

[3] Essentially, it can ensure that two different owners won't accidentally tread on each other. Nothing can prevent a given owner from reusing public identifiers, except maybe common sense.

prefix The *prefix* is either a "+" or a "-" Registered public identifiers begin with "+"; unregistered identifiers begin with "-".

(ISO standards sometimes use a third form beginning with `ISO` and the standard number, but this form is only available to ISO.)

The purpose of registration is to guarantee a unique owner-identifier. There are few authorities with the power to issue registered public identifiers, so in practice unregistered identifiers are more common.

The Graphics Communication Association http://www.gca.org/ (GCA) can assign registered public identifiers. They do this by issuing the applicant a unique string and declaring the format of the owner identifier. For example, the Davenport Group was issued the string "A00002" and could have published DocBook using an FPI of the following form:

```
+//ISO/IEC 9070/RA::A00002//...
```

Another way to use a registered public identifier is to use the format reserved for internet domain names. For example, O'Reilly can issue documents using an FPI of the following form:

```
+//IDN oreilly.com//...
```

As of DocBook V3.1, the OASIS Technical Committee responsible for DocBook has elected to use the unregistered owner identifier, `OASIS`, thus its prefix is `-`.

```
-//OASIS//...
```

owner-identifier Identifies the person or organization that owns the identifier. Registration guarantees a unique owner identifier. Short of registration, some effort should be made to ensure that the owner identifier is globally unique. A company name, for example, is a reasonable choice as are Internet domain names. It's also not uncommon to see the names of individuals used as the owner-identifier, although clearly this may introduce collisions over time.

The owner-identifier for DocBook V3.1 is `OASIS`. Earlier versions used the owner-identifier `Davenport`.

text-class The text class identifies the kind of document that is associated with this public identifier. Common text classes are

DOCUMENT An SGML or XML document.

DTD A DTD or part of a DTD.

ELEMENTS A collection of element declarations.

ENTITIES A collection of entity declarations.

NONSGML Data that is not in SGML or XML.

DocBook is a DTD, thus its text class is DTD.

text-description This field provides a description of the document. The text description is free-form, but cannot include the string //.

The text description of DocBook is `DocBook V3.1`.

In the uncommon case of unavailable public texts (FPIs for proprietary DTDs, for example), there are a few other options available (technically in front of or in place of the text description), but they're rarely used. [4]

[4] See Appendix A of [maler96], for more details.

language Indicates the language in which the document is written. It is recommended that the ISO standard two-letter language codes be used if possible.

DocBook is an English-language DTD, thus its language is `EN`.

display-version This field, which is not frequently used, distinguishes between public texts that are the same except for the display device or system to which they apply.

For example, the FPI for the ISO Latin 1 character set is:

```
-//ISO 8879-1986//ENTITIES Added Latin 1//EN
```

A reasonable FPI for an XML version of this character set is:

```
-//ISO 8879-1986//ENTITIES Added Latin 1//EN//XML
```

2.3.2 System Identifiers

System identifiers are usually filenames on the local system. In SGML, there's no constraint on what they can be. Anything that your SGML processing system recognizes is allowed. In XML, system identifiers must be URIs (Uniform Resource Identifiers).

The use of URIs as system identifiers introduces the possibility that a system identifier can be a URN. This allows the system identifier to benefit from the same global uniqueness benefit as the public identifier. It seems likely that XML system identifiers will eventually move in this direction.

2.3.3 Catalog Files

Catalog files are the standard mechanism for resolving public identifiers into system identifiers. Some resolution mechanism is necessary because DocBook refers to its component modules with public identifiers, and those must be mapped to actual files on the system before any piece of software can actually load them.

The catalog file format was defined in 1994 by SGML Open (now OASIS). The formal specification is contained in OASIS Technical Resolution 9401:1997.

Informally, a catalog is a text file that contains a number of keyword/value pairs. The most frequently used keywords are `PUBLIC`, `SYSTEM`, `SGMLDECL`, `DTDDECL`, `CATALOG`, `OVERRIDE`, `DELEGATE`, and `DOCTYPE`.

PUBLIC The `PUBLIC` keyword maps public identifiers to system identifiers:

```
PUBLIC "-//OASIS//DTD DocBook V3.1//EN" "docbook/3.1/docbook.dtd"
```

SYSTEM The `SYSTEM` keyword maps system identifiers to system identifiers:

```
SYSTEM "http://nwalsh.com/docbook/xml/1.3/db3xml.dtd"
    "docbook/xml/1.3/db3xml.dtd"
```

SGMLDECL The `SGMLDECL` keyword identifies the system identifier of the SGML Declaration that should be used:

```
SGMLDECL "docbook/3.1/docbook.dcl"
```

DTDDECL Like `SGMLDECL`, `DTDDECL` identifies the SGML Declaration that should be used. `DTDDECL` associates a declaration with a particular public identifier for a DTD:

```
DTDDECL "-//OASIS//DTD DocBook V3.1//EN" "docbook/3.1/docbook.dcl"
```

Unfortunately, it is not supported by the free tools that are available. The practical benefit of `DTDDECL` can usually be achieved, albeit in a slightly cumbersome way, with multiple catalog files.

CATALOG The CATALOG keyword allows one catalog to include the content of another. This can make maintenance somewhat easier and allows a system to directly use the catalog files included in DTD distributions. For example, the DocBook distribution includes a catalog file. Rather than copying each of the declarations in that catalog into your system catalog, you can simply include the contents of the DocBook catalog:

```
CATALOG "docbook/3.1/catalog"
```

OVERRIDE The OVERRIDE keyword indicates whether or not public identifiers override system identifiers. If a given declaration includes both a system identifer and a public identifier, most systems attempt to process the document referenced by the system identifier, and consequently ignore the public identifier. Specifying

```
OVERRIDE YES
```

in the catalog informs the processing system that resolution should be attempted first with the public identifier.

DELEGATE The DELEGATE keyword allows you to specify that some set of public identifiers should be resolved by another catalog. Unlike the CATALOG keyword, which loads the referenced catalog, DELEGATE does nothing until an attempt is made to resolve a public identifier.

The DELEGATE entry specifies a partial public identifier and an alternate catalog:

```
DELEGATE "-//OASIS" "/usr/sgml/oasis/catalog"
```

Partial public identifers are simply initial substring matches. Given the preceding entry, if an attempt is made to match any public identifier that begins with the string -//OASIS, the alternate catalog /usr/sgml/oasis/catalog will be used instead of the current catalog.

DOCTYPE The DOCTYPE keyword allows you to specify a default system identifier. If an SGML document begins with a DOCTYPE declaration that specifies neither a public identifier nor a system identifier (or is missing a DOCTYPE declaration altogether), the DOCTYPE declaration may provide a default:

```
DOCTYPE BOOK n:/share/sgml/docbook/3.1/docbook.dtd
```

A small fragment of an actual catalog file is shown in Example 2.3.1.

Example 2.3.1: A Sample Catalog

```
-- Comments are delimited by pairs of double-hyphens,
   as in SGML and XML comments. --

OVERRIDE YES

SGMLDECL "n:/share/sgml/docbook/3.1/docbook.dcl"

DOCTYPE  BOOK  n:/share/sgml/docbook/3.1/docbook.dtd

PUBLIC "-//OASIS//DTD DocBook V3.1//EN"
  n:/share/sgml/docbook/3.1/docbook.dtd

SYSTEM "http://nwalsh.com/docbook/xml/1.3/db3xml.dtd"
  n:/share/sgml/Norman_Walsh/db3xml/db3xml.dtd
```

callout ??? Catalog files may also include comments. callout ??? This catalog specifies that public identifiers should be used in favor of system identifiers, if both are present. callout ??? The default declaration specified by this catalog is the DocBook declaration. callout ??? Given an explicit (or implied) SGML DOCTYPE of

```
<!DOCTYPE BOOK SYSTEM>
```

use `n:/share/sgml/docbook/3.1/docbook.dtd` as the default system identifier. Note that this can only apply to SGML documents because the DOCTYPE declaration above is not a valid XML element. callout ??? Map the OASIS public identifer to the local copy of the DocBook V3.1 DTD. callout ??? Map a system identifer for the XML version of DocBook to a local version.

A few notes:

- It's not uncommon to have several catalog files. See below, Section 2.3.3.1".
- Like attributes on elements you can quote, the public identifier and system identifier are surrounded by either single or double quotes.
- White space in the catalog file is generally irrelevant. You can use spaces, tabs, or new lines between keywords and their arguments.
- When a relative system identifier is used, it is considered to be relative to the location of the catalog file, not the document being processed.

2.3.3.1 Locating catalog files

Catalog files go a long way towards making documents more portable by introducing a level of indirection. A problem still remains, however: how does a processor locate the appropriate catalog file(s)? OASIS outlines a complete interchange packaging scheme, but for most applications the answer is simply that the processor looks for a file called `catalog` or `CATALOG`.

Some applications allow you to specify a list of directories that should be examined for catalog files. Other tools allow you to specify the actual files.

Note that even if a list of directories or catalog files is provided, applications may still load catalog files that occur in directories in which other documents are found. For example, SP and Jade always load the catalog file that occurs in the directory in which a DTD or document resides, even if that directory is not on the catalog file list.

2.4 Physical Divisions: Breaking a Document into Physical Chunks

The rest of this chapter describes how you can break documents into logical chunks, such as books, chapters, sections, and so on. Before we begin, and while the subject of the internal subset is fresh in your mind, let's take a quick look at how to break documents into separate physical chunks.

Actually, we've already told you how to do it. If you recall, in the preceding sections we had declarations of the form:

```
<!ENTITY name SYSTEM "filename">
```

If you refer to the entity *`name`* in your document after this declaration, the system will insert the contents of the file *`filename`* into your document at that point. So, if you've got a book that consists of three chapters and two appendixes, you might create a file called `book.sgm`, which looks like this:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY chap1 SYSTEM "chap1.sgm">
<!ENTITY chap2 SYSTEM "chap2.sgm">
<!ENTITY chap3 SYSTEM "chap3.sgm">
<!ENTITY appa SYSTEM "appa.sgm">
<!ENTITY appb SYSTEM "appb.sgm">
]>
<book><title>My First Book</title>
&chap1;
&chap2;
&chap3;
&appa;
&appb;
</book>
```

You can then write the chapters and appendixes conveniently in separate files. Note that these files do not and must not have document type declarations.

For example, Chapter 1 might begin like this:

```
<chapter id="ch1"><title>My First Chapter</title>
<para>My first paragraph.</para>
...
```

But it should not begin with its own document type declaration:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter id="ch1"><title>My First Chapter</title>
<para>My first paragraph.</para>
...
```

2.5 Logical Divisions: The Categories of Elements in DocBook

DocBook elements can be divided broadly into these categories:

- Sets
- Books
- Divisions, which divide books into parts
- Components, which divide books or divisions into chapters
- Sections, which subdivide components
- Meta-information elements
- Block elements
- Inline elements

In the rest of this section, we'll describe briefly the elements that make up these categories. This section is designed to give you an overview. It is not an exhaustive list of every element in DocBook.

For more information about any specific element and the elements that it may contain, consult the reference page for the element in question.

2.5.1 Sets

A `Set` contains two or more `Book`s. It's the hierarchical top of DocBook. You use the `Set` tag, for example, for a series of books on a single subject that you want to access and maintain as a single unit, such as the manuals for an airplane engine or the documentation for a programming language.

2.5.2 Books

A Book is probably the most common top-level element in a document. The DocBook definition of a book is very loose and general. Given the variety of books authored with DocBook and the number of different conventions for book organization used in countries around the world, attempting to impose a strict ordering of elements can make the content model extremely complex. But DocBook gives you free reign. It's very reasonable to use a local customization layer to impose a more strict ordering for your applications.

Books consist of a mixture of the following elements:

Dedication Dedication pages almost always occur at the front of a book.

Navigational Components There are a few component-level elements designed for navigation: ToC, for Tables of Contents; LoT, for Lists of Titles (for lists of figures, tables, examples, and so on); and Index, for indexes.

Divisions Divisions are the first hierarchical level below Book. They contain Parts and References. Parts, in turn, contain components. References contain RefEntrys. These are discussed more thoroughly in Section 2.9".

Books can contain components directly and are not required to contain divisions.

Components These are the chapter-like elements of a Book.

2.5.3 Components

Components are the chapter-like elements of a Book or Part: Preface, Chapter, Appendix, Glossary, and Bibliography. Articles can also occur at the component level. We describe Articles in more detail in the section titled Section 2.8". Components generally contain block elements and/or sections, and some can contain navigational components and RefEntrys.

2.5.4 Sections

There are several flavors of sectioning elements in DocBook:

Sect1...Sect5 elements The Sect1...Sect5 elements are the most common sectioning elements. They can occur in most component-level elements. These numbered section elements must be properly nested (Sect2s can only occur inside Sect1s, Sect3s can only occur inside Sect2s, and so on). There are five levels of numbered sections.

Section element The Section element, introduced in DocBook V3.1, is an alternative to numbered sections. Sections are recursive, meaning that you can nest them to any depth desired.

SimpleSect element In addition to numbered sections, there's the SimpleSect element. It is a terminal section that can occur at any level, but it cannot have any other sectioning element nested within it.

BridgeHead A BridgeHead provides a section title without any containing section.

RefSect1...RefSect3 elements These elements, which occur only in RefEntrys, are analogous to the numbered section elements in components. There are only three levels of numbered section elements in a RefEntry.

GlossDiv, BiblioDiv, and IndexDiv Glossarys, Bibliographys, and Indexes can be broken into top-level divisions, but not sections. Unlike sections, these elements do not nest.

2.5.5 Meta-Information

All of the elements at the section level and above include a wrapper for meta-information about the content. See, for example, `BookInfo`.

The meta-information wrapper is designed to contain bibliographic information about the content (`Author`, `Title`, `Publisher`, and so on) as well as other meta-information such as revision histories, keyword sets, and index terms.

2.5.6 Block Elements

The block elements occur immediately below the component and sectioning elements. These are the (roughly) paragraph-level elements in DocBook. They can be divided into a number of categories: lists, admonitions, line-specific environments, synopses of several sorts, tables, figures, examples, and a dozen or more miscellaneous elements. Block vs. Inline Elements

At the paragraph-level, it's convenient to divide elements into two classes, *block* and *inline*. From a structural point of view, this distinction is based loosely on their relative size, but it's easiest to describe the difference in terms of their presentation.

Block elements are usually presented with a paragraph (or larger) break before and after them. Most can contain other block elements, and many can contain character data and inline elements. Paragraphs, lists, sidebars, tables, and block quotations are all common examples of block elements.

Inline elements are generally represented without any obvious breaks. The most common distinguishing mark of inline elements is a font change, but inline elements may present no visual distinction at all. Inline elements contain character data and possibly other inline elements, but they never contain block elements. Inline elements are used to mark up data such as cross references, filenames, commands, options, subscripts and superscripts, and glossary terms.

2.5.6.1 Lists

There are seven list elements in DocBook:

`CalloutList` A list of `CallOut`s and their descriptions. `CallOut`s are marks, frequently numbered and typically on a graphic or verbatim environment, that are described in a `CalloutList`, outside the element in which they occur.

`GlossList` A list of glossary terms and their definitions.

`ItemizedList` An unordered (bulleted) list. There are attributes to control the marks used.

`OrderedList` A numbered list. There are attributes to control the type of enumeration.

`SegmentedList` A repeating set of named items. For example, a list of states and their capitals might be represented as a `SegmentedList`.

`SimpleList` An unadorned list of items. `SimpleList`s can be inline or arranged in columns.

`VariableList` A list of terms and definitions or descriptions. (This list of list types is a `VariableList`.)

2.5.6.2 Admonitions

There are five types of admonitions in DocBook: `Caution`, `Important`, `Note`, `Tip`, and `Warning`.

All of the admonitions have the same structure: an optional `Title` followed by paragraph-level elements. The DocBook DTD does not impose any specific semantics on the individual admonitions. For example, DocBook does not mandate that `Warning`s be reserved for cases where bodily harm can result.

2.5.6.3 Line-specific environments

These environments preserve whitespace and line breaks in the source text. DocBook does not provide the equivalent of HTML's BR tag, so there's no way to interject a line break into normal running text.

Address The Address element is intended for postal addresses. In addition to being line-specific, Address contains additional elements suitable for marking up names and addresses.

LiteralLayout A LiteralLayout does not have any semantic association beyond the preservation of whitespace and line breaks. In particular, while ProgramListing and Screen are frequently presented in a fixed-width font, a change of fonts is not necessarily implied by LiteralLayout.

ProgramListing A ProgramListing is a verbatim environment, usually presented in Courier or some other fixed-width font, for program sources, code fragments, and similar listings.

Screen A Screen is a verbatim or literal environment for text screen-captures, other fragments of an ASCII display, and similar things. Screen is also a frequent catch-all for any verbatim text.

ScreenShot ScreenShot is actually a wrapper for a Graphic intended for screen shots of a GUI for example.

Synopsis A Synopsis is a verbatim environment for command and function synopsis.

2.5.6.4 Examples, figures, and tables

Examples, Figures, and Tables are common block-level elements: Example, InformalExample, Figure, InformalFigure, Table, and InformalTable.

The distinction between formal and informal elements is that formal elements have titles while informal ones do not. The InformalFigure element was introduced in DocBook V3.1. In prior versions of DocBook, you could only achieve the effect of an informal figure by placing its content, unwrapped, at the location where the informal figure was desired.

2.5.6.5 Paragraphs

There are three paragraph elements: Para, SimPara (simple paragraphs may not contain other block-level elements), and FormalPara (formal paragraphs have titles).

2.5.6.6 Equations

There are two block-equation elements, Equation and InformalEquation (for inline equations, use InlineEquation).

Informal equations don't have titles. For reasons of backward-compatibility, Equations are not required to have titles. However, it may be more difficult for some stylesheet languages to properly enumerate Equations if they lack titles.

2.5.6.7 Graphics

Graphics occur most frequently in Figures and ScreenShots, but they can also occur without a wrapper. DocBook considers a Graphic a block element, even if it appears to occur inline. For graphics that you want to be represented inline, use InlineGraphic.

DocBook V3.1 introduced a new element to contain graphics and other media types: MediaObject and its inline cousin, InlineMediaObject. These elements may contain video, audio, image, and text data. A single media object can contain several alternative forms from which the presentation system can select the most appropriate object.

2.5.6.8 Questions and answers

DocBook V3.1 introduced the `QandASet` element, which is suitable for FAQs (Frequently Asked Questions) and other similar collections of `Question`s and `Answer`s.

2.5.6.9 Miscellaneous block elements

The following block elements are also available:

`BlockQuote` A block quotation. Block quotations may have `Attribution`s.

`CmdSynopsis` An environment for marking up all the parameters and options of a command.

`Epigraph` A short introduction, typically a quotation, at the beginning of a document. `Epigraph`s may have `Attribution`s.

`FuncSynopsis` An environment for marking up the return value and arguments of a function.

`Highlights` A summary of the main points discussed in a book component (chapter, section, and so on).

`MsgSet` A set of related error messages.

`Procedure` A procedure. Procedures contain `Step`s, which may contain `SubStep`s.

`Sidebar` A sidebar.

2.5.7 Inline Elements

Users of DocBook are provided with a surfeit of inline elements. Inline elements are used to mark up running text. In published documents, inline elements often cause a font change or other small change, but they do not cause line or paragraph breaks.

In practice, writers generally settle on the tagging of inline elements that suits their time and subject matter. This may be a large number of elements or only a handful. What is important is that you choose to mark up not every possible item, but only those for which distinctive tagging will be useful in the production of the finished document for the readers who will search through it.

The following comprehensive list may be a useful tool for the process of narrowing down the elements that you will choose to mark up; it is not intended to overwhelm you by its sheer length. For convenience, we've divided the inlines into several subcategories.

The classification used here is not meant to be authoritative, only helpful in providing a feel for the nature of the inlines. Several elements appear in more than one category, and arguments could be made to support the placement of additional elements in other categories or entirely new categories.

2.5.7.1 Traditional publishing inlines

These inlines identify things that commonly occur in general writing:

`Abbrev` An abbreviation, especially one followed by a period.

`Acronym` An often pronounceable word made from the initial (or selected) letters of a name or phrase.

`Emphasis` Emphasized text.

Footnote A footnote. The location of the `Footnote` element identifies the location of the first reference to the footnote. Additional references to the same footnote can be inserted with `FootnoteRef`.

Phrase A span of text.

Quote An inline quotation.

Trademark A trademark.

2.5.7.2 Cross references

The cross reference inlines identify both explicit cross references, such as `Link`, and implicit cross references like `GlossTerm`. You can make the most of the implicit references explicit with a LinkEnd attribute.

Anchor A spot in the document.

Citation An inline bibliographic reference to another published work.

CiteRefEntry A citation to a reference page.

CiteTitle The title of a cited work.

FirstTerm The first occurrence of a term.

GlossTerm A glossary term.

Link A hypertext link.

OLink A link that addresses its target indirectly, through an entity.

ULink A link that addresses its target by means of a URL (Uniform Resource Locator).

XRef A cross reference to another part of the document.

2.5.7.3 Markup

These inlines are used to mark up text for special presentation:

ForeignPhrase A word or phrase in a language other than the primary language of the document.

WordAsWord A word meant specifically as a word and not representing anything else.

ComputerOutput Data, generally text, displayed or presented by a computer.

Literal Inline text that is some literal value.

Markup A string of formatting markup in text that is to be represented literally.

Prompt A character or string indicating the start of an input field in a computer display.

Replaceable Content that may or must be replaced by the user.

SGMLTag A component of SGML markup.

UserInput Data entered by the user.

2.5.7.4 Mathematics

DocBook does not define a complete set of elements for representing equations. No one has ever pressed the DocBook maintainers to add this functionality, and the prevailing opinion is that incorporating MathML http://www.w3.org/TR/REC-MathML/ using a mechanism like namespaces http://www.w3.org/TR/REC-xml-names/ is probably the best long-term solution.

InlineEquation A mathematical equation or expression occurring inline.

Subscript A subscript (as in H_2O, the molecular formula for water).

Superscript A superscript (as in x^2, the mathematical notation for x multiplied by itself).

2.5.7.5 User interfaces

These elements describe aspects of a user interface:

Accel A graphical user interface (GUI) keyboard shortcut.

GUIButton The text on a button in a GUI.

GUIIcon Graphic and/or text appearing as a icon in a GUI.

GUILabel The text of a label in a GUI.

GUIMenu The name of a menu in a GUI.

GUIMenuItem The name of a terminal menu item in a GUI.

GUISubmenu The name of a submenu in a GUI.

KeyCap The text printed on a key on a keyboard.

KeyCode The internal, frequently numeric, identifier for a key on a keyboard.

KeyCombo A combination of input actions.

KeySym The symbolic name of a key on a keyboard.

MenuChoice A selection or series of selections from a menu.

MouseButton The conventional name of a mouse button.

Shortcut A key combination for an action that is also accessible through a menu.

2.5.7.6 Programming languages and constructs

Many of the technical inlines in DocBook are related to programming.

Action A response to a user event.

ClassName The name of a class, in the object-oriented programming sense.

Constant A programming or system constant.

ErrorCode An error code.

ErrorName An error name.

ErrorType The classification of an error message.

Function The name of a function or subroutine, as in a programming language.

Interface An element of a GUI.

InterfaceDefinition The name of a formal specification of a GUI.

Literal Inline text that is some literal value.

MsgText The actual text of a message component in a message set.

Parameter A value or a symbolic reference to a value.

Property A unit of data associated with some part of a computer system.

Replaceable Content that may or must be replaced by the user.

ReturnValue The value returned by a function.

StructField A field in a structure (in the programming language sense).

StructName The name of a structure (in the programming language sense).

Symbol A name that is replaced by a value before processing.

Token A unit of information.

Type The classification of a value.

VarName The name of a variable.

2.5.7.7 Operating systems

These inlines identify parts of an operating system, or an operating environment:

Application The name of a software program.

Command The name of an executable program or other software command.

EnVar A software environment variable.

Filename The name of a file.

MediaLabel A name that identifies the physical medium on which some information resides.

MsgText The actual text of a message component in a message set.

Option An option for a software command.

Parameter A value or a symbolic reference to a value.

Prompt A character or string indicating the start of an input field in a computer display.

SystemItem A system-related item or term.

2.5.7.8 General purpose

There are also a number of general-purpose technical inlines.

Application The name of a software program.

Database The name of a database, or part of a database.

Email An email address.

Filename The name of a file.

Hardware A physical part of a computer system.

InlineGraphic An object containing or pointing to graphical data that will be rendered inline.

Literal Inline text that is some literal value.

MediaLabel A name that identifies the physical medium on which some information resides.

Option An option for a software command.

Optional Optional information.

Replaceable Content that may or must be replaced by the user.

Symbol A name that is replaced by a value before processing.

Token A unit of information.

Type The classification of a value.

2.6 Making a DocBook Book

A typical `Book`, in English at least, consists of some meta-information in a `BookInfo` (`Title`, `Author`, `Copyright`, and so on), one or more `Preface`s, several `Chapter`s, and perhaps a few `Appendix`es. A `Book` may also contain `Bibliography`s, `Glossary`s, `Index`es and a `Colophon`.

Example 2.6.1 shows the structure of a typical book. Additional content is required where the ellipses occur.

Example 2.6.1: A Typical Book

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<book>
<bookinfo>
  <title>My First Book</title>
  <author><firstname>Jane</firstname><surname>Doe</surname></author>
  <copyright><year>1998</year><holder>Jane Doe</holder></copyright>
</bookinfo>
<preface><title>Foreword</title> ... </preface>
<chapter> ... </chapter>
<chapter> ... </chapter>
<chapter> ... </chapter>
<appendix> ... </appendix>
<appendix> ... </appendix>
<index> ... </index>
</book>
```

2.7 Making a Chapter

`Chapter`s, `Preface`s, and `Appendix`es all have a similar structure. They consist of a `Title`, possibly some additional meta-information, and any number of block-level elements followed by any number of top-level sections. Each section may in turn contain any number of block-level elements followed by any number from the next section level, as shown in Example 2.7.1.

Example 2.7.1: A Typical Chapter

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>My Chapter</title>
<para> ... </para>
<sect1><title>First Section</title>
<para> ... </para>
<example> ... </example>
</sect1>
</chapter>
```

2.8 Making an Article

For documents smaller than a book, such as: journal articles, white papers, or technical notes, `Article` is frequently the most logical starting point. The body of an `Article` is essentially the same as the body of a `Chapter` or any other component-level element, as shown in Example 2.8.1

`Article`s may include `Appendix`es, `Bibliography`s, `Index`es and `Glossary`s.

Example 2.8.1: A Typical Article

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<article>
<artheader>
  <title>My Article</title>
  <author><honorific>Dr</honorific><firstname>Emilio</firstname>
          <surname>Lizardo</surname></author>
</artheader>
<para> ... </para>
<sect1><title>On the Possibility of Going Home</title>
<para> ... </para>
</sect1>
<bibliography> ... </bibliography>
</article>
```

2.9 Making a Reference Page

The reference page or manual page in DocBook was inspired by, and in fact designed to reproduce, the common UNIX “manpage” concept. (We use the word "page" loosely here to mean a document of variable length containing reference material on a specific topic.) DocBook is rich in markup tailored for such documents, which often vary greatly in content, however well-structured they may be. To reflect both the structure and the variability of such texts, DocBook specifies that reference pages have a strict sequence of parts, even though several of them are actually optional.

Of the following sequence of elements that may appear in a `RefEntry`, only two are obligatory: `RefNameDiv` and `RefSect1`.

`DocInfo` The `DocInfo` element contains meta-information about the reference page (which should not be confused with `RefMeta`, which it precedes). It marks up information about the author of the document, or the product to which it pertains, or the document’s revision history, or other such information.

`RefMeta` `RefMeta` contains a title for the reference page (which may be inferred if the `RefMeta` element is not present) and an indication of the volume number in which this reference page occurs. The `ManVolNum`

is a very UNIX-centric concept. In traditional UNIX documentation, the subject of a reference page is typically identified by name and volume number; this allows you to distinguish between the **uname** command, "uname(1)" in volume 1 of the documentation and the `uname` function, "uname(3)" in volume 3.

Additional information of this sort such as conformance or vendor information specific to the particular environment you are working in, may be stored in `RefMiscInfo`.

`RefNameDiv` The first obligatory element is `RefNameDiv`, which is a wrapper for information about whatever you're documenting, rather than the document itself. It can begin with a `RefDescriptor` if several items are being documented as a group and the group has a name. The `RefNameDiv` must contain at least one `RefName`, that is, the name of whatever you're documenting, and a single short statement that sums up the use or function of the item(s) at a glance: their `RefPurpose`. Also available is the `RefClass`, intended to detail the operating system configurations that the software element in question supports.

If no `RefEntryTitle` is given in the `RefMeta`, the title of the reference page is the `RefDescriptor`, if present, or the first `RefName`.

`RefSynopsisDiv` A `RefSynopsisDiv` is intended to provide a quick synopsis of the topic covered by the reference page. For commands, this is generally a syntax summary of the command, and for functions, the function prototype, but other options are possible. A `Title` is allowed, but not required, presumably because the application that processes reference pages will generate the appropriate title if it is not given. In traditional UNIX documentation, its title is always "Synopsis".

`RefSect1`...`RefSect3` Within `RefEntry`s, there are only three levels of sectioning elements: `RefSect1`, `RefSect2`, and `RefSect3`.

Example 2.9.1 shows the beginning of a `RefEntry` that illustrates one possible reference page:

2.10 Making Front- and Backmatter

DocBook contains markup for the usual variety of front- and backmatter necessary for books and articles: indexes, glossaries, bibliographies, and tables of contents. In many cases, these components are generated automatically, at least in part, from your document by an external processor, but you can create them by hand, and in either case, store them in DocBook.

Some forms of backmatter, like indexes and glossaries, usually require additional markup *in the document* to make generation by an application possible. Bibliographies are usually composed by hand like the rest of your text, unless you are automatically selecting bibliographic entries out of some larger database. Our principal concern here is to acquaint you with the kind of markup you need to include in your documents if you want to construct these components.

Frontmatter, like the table of contents, is almost always generated automatically from the text of a document by the processing application. If you need information about how to mark up a table of contents in DocBook, please consult the reference page for `ToC`.

2.10.1 Making an Index

In some highly-structured documents such as reference manuals, you can automate the whole process of generating an index successfully without altering or adding to the original source. You can design a processing application to select the information and compile it into an adequate index. But this is rare.

In most cases—and even in the case of some reference manuals—a useful index still requires human intervention to mark occurrences of words or concepts that will appear in the text of the index.

2.10.1.1 Marking index terms

Docbook distinguishes two kinds of index markers: those that are singular and result in a single page entry in the index itself, and those that are multiple and refer to a range of pages.

You put a singular index marker where the subject it refers to actually occurs in your text:

```
<para>
The tiger<indexterm>
<primary>Big Cats</primary>
<secondary>Tigers</secondary></indexterm>
is a very large cat indeed.
</para>
```

This index term has two levels, `primary` and `secondary`. They correspond to an increasing amount of indented text in the resultant index. DocBook allows for three levels of index terms, with the third labeled `tertiary`.

There are two ways that you can index a range of text. The first is to put index marks at both the beginning and end of the discussion. The mark at the beginning asserts that it is the start of a range, and the mark at the end refers back to the beginning. In this way, the processing application can determine what range of text is indexed. Here's the previous tiger example recast as starting and ending index terms:

```
<para>
The tiger<indexterm id="tiger-desc" class="startofrange">
<primary>Big Cats</primary>
<secondary>Tigers</secondary></indexterm>
is a very large cat indeed...
</para>
⋮
<para>
So much for tigers<indexterm startref="tiger-desc" class="endofrange">. Let's talk ab
leopards.
</para>
```

Note that the mark at the start of the range identifies itself as the start of a range with the Class attribute, and provides an ID. The mark at the end of the range points back to the start.

Another way to mark up a range of text is to specify that the entire content of an element, such as a chapter or section, is the complete range. In this case, all you need is for the index term to point to the ID of the element that contains the content in question. The Zone attribute of `indexterm` provides this functionality.

One of the interesting features of this method is that the actual index marks do not have to occur anywhere near the text being indexed. It is possible to collect all of them together, for example, in one file, but it is not invalid to have the index marker occur near the element it indexes.

Suppose the discussion of tigers in your document comprises a whole text object (like a `Sect1` or a `Chapter`) with an ID value of `tiger-desc`. You can put the following tag anywhere in your document to index that range of text:

```
<indexterm zone="tiger-desc">
<primary>Big Cats</primary>
<secondary>Tigers</secondary></indexterm>
```

DocBook also contains markup for index hits that point to other index hits (of the same type such as "See Cats, big" or "See also Lions"). See the reference pages for `See` and `SeeAlso`.

2.10.1.2 Printing an index

After you have added the appropriate markup to your document, an external application can use this information to build an index. The resulting index must have information about the page numbers on which the concepts appear. It's usually the document formatter that builds the index. In this case, it may never be instantiated in DocBook.

However, there are applications that can produce an index marked up in DocBook. The following example includes some one- and two-level `IndexEntry` elements (which correspond to the primary and secondary levels in the `indexterm`s themselves) that begin with the letter D:

```
<!DOCTYPE index PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<index><title>Index</title>
<indexdiv><title>D</title>
<indexentry>
  <primaryie>database (bibliographic), 253, 255</primaryie>
     <secondaryie>structure, 255</secondaryie>
     <secondaryie>tools, 259</secondaryie>
</indexentry>
<indexentry>
  <primaryie>dates (language specific), 179</primaryie>
</indexentry>
<indexentry>
  <primaryie>DC fonts, <emphasis>172</emphasis>, 177</primaryie>
     <secondaryie>Math fonts, 177</secondaryie>
</indexentry>
</indexdiv>
</index>
```

2.10.2 Making a Glossary

`Glossary`s, like `Bibliography`s, are often constructed by hand. However, some applications are capable of building a skeletal index from glossary term markup in the document. If all of your terms are defined in some glossary database, it may even be possible to construct the complete glossary automatically.

To enable automatic glossary generation, or simply automatic linking from glossary terms in the text to glossary entries, you must add markup to your documents. In the text, you markup a term for compilation later with the inline `GlossTerm` tag. This tag can have a LinkEnd attribute whose value is the ID of the actual entry in the glossary.[5]

For instance, if you have this markup in your document:

```
<glossterm linkend="xml">Extensible Markup Language</glossterm> is a new standard...
```

your glossary might look like this:

```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<glossary><title>Example Glossary</title>
⋮
<glossdiv><title>E</title>

<glossentry id="xml"><glossterm>Extensible Markup Language</glossterm>
  <acronym>XML</acronym>
<glossdef>
  <para>Some reasonable definition here.</para>
  <glossseealso otherterm="sgml">
</glossdef>
</glossentry>

</glossdiv>
```

Note that the `GlossTerm` tag reappears in the glossary to mark up the term and distinguish it from its definition within the `GlossEntry`. The ID that the `GlossEntry` referenced in the text is the ID of the `GlossEntry` in the `Glossary` itself. You can use the link between source and glossary to create a link in the online form of your document, as we have done with the online form of the glossary in this book.

[5] Some sophisticated formatters might even be able to establish the link simply by examining the content of the terms and the glossary. In that case, the author is not required to make explicit links.

2.10.3 Making a Bibliography

There are two ways to set up a bibliography in DocBook: you can have the data *raw* or *cooked*. Here's an example of a raw bibliographical item, wrapped in the `Biblioentry` element:

```
<biblioentry xreflabel="Kites75">
  <authorgroup>
    <author><firstname>Andrea</firstname><surname>Bahadur</surname></author>
    <author><firstname>Mark</><surname>Shwarek</></author>
  </authorgroup>
  <copyright><year>1974</year><year>1975</year>
     <holder>Product Development International Holding N. V.</holder>
     </copyright>
  <isbn>0-88459-021-6</isbn>
  <publisher>
    <publishername>Plenary Publications International, Inc.</publishername>
  </publisher>
  <title>Kites</title>
  <subtitle>Ancient Craft to Modern Sport</subtitle>
  <pagenums>988-999</pagenums>
  <seriesinfo>
    <title>The Family Creative Workshop</title>
    <seriesvolnums>1-22</seriesvolnums>
    <editor>
      <firstname>Allen</firstname>
      <othername role=middle>Davenport</othername>
      <surname>Bragdon</surname>
      <contrib>Editor in Chief</contrib>
    </editor>
  </seriesinfo>
</biblioentry>
```

The "raw" data in a `Biblioentry` is comprehensive to a fault—there are enough fields to suit a host of different bibliographical styles, and that is the point. An abundance of data requires processing applications to select, punctuate, order, and format the bibliographical data, and it is unlikely that all the information provided will actually be output.

All the "cooked" data in a `Bibliomixed` entry in a bibliography, on the other hand, is intended to be presented to the reader in the form and sequence in which it is provided. It even includes punctuation between the fields of data:

```
<bibliomixed>
  <bibliomset relation=article>
    <surname>Walsh</surname>, <firstname>Norman</firstname>.
    <title role=article>Introduction to Cascading Style Sheets</title>.
  </bibliomset>
  <bibliomset relation=journal>
    <title>The World Wide Web Journal</title>
    <volumenum>2</volumenum><issuenum>1</issuenum>.
    <publishername>O'Reilly & Associates, Inc.</publishername> and
    <corpname>The World Wide Web Consortium</corpname>.
    <pubdate>Winter, 1996</pubdate></bibliomset>.
</bibliomixed>
```

Clearly, these two ways of marking up bibliographical entries are suited to different circumstances. You should use one or the other for your bibliography, not both. Strictly speaking, mingling the raw and the cooked may be "kosher" as far as the DTD is concerned, but it will almost certainly cause problems for most processing applications.

Example 2.9.1: A Sample Reference Page

```
<refentry id="printf">

<refmeta>
<refentrytitle>printf</refentrytitle>
<manvolnum>3S</manvolnum>
</refmeta>

<refnamediv>
<refname>printf</refname>
<refname>fprintf</refname>
<refname>sprintf</refname>
<refpurpose>print formatted output</refpurpose>
</refnamediv>

<refsynopsisdiv>

<funcsynopsis>
<funcsynopsisinfo>
#include &lt;stdio.h&gt;
</funcsynopsisinfo>
<funcprototype>
  <funcdef>int <function>printf</function></funcdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>

<funcprototype>
  <funcdef>int <function>fprintf</function></funcdef>
  <paramdef>FILE *<parameter>strm</parameter></paramdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>

<funcprototype>
  <funcdef>int <function>sprintf</function></funcdef>
  <paramdef>char *<parameter>s</parameter></paramdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>
</funcsynopsis>

</refsynopsisdiv>

<refsect1><title>Description</title>
<para>
<indexterm><primary>functions</primary>
  <secondary>printf</secondary></indexterm>
<indexterm><primary>printing function</primary></indexterm>

<function>printf</function> places output on the standard
output stream stdout.
&hellip;
</para>
```

Chapter 3

Parsing DocBook Documents

A key feature of SGML and XML markup is that you *validate* it. The DocBook DTD is a precise description of valid nesting, the order of elements, and their content. All DocBook documents must conform to this description or they are not DocBook documents (by definition).

A *validating parser* is a program that can read the DTD and a particular document and determine whether the exact nesting and order of elements in the document is valid according to the DTD.

If you are not using a structured editor that can enforce the markup as you type, validation with an external parser is a particularly important step in the document creation process. You cannot expect to get rational results from subsequent processing (such as document publishing) if your documents are not valid.

The most popular free SGML parser is SP by James Clark, available at `http://www.jclark.com/`.

SP includes **nsgmls**, a fast command-line parser. In the world of free validating XML parsers, James Clark's **xp** is a popular choice.

NOTE

Not all XML parsers are validating, and although a non-validating parser may have many uses, it cannot ensure that your documents are valid according to the DTD.

3.1 Validating Your Documents

The exact way in which the parser is executed varies according to the parser in use, naturally. For information about your particular parser, consult the documentation that came with it.

3.1.1 Using nsgmls

The **nsgmls** command from SP is a validating SGML parser. The options used in the example below suppress the normal output (`-s`), except for error messages, print the version number (`-v`), and specify the catalog file that should be used to map public identifiers to system identifiers. Printing the version number guarantees that you always get *some* output, so that you know the command ran successfully:

```
[n:\dbtdg] nsgmls -sv -c \share\sgml\catalog test.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
```

Because no error messages were printed, we know our document is valid. If you're working with a document that you discover has many errors, the `-f` option offers a handy way to direct the errors to a file so they don't all scroll off your screen.

If you want to validate an XML document with SP, you must make sure that SP uses the correct declaration. An XML declaration called `xml.dcl` is included with SP.

The easiest way to make sure that SP uses `xml.dcl` is to include the declaration explicitly on the command line when you run **nsgmls** (or Jade, or other SP tools):

```
[n:\dbtdg] nsgmls -sv -c \share\sgml\catalog m:\jade\xml.dcl test.xml
m:\jade\nsgmls.exe:I: SP version "1.3.2"
```

3.1.2 Using xp

The xp distribution includes several sample programs. One of these programs, **Time**, performs a validating parse of the document and prints the amount of time required to parse the DTD and the document. This program makes an excellent validity checker:

```
java com.jclark.xml.apps.Time examples\simple.xml
6.639
```

The result states that it took 6.639 seconds to parse the DTD and the document. This indicates that the document is valid. If the document is invalid, additional error messages are displayed.

3.2 Understanding Parse Errors

Every parser produces slightly different error messages, but most indicate exactly (at least technically)[1] what is wrong and where the error occurred. With a little experience, this information is all you'll need to quickly identify what's wrong.

In the rest of this section, we'll look at a number of common errors and the messages they produce in SP. We've chosen SP for the rest of these examples because that is the same parser used by Jade, which we'll be discussing further in Chapter 4, *Publishing DocBook Documents*.

3.2.1 DTD Cannot Be Found

The telltale sign that SP could not find the DTD, or some module of the DTD, is the error message: "cannot generate system identifier for public text . . . ". Generally, the errors that occur after this are spurious; if SP couldn't find some part of the DTD, it's likely to think that *everything* is wrong.

Careful examination of the following document will show that we've introduced a simple typographic error into the public identifier (the word "DocBook" is misspelled with a lowercase "b"):

SP responds dramatically to this error:

```
hermes:/documents/books/tdg/examples/errs$ nsgmls -sv -c cat1 /usr/lib/sgml/declaration/xm
nsgmls:I: SP version "1.3.4"
nsgmls:nodtd.sgm:2:76:E: could not resolve host "www.oasis-open.org" (try again later)
nsgmls:nodtd.sgm:2:76:E: DTD did not contain element declaration for document type name
nsgmls:nodtd.sgm:3:8:E: element "chapter" undefined
nsgmls:nodtd.sgm:3:15:E: element "title" undefined
nsgmls:nodtd.sgm:4:5:E: element "para" undefined
nsgmls:nodtd.sgm:10:5:E: element "para" undefined
nsgmls:nodtd.sgm:11:15:E: there is no attribute "role"
nsgmls:nodtd.sgm:11:21:E: element "emphasis" undefined
nsgmls:nodtd.sgm:12:9:E: element "emphasis" undefined
nsgmls:nodtd.sgm:12:24:E: element "emphasis" undefined
nsgmls:nodtd.sgm:13:18:E: element "superscript" undefined
nsgmls:nodtd.sgm:14:16:E: element "subscript" undefined
nsgmls:nodtd.sgm:16:5:E: element "para" undefined
```

Other things to look for, if you haven't misspelled the public identifier, are typos in the catalog or failure to specify a catalog that resolves the public identifier that can't be found.

[1] It is often the case that you can correct an error in the document in several ways. The parser suggests one possible fix, but this is not always the right fix. For example, the parser may suggest that you can correct out of context data by adding another element, when in fact it's "obvious" to human eyes that the problem is a missing end tag.

3.2.2 ISO Entity Set Missing

A missing entity set is another example of either a misspelled public identifier, or a missing catalog or catalog entry.

In this case, there's nothing wrong with the document, but the catalog that's been specified is missing the public identifiers for the ISO entity sets:

```
[n:\dbtdg]nsgmls -sv -c examples\errs\cat2 examples\simple.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:53:65:W: cannot generate syst
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:54:8:E: reference to entity "
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:52:0: entity was defined here
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:60:66:W: cannot generate syst
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:61:8:E: reference to entity "
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:59:0: entity was defined here
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:67:60:W: cannot generate syst
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:68:8:E: reference to entity "
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:66:0: entity was defined here
m:\jade\nsgmls.exe:n:/share/sgml/docbook/3.1/dbcent.mod:74:67:W: cannot generate syst
...
```

The ISO entity sets are required by the DocBook DTD, but they are not distributed with it. That's because they aren't maintained by OASIS.[2]

3.2.3 Character Data Not Allowed Here

Out of context character data is frequently caused by a missing start tag, but sometimes it's just the result of typing in the wrong place!

```
[n:\dbtdg] nsgmls -sv -c \share\sgml\catalog examples\errs\badpcdata.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\badpcdata.sgm:9:0:E: character data is not allowed h
```

`Chapter`s aren't allowed to contain character data directly. Here, a wrapper element, such as `Para`, is missing around the sentence between the first two paragraphs.

3.2.4 Misspelled Start Tag

If you spell it wrong, the parser gets confused.

```
[n:\documents\books\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\misspe
ll.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\misspell.sgm:9:5:E: element "PAAR" undefined
m:\jade\nsgmls.exe:examples\errs\misspell.sgm:14:6:E: end tag for element "PARA" whic
m:\jade\nsgmls.exe:examples\errs\misspell.sgm:21:9:E: end tag for "PAAR" omitted, but
m:\jade\nsgmls.exe:examples\errs\misspell.sgm:9:0: start tag was here
```

Luckily, these are pretty easy to spot, unless you accidentally spell the name of another element. In that case, your error might appear to be out of context.

[2] If you need to locate the entity sets, consult http://www.oasis-open.org/cover/topics.html#entities.

3.2.5 Misspelled End Tag

Spelling the end tag wrong is just as confusing.

```
[n:\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\misspell2.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:2:35:E: end tag for element "TITEL" which i
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:3:5:E: document type does not allow element
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:9:5:E: document type does not allow element
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:15:5:E: document type does not allow elemer
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:21:9:E: end tag for "TITLE" omitted, but OM
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:2:9: start tag was here
m:\jade\nsgmls.exe:examples\errs\misspell2.sgm:21:9:E: end tag for "CHAPTER" which is not
```

These are pretty easy to spot as well, but look at how confused the parser became. From the parser's point of view, failure to close the open `Title` element means that all the following elements appear out of context.

3.2.6 Out of Context Start Tag

Sometimes the problem isn't spelling, but placing a tag in the wrong context. When this happens, the parser tries to figure out what it can add to your document to make it valid. Then it proceeds as if it had seen what was added in order to recover from the error seen, which can cause future errors.

```
[n:\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\badstarttag.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\badstarttag.sgm:9:12:E: document type does not allow elem
```

In this example, we probably wanted a `FormalPara`, so that we could have a title on the paragraph. But note that the parser didn't suggest this alternative. The parser only tries to add additional elements, rather than rename elements that it's already seen.

3.2.7 Missing End Tag

Leaving out an end tag is a lot like an out of context start tag. In fact, they're really the same error. The problem is never caused by the missing end tag per se, rather it's caused by the fact that something following it is now out of context.

```
[n:\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\noendtag.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\noendtag.sgm:14:5:E: document type does not allow element
m:\jade\nsgmls.exe:examples\errs\noendtag.sgm:20:9:E: end tag for "PARA" omitted, but OMIT
m:\jade\nsgmls.exe:examples\errs\noendtag.sgm:9:0: start tag was here
```

In this case, the parser figured out that the best thing it could do is end the paragraph.

3.2.8 Bad Entity Reference

If you spell an entity name wrong, the parser will catch it.

```
[n:\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\badent.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\badent.sgm:10:26:E: general entity "xyzzy" not defin
```

More often than not, you'll see this when you misspell a character entity name. For example, this happens when you type `&ldqou;` instead of `“`.

3.2.9 Invalid 8-Bit Character

In XML, the entire range of Unicode characters is available to you, but in SGML, the declaration indicates what characters are valid. The distributed DocBook declaration doesn't allow a bunch of fairly common 8-bit characters.

```
[n:\dbtdg]nsgmls -sv -c \share\sgml\catalog examples\errs\badchar.sgm
m:\jade\nsgmls.exe:I: SP version "1.3.2"
m:\jade\nsgmls.exe:examples\errs\badchar.sgm:11:0:E: non SGML character number 147
m:\jade\nsgmls.exe:examples\errs\badchar.sgm:11:5:E: non SGML character number 148
```

In this example, the Windows code page values for curly left and right quotes have been used, but they aren't in the declared character set. Fix this by converting them to character entities.

You can also fix them by changing the declaration, but if you do that, make sure all your interchange partners are aware of, and have a copy of, the modified declaration. See Appendix F.

3.3 Considering Other Schema Languages

Historically, DTDs were the only way to describe the valid stricture of SGML and XML documents, but that is no longer the case. At the time of this writing (January, 2001), DocBook is experimentally available in three other schema languages:

XML Schema `http://www.w3.org/XML/Schema` The schema language being defined by the W3C `http://www.w3.org/` as the successor to DTDs for describing the structure of XML. XML Schema are likely to become a W3C `http://www.w3.org/` Recommendation in 2001.

RELAX `http://www.xml.gr.jp/relax/` RELAX, the Regular Language description for XML) is a less complex alternative to XML Schemas. The RELAX Core module is defined by ISO in *ISO/IEC DTR 22250-1, Document Description and Processing Languages – Regular Language Description for XML (RELAX) – Part 1: RELAX Core, 2000*. The RELAX Namespaces module is currently under development.

TREX `http://www.thaiopensource.com/trex/` TREX, Tree Regular Expressions for XML, is another less complex alternative to XML Schemas. It is concise, powerful, and datatype neutral.

3.3.1 Parsing and Validation

Before we look closer at these new schema languages, there's one significant difference between DTDs and all of them that we should get out of the way: XML parsers (which may understand DTDs) build an XML information set out of a stream of characters, all of these other schema languages begin with an information set and perform validation on it.

What I mean by that is that an XML parser reads a stream of bytes:

```
"<" "?" "x" "m" "l" " " "v" "e" ...
"<" "!" "D" "O" "C" "T" "Y" "P" "E" " " "b" "o" "o" "k" ...
"<" "b" "o" "o" "k" " " "i" "d" "=" "'" "f" "o" "o" "'" ">"
...
"<" "/" "b" "o" "o" "k" ">"
```

interprets them as a stream of characters (which may change the interpretation of some sequences of bytes) and constructs some representation of the XML document. This representation is the set of all the XML information items encountered: the information set of the document. The W3C http://www.w3.org/ XML Core Working Group http://www.w3.org/XML/ is in the process of defining what an XML Information Set http://www.w3.org/TR/xml-infoset contains.

The other schema languages are defined not in terms of the sequence of characters in the file but in terms of the information set of the XML document. They have to work this way because the XML Recommendation http://www.w3.org/TR/REC-xml says what an XML document is and they all want to work on top of XML.

So what, you might ask? Well, it turns out that this has at least one very significant implication: there's no way for these languages to provide support for entity declarations.

An entity, like "`&ora;`" as a shortcut for "O'Reilly & Associates" or "`é`" as a mnemonic for "é", is a feature of the character stream seen by the XML parser, it doesn't exist in the information set of valid XML documents. More importantly, this means that even if the schema language had a syntax for declaring entities, it wouldn't help the XML parser that needs to know the definitions long before the schema language processor comes into play.

There are a couple of other XML features that are impacted, though not necessarily as significantly: notations and default attribute values. One use for notations is on external entity declarations, and as we've already seen, the schema language is too late to be useful for anything entity related. Default attribute values are also problematic since you would like them to be in the information set produced by the parser so that the schema language sees them.

3.3.2 A Coarse Comparison of Three XML Schema Languages

FIXME: write a short synopsis of how these languages compare.

Chapter 4

Publishing DocBook Documents

Creating and editing SGML/XML documents is usually only half the battle. After you've composed your document, you'll want to publish it. Publishing, for our purposes, means either print or web publishing. For SGML and XML documents, this is usually accomplished with some kind of stylesheet. In the (not too distant) future, you may be able to publish an XML document on the Web by simply putting it online with a stylesheet, but for now you'll probably have to translate your document into HTML.

There are many ways, using both free and commercial tools, to publish SGML documents. In this chapter, we're going to survey a number of possibilities, and then look at just one solution in detail: Jade http://www.jclark.com/jade/ and the Modular DocBook Stylesheets. http://nwalsh.com/docbook/dsssl/ We used jade to produce this book and to produce the online versions on the CD-ROM; it is also being deployed in other projects such as <SGML>&tools;, http://www.sgmltools.org/ which originated with the Linux Documentation Project.

For a brief survey of other tools, see Appendix D.

4.1 A Survey of Stylesheet Languages

Over the years, a number of attempts have been made to produce a standard stylesheet language and, failing that, a large number of proprietary languages have been developed.

FOSIs First, the U.S. Department of Defense, in an attempt to standardize stylesheets across military branches, created the *Output Specification*, which is defined in MIL-PRF-28001C, *Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text*.[1]

Commonly called FOSIs (for Formatting Output Specification Instances), they are supported by a few products including ADEPT Publisher by Arbortext http://www.arbortext.com/ and DL Composer by Datalogics http://www.datalogics.com/.

DSSSL Next, the International Organization for Standardization (ISO) created DSSSL, the Document Style Semantics and Specification Language. Subsets of DSSSL are supported by Jade and a few other tools, but it never achieved widespread support.

CSS The W3C CSS Working Group created CSS as a style attachment language for HTML, and, more recently, XML.

XSL Most recently, the XML effort has identified a standard Extensible Style Language (XSL) as a requirement. The W3C XSL Working Group is currently pursuing that effort.

4.1.1 Stylesheet Examples

By way of comparison, here's an example of each of the standard style languages. In each case, the stylesheet fragment shown contains the rules that reasonably formatted the following paragraph:

[1] See *Formally Published CALS Standards* http://www-cals.itsi.disa.mil/core/formal/fps.htm for more information.

```
<para>
This is an example paragraph. It should be presented in a
reasonable body font. <emphasis>Emphasized</emphasis> words
should be printed in italics. A single level of
<emphasis>Nested <emphasis>emphasis</emphasis> should also
be supported.</emphasis>
</para>
```

4.1.1.1 FOSI stylesheet

FOSIs are SGML documents. The element in the FOSI that controls the presentation of specific elements is the `e-i-c` (element in context) element. A sample FOSI fragment is shown in Example 4.1.1.

Example 4.1.1: A Fragment of a FOSI Stylesheet

4.1.1.2 DSSSL stylesheet

DSSSL stylesheets are written in a Scheme-like language (see Section 4.3.1 later in this chapter). It is the `element` function that controls the presentation of individual elements. See the example in Example 4.1.2.

Example 4.1.2: A Fragment of a DSSSL Stylesheet

4.1.1.3 CSS stylesheet

CSS stylesheets consist of selectors and formatting properties, as shown in Example 4.1.3.

Example 4.1.3: A Fragment of a CSS Stylesheet

4.1.1.4 XSL stylesheet

XSL stylesheets are XML documents, as shown in Example 4.1.4. The element in the XSL stylesheet that controls the presentation of specific elements is the `xsl:template` element.

4.2 Using Jade and DSSSL to Publish DocBook Documents

Jade is a free tool that applies DSSSL `http://www.jclark.com/dsssl/` stylesheets to SGML and XML documents. As distributed, Jade can output RTF, TeX, MIF, and SGML. The SGML backend can be used for SGML to SGML transformations (for example, DocBook to HTML).

A complete set of DSSSL stylesheets for creating print and HTML output from DocBook is included on the CD-ROM. More information about obtaining and installing Jade appears in Appendix A.

4.3 A Brief Introduction to DSSSL

DSSSL is a stylesheet language for both print and online rendering. The acronym stands for *Document Style Semantics and Specification Language*. It is defined by ISO/IEC 10179:1996. For more general information about DSSSL, see the DSSSL Page `http://www.jclark.com/dsssl/`.

Example 4.1.4: A Fragment of an XSL Stylesheet

4.3.1 Scheme

The DSSSL expression language is Scheme, a variant of Lisp. Lisp is a functional programming language with a remarkably regular syntax. Every expression looks like this:

```
(operator arg1 arg2 ... argn )
```

This is called "prefix" syntax because the operator comes before its arguments.

In Scheme, the expression that subtracts 2 from 3, is `(- 3 2)`. And `(+ (- 3 2) (* 2 4))` is 9. While the prefix syntax and the parentheses may take a bit of getting used to, Scheme is not hard to learn, in part because there are no exceptions to the syntax.

4.3.2 DSSSL Stylesheets

A complete DSSSL stylesheet is shown in Example 4.3.1. After only a brief examination of the stylesheet, you'll probably begin to have a feel for how it works. For each element in the document, there is an element rule that describes how you should format that element. The goal of the rest of this chapter is to make it possible for you to read, understand, and even write stylesheets at this level of complexity.

Example 4.3.1: A Complete DSSSL Stylesheet

This stylesheet is capable of formatting simple DocBook documents like the one shown in Example 4.3.2.

Example 4.3.2: A Simple DocBook Document

The result of formatting a simple document with this stylesheet can be seen in Figure 4.1.

We'll take a closer look at this stylesheet after you've learned a little more DSSSL.

4.3.3 DSSSL Stylesheets Are SGML Documents

One of the first things that may strike you about DSSSL stylesheets (aside from all the parentheses), is the fact that the stylesheet itself is an SGML document! This means that you have all the power of SGML documents at your disposal in DSSSL stylesheets. In particular, you can use entities and marked sections to build a modular stylesheet.

In fact, DSSSL stylesheets are defined so that they correspond to a particular architecture. This means that you can change the DTD used by stylesheets within the bounds of the architecture. A complete discussion of document architectures is beyond the scope of this book, but we'll show you one way to take advantage of them in your DSSSL stylesheets in Section 4.3.10 later in the chapter.

4.3.4 DSSSL Processing Model

A DSSSL processor builds a tree out of the source document. Each element in the source document becomes a node in the tree (processing instructions and other constructs become nodes as well). Processing the source tree begins with the root rule and continues until there are no more nodes to process.

4.3.5 Global Variables and Side Effects

There aren't any global variables or side effects. It can be difficult to come to grips with this, especially if you're just starting out.

It is possible to define constants and functions and to create local variables with `let` expressions, but you can't create any global variables or change anything after you've defined it.

4.3.6 DSSSL Expressions

DSSSL has a rich vocabulary of expressions for dealing with all of the intricacies of formatting. Many, but by no means all of them, are supported by Jade. In this introduction, we'll cover only a few of the most common.

4.3.6.1 Element expressions

Element expressions, which define the rules for formatting particular elements, make up the bulk of most DSSSL stylesheets. A simple element rule can be seen in Example 4.3.3. This rule says that a `para` element should be formatted by making a paragraph (see Section 4.3.6.2).

Example 4.3.3: A Simple DSSSL Rule

```
(element para
  (make paragraph
    space-before: 8pt
    (process-children)))
```

An element expression can be made more specific by specifying an element and its ancestors instead of just specifying an element. The rule `(element title ...)` applies to all `Title` elements, but a rule that begins `(element (figure title) ...)` applies only to `Title` elements that are immediate children of `Figure` elements.

If several rules apply, the most specific rule is used.

When a rule is used, the node in the source tree that was matched becomes the "current node" while that element expression is being processed.

4.3.6.2 Make expressions

A make expression specifies the characteristics of a "flow object." Flow objects are abstract representations of content (paragraphs, rules, tables, and so on). The expression:

```
(make paragraph
  font-size: 12pt
  line-spacing: 14pt ...)
```

specifies that the content that goes "here" is to be placed into a paragraph flow object with a font-size of 12pt and a line-spacing of 14pt (all of the unspecified characteristics of the flow object are defaulted in the appropriate way).

They're called flow objects because DSSSL, in its full generality, allows you to specify the characteristics of a sequence of flow objects and a set of areas on the physical page where you can place content. The content of the flow objects is then "poured on to" (or flows in to) the areas on the page(s).

In most cases, it's sufficient to think of the make expressions as constructing the flow objects, but they really only specify the *characteristics* of the flow objects. This detail is apparent in one of the most common and initially confusing pieces of DSSSL jargon: the *sosofo*. Sosofo stands for a "specification of a sequence of flow objects." All this means is that processing a document may result in a nested set of `make` expressions (in other words, the paragraph may contain a table that contains rows that contain cells that contain paragraphs, and so on).

The general form of a `make` expression is:

```
(make flow-object-name
  keyword1: value1
  keyword2: value2
  ...
  keywordn: valuen
  (content-expression))
```

Keyword arguments specify the characteristics of the flow object. The specific characteristics you use depends on the flow object. The *content-expression* can vary; it is usually another make expression or one of the processing expressions.

Some common flow objects in the print stylesheet are:

simple-page-sequence Contains a sequence of pages. The keyword arguments of this flow object let you specify margins, headers and footers, and other page-related characteristics. Print stylesheets should always produce one or more `simple-page-sequence` flow objects.

Nesting `simple-page-sequence` does not work. Characteristics on the inner sequences are ignored.

paragraph A paragraph is used for any block of text. This may include not only paragraphs in the source document, but also titles, the terms in a definition list, glossary entries, and so on. Paragraphs in DSSSL can be nested.

sequence A sequence is a wrapper. It is most frequently used to change inherited characteristics (like font style) of a set of flow objects without introducing other semantics (such as line breaks).

score A score flow object creates underlining, strike-throughs, or overlining.

table A table flow object creates a table of rows and cells.

The HTML stylesheet uses the SGML backend, which has a different selection of flow objects.

element Creates an element. The content of this `make` expression will appear between the start and end tags. The expression:

```
(make element gi: "H1"
      (literal "Title"))
```

produces `<H1>Title</H1>`.

empty-element Creates an empty element that may not have content. The expression:

```
(make empty-element gi: "BR"
     attributes: '(("CLEAR" "ALL")))
```

produces `<BR CLEAR="ALL">`.

sequence Produces no output in of itself as a wrapper, but is still required in DSSSL contexts in which you want to output several flow objects but only one object top-level object may be returned.

entity-ref Inserts an entity reference. The expression:

```
(make entity-ref name: "nbsp")
```

produces ` `.

In both stylesheets, a completely empty flow object is constructed with `(empty-sosofo)`.

4.3.6.3 Selecting data

Extracting parts of the source document can be accomplished with these functions:

(data *nd*) Returns all of the character data from *nd* as a string.

(attribute-string "*attr*" *nd*) Returns the value of the *attr* attribute of *nd*.

(inherited-attribute-string "*attr*" *nd*) Returns the value of the *attr* attribute of *nd*. If that attribute is not specified on *nd*, it searches up the hierarchy for the first ancestor element that does set the attribute, and returns its value.

4.3.6.4 Selecting elements

A common requirement of formatting is the ability to reorder content. In order to do this, you must be able to select other elements in the tree for processing. DSSSL provides a number of functions that select other elements. These functions all return a list of nodes.

(current-node) Returns the current node.

(children *nd*) Returns the children of *nd*.

(descendants *nd*) Returns the descendants of *nd* (the children of *nd* and all their children's children, and so on).

(parent *nd*) Returns the parent of *nd*.

(ancestor "*name*" *nd*) Returns the first ancestor of *nd* named *name*.

(element-with-id "*id*") Returns the element in the document with the ID *id*, if such an element exists.

(select-elements *node-list* "*name*") Returns all of the elements of the *node-list* that have the name *name*. For example, `(select-elements (descendants (current-node)) "para")` returns a list of all the paragraphs that are descendants of the current node.

(empty-node-list) Returns a node list that contains no nodes.

Other functions allow you to manipulate node lists.

(node-list-empty? *nl*) Returns true if (and only if) *nl* is an empty node list.

(node-list-length *nl*) Returns the number of nodes in *nl*.

(node-list-first *nl*) Returns a node list that consists of the single node that is the first node in *nl*.

(node-list-rest *nl*) Returns a node list that contains all of the nodes in *nl* except the first node.

There are many other expressions for manipulating nodes and node lists.

4.3.6.5 Processing expressions

Processing expressions control which elements in the document will be processed and in what order. Processing an element is performed by finding a matching element rule and using that rule.

`(process-children)` Processes all of the children of the current node. In most cases, if no process expression is given, processing the children is the default behavior.

`(process-node-list nl)` Processes each of the elements in `nl`.

4.3.6.6 Define expressions

You can declare your own functions and constants in DSSSL. The general form of a function declaration is:

```
(define (function args)
  function-body)
```

A constant declaration is:

```
(define constant
  constant-function-body)
```

The distinction between constants and functions is that the body of a constant is evaluated when the definition occurs, while functions are evaluated when they are used.

4.3.6.7 Conditionals

In DSSSL, the constant `#t` represents true and `#f` false. There are several ways to test conditions and take action in DSSSL.

`if` The form of an `if` expression is:

```
(if condition
  true-expression
  false-expression)
```

If the condition is true, the *`true-expression`* is evaluated, otherwise the *`false-expression`* is evaluated. You must always provide an expression to be evaluated when the condition is not met. If you want to produce nothing, use `(empty-sosofo)`.

`case` `case` selects from among several alternatives:

```
(case expression
  ((constant1) (expression1)
  ((constant2) (expression2)
  ((constant3) (expression3)
  (else else-expression))
```

The value of the expression is compared against each of the constants in turn and the expression associated with the first matching constant is evaulated.

`cond` `cond` also selects from among several alternatives, but the selection is performed by evaluating each expression:

```
(cond
  ((condition1) (expression1)
  ((condition2) (expression2)
  ((condition3) (expression3)
  (else else-expression))
```

The value of each conditional is calculated in turn. The expression associated with the first condition that is true is evaluated.

Any expression that returns `#f` is false; all other expressions are true. This can be somewhat counterintuitive. In many programming languages, it's common to assume that "empty" things are false (0 is false, a null pointer is false, an empty set is false, for example.) In DSSSL, this isn't the case; note, for example, that an empty node list is not `#f` and is therefore true. To avoid these difficulties, always use functions that return true or false in conditionals. To test for an empty node list, use `(node-list-empty?)`.

4.3.6.8 Let expressions

The way to create local variables in DSSSL is with `(let)`. The general form of a `let` expression is:

```
(let ((var1 expression1)
      (var2 expression2)
      ...
      (varn expressionn))
  let-body)
```

In a `let`; expression, all of the variables are defined "simultaneously." The expression that defines *`var2`* cannot contain any references to any other variables defined in the same `let` expression. A `let*` expression allows variables to refer to each other, but runs slightly slower.

Variables are available only within the *`let-body`*. A common use of `let` is within a `define` expression:

```
(define (cals-rule-default nd)
   (let* ((table (ancestor "table" nd))
          (frame (if (attribute-string "frame" table)
                     (attribute-string "frame" table)
                     "all")))
    (equal? frame "all")))
```

This function creates two local variables `table` and `frame`. `let` returns the value of the last expression in the body, so this function returns true if the frame attribute on the table is `all` or if no frame attribute is present.

4.3.6.9 Loops

DSSSL doesn't have any construct that resembles the "for loop" that occurs in most imperative languages like C and Java. Instead, DSSSL employs a common trick in functional languages for implementing a loop: tail recursion.

Loops in DSSSL use a special form of `let`. This loop counts from 1 to 10:

```
(let loopvar ((count 1))
  (if (> count 10)
    #t
    (loopvar (+ count 1))))
```

dl1 This variable controls the loop. It is declared without an initial value, immediately after the `let` operand. dl2

Any number of additional local variables can be defined after the loop variable, just as they can in any other `let` expression. dl3 If you ever want the loop to end, you have to put some sort of a test in it. dl4 This is the value that will be returned. dl5 Note that you iterate the loop by using the loop variable as if it was a function name. dl6 The arguments to this "function" are the values that you want the local variables declared in (2) to have in the next iteration.

4.3.7 A Closer Look at Example 4.3.1

Example 4.3.1 is a style sheet that contains a style specification. Stylesheets may consist of multiple specifications, as we'll see in Section 4.3.8.3.

The actual DSSSL code goes in the style specification body, within the style specification. Each construction rule processes different elements from the source document.

4.3.7.1 Processing chapters

`Chapter`s are processed by the `chapter` construction rule. Each `Chapter` is formatted as a `simple-page-sequence`. Every print stylesheet should format a document as one or more simple page sequences. Characteristics on the simple page sequence can specify headers and footers as well as margins and other page parameters.

One important note about simple page sequences: they cannot nest. This means that you cannot blindly process divisions (`Part`s, `Reference`) and the elements they contain (`Chapter`s, `RefEntry`s) as simple page sequences. This sometimes involves a little creativity.

4.3.7.2 Processing titles

The `make` expression in the `title` element rule ensures that `Title`s are formatted in large, bold print.

This construction rule applies equally to `Chapter` titles, `Figure` titles, and `Book` titles. It's unlikely that you'd want all of these titles to be presented in the same way, so a more robust stylesheet would have to arrange the processing of titles with more context. This might be achieved in the way that nested `Emphasis` elements are handled in Section 4.3.7.4.

4.3.7.3 Processing paragraphs

`Para` elements are simply formatted as paragraphs.

4.3.7.4 Processing emphasis

Processing `Emphasis` elements is made a little more interesting because we want to consider an attribute value and the possibility that `Emphasis` elements can be nested.

In the simple case, in which we're processing an `Emphasis` element that is not nested, we begin by testing the value of the role attribute. If the content of that attribute is the string `strong`, it is formatted in bold; otherwise, it is formatted in italic.

The nested case is handled by the `(emphasis emphasis)` rule. This rule simply formats the content using an upright (nonitalic) font. This rule, like the rule for `Title`s, is not robust. `Emphasis` nested inside `strong` `Emphasis` won't be distinguished, for example, and nestings more than two elements deep will be handled just as nestings that are two deep.

4.3.7.5 Processing subscripts and superscripts

Processing `Subscript` and `Superscript` elements is really handled by the `super-sub-script` function. There are several interesting things about this function:

The `plus-or-minus` argument You might ordinarily think of passing a keyword or boolean argument to the `super-sub-script` function to indicate whether subscripts or superscripts are desired. But with Scheme, it's possible to pass the actual function as an argument!

Note that in the element construction rules for `Superscript` and `Subscript`, we pass the actual functions `+` and `-`. In the body of `super-sub-script`, we use the `plus-or-minus` argument as a function name (it appears immediately after an open parenthesis).

The optional argument `optional` arguments are indicated by `#!optional` in the function declaration. Any number of `optional` arguments may be given, but each must specify a default value. This is accomplished by listing each argument and default value (an expression) as a pair.

In `super-sub-script`, the optional argument `sosofo` is initialized to `process-children`. This means that at the point where the function is *called*, `process-children` is evaluated and the resulting `sosofo` is passed to the function.

Use of inherited characteristics It is possible to use the "current" value of an inherited characteristic to calculate a new value. Using this technique, superscripts and subscripts will be presented at 80 percent of the current font size.

4.3.8 Customizing the Stylesheets

The best way to customize the stylesheets is to write your own "driver" file; this is a stylesheet that contains your local modifications and then includes the appropriate stylesheet from the standard distribution by reference. This allows you to make local changes and extensions without modifying the distributed files, which makes upgrading to the next release much simpler.

4.3.8.1 Writing Your Own Driver

A basic driver file looks like this:

```
<!DOCTYPE style-sheet PUBLIC "-//James Clark//DTD DSSSL Style Sheet//EN" [
<!ENTITY dbstyle PUBLIC "-//Norman Walsh//DOCUMENT DocBook Print Stylesheet//EN" CDATA DSS
]>

<style-sheet>
<style-specification use="docbook">
<style-specification-body>

;; your changes go here...

</style-specification-body>
</style-specification>
<external-specification id="docbook" document="dbstyle">
</style-sheet>
```

There are two public identifiers associated with the Modular DocBook Stylesheets:

- `-//Norman Walsh//DOCUMENT DocBook Print Stylesheet//EN`
- `-//Norman Walsh//DOCUMENT DocBook HTML Stylesheet//EN`

The former selects the print stylesheet and the latter selects the HTML stylesheet. There is an SGML Open catalog file in the distribution that maps these public identifiers to the stylesheet files.

You can add your own definitions, or redefinitions, of stylesheet rules and parameters so that

```
;; your changes go here...
```

occurs in the previous example.

For a concrete example of a driver file, see `plain.dsl` in the `docbook/print` directory in the stylesheet distribution (or on the CD-ROM). This is a customization of the print stylesheet, which turns off title page and TOC generation.

4.3.8.2 Changing the Localization

As distributed, the stylesheets use English for all generated text, but other localization files are also provided. At the time of this writing, the stylesheets support Catalan, Czech, Danish, Dutch, English, Finnish, French, German, Greek, Italian, Japanese, Norwegian, Polish, Portuguese, Portuguese (Brazil), Romanian, Russian, Slovak, Spanish, and Swedish. (If you can write a localization for another language, *please* contribute it.)

There are two ways to switch languages: by specifying a lang attribute, or by changing the default language in a customization.

Using the lang attribute One of the DocBook common attributes is lang. If you specify a language, the DocBook stylesheets will use that language (and all its descendants, if no other language is specified) for generated text within that element.

Table **??** summarizes the language codes for the supported languages.[2]

The following chapter uses text generated in French:

[2] Language codes should conform to IETF RFC 3066.

```
<chapter lang="fr"><title>Bêtises</title>
<para>Pierre qui roule n’amasse pas de mousse.</para>
</chapter>
```

Language Code	Language
af	Afrikaans
ca	Catalan
cs	Czech
da	Danish
de	German
el	Greek
en	English
es	Spanish
et	Estonian
eu	Basque
fi	Finnish
fr	French
he	Hebrew
hu	Hungarian
id	Indonesian
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
nn	Nyorsk
no	Norwegian
pl	Polish
pt	Portuguese
pt-br	Portuguese (Brazil)
ro	Romanian
ru	Russian
sk	Slovak
sl	Slovenian
sv	Swedish
th	Thai
tr	Turkish
uk	Ukranian
xh	Xhosa
zh-cn	Chinese (Continental)
zh-tw	Chinese (Traditional)

Changing the default language If no lang attribute is specified, the default language is used. You can change the default language with a driver.

In the driver, define the default language. Table **??** summarizes the language codes for the supported languages. The following driver makes German the default language:

```
<!DOCTYPE style-sheet PUBLIC "-//James Clark//DTD DSSSL Style Sheet//EN" [
<!ENTITY dbstyle PUBLIC "-//Norman Walsh//DOCUMENT DocBook Print Stylesheet//EN" CDATA DSS
]>

<style-sheet>
<style-specification use="docbook">
<style-specification-body>

(define %default-language% "dege")

</style-specification-body>
</style-specification>
<external-specification id="docbook" document="dbstyle">
</style-sheet>
```

There are two other settings that can be changed only in a driver. Both of these settings are turned off in the

distributed stylesheet:

%gentext-language% If a language code is specified in %gentext-language%, then that language will be used for all generated text, regardless of any lang attribute settings in the document.

%gentext-use-xref-language% If turned on (defined as #t), then the stylesheets will generate the text associated with a cross reference using the language of the target, not the current language. Consider the following book:

```
<book><title>A Test Book</title>
<preface>
<para>There are three chapters in this book: <xref linkend="c1">,
<xref linkend="c2">, and <xref linkend="c3">.
</para>
</preface>
<chapter lang="usen"><title>English</title> ... </chapter>
<chapter lang="fr"><title>French</title> ... </chapter>
<chapter lang="dege"><title>Deutsch</title> ... </chapter>
</book>
```

The standard stylesheets render the Preface as something like this:

> There are three chapters in this book: Chapter 1, Chapter 2, and Chapter 3.

With %gentext-use-xref-language% turned on, it would render like this:

> There are are three chapters in this book: Chapter 1, Chapitre 2, and Kapitel 3.

4.3.8.3 A Single Stylesheet for Both Print and HTML

A DSSSL stylesheet consists of one or more "style specifications." Using more than one style specification allows you to build a single stylesheet file that can format with either the print or SGML backends. Example 4.3.4 shows a stylesheet with two style specifications.

Example 4.3.4: both.dsl: A Stylesheet with Two Style Specifications

```
<inlinegraphic format="linespecific" fileref="examples/dsssl.twostyles"/>
```

Once you have stylesheets with more than one style specification, you have to be able to indicate which style specification you want to use. In Jade, you indicate this by providing the ID of the style specification after the stylesheet filename, separated with a hash mark: #.

Using the code from Example 4.3.4, you can format a document using the print stylesheet by running:

```
jade -t rtf -d both.dsl#print file.sgm
```

and using the HTML stylesheet by running:

```
jade -t sgml -d both.dsl#html file.sgm
```

4.3.9 Dealing with Multiple Declarations

The DocBook SGML DTD and the DocBook DSSSL Stylesheets happen to use the same SGML declaration. This makes it very easy to run Jade with DocBook. However, you may sometimes wish to use Jade with other document types, for example the DocBook XML DTD, which has a different declaration. There are a couple of ways to do this.

4.3.9.1 Pass the Declaration Explicitly

If your stylesheets parse fine with the default declaration, but you want to use an alternate declaration with a particular document, just pass the declaration on the command line:

```
jade options the-declaration the-document
```

Note that there's no option required before the declaration; it simply occurs before the first filename. Jade concatenates all of the files that you give it together, and parses them as if they were one document.

4.3.9.2 Use the Catalogs

The other way to fix this is with a little catalog trickery.

First, note that Jade always looks in the file called `catalog` in the same directory as the document that it is loading, and uses settings in that file in preference to settings in other catalogs.

With this fact, we can employ the following trick:

- Put a `catalog` file in the directory that contains your stylesheets, which contain an `SGMLDECL` directive. Jade understands the directive, which points to the SGML declaration that you should use when parsing the stylesheets. For the DocBook stylesheets, the DocBook declaration works fine.

- In the directory that contains the document you want to process, create a `catalog` file that contains an `SGMLDECL` directive that points to the SGML declaration that should be used when parsing the document.

There's no easy way to have both the stylesheet and the document in the same directory if they must be processed with different declarations. But this is usually not too inconvenient.

4.3.10 The DSSSL Architecture

The concept of an architecture was promoted by HyTime. In some ways, it takes the standard SGML/XML notions of the role of elements and attributes and inverts them. Instead of relying on the name of an element to assign its primary semantics, it uses the values of a small set of fixed attributes.

While this may be counterintuitive initially, it has an interesting benefit. An architecture-aware processor can work transparently with many different DTDs. A small example will help illustrate this point.

NOTE

The following example demonstrates the concept behind architectures, but for the sake of simplicity, it does not properly implement an architecture as defined in HyTime.

Imagine that you wrote an application that can read an SGML/XML document containing a letter (conforming to some letter DTD), and automatically print an envelope for the letter. It's easy to envision how this works. The application reads the content of the letter, extracts the address and return address elements from the source, and uses them to generate an envelope:

```
<?xml version='1.0'>
<!DOCTYPE letter "/share/sgml/letter/letter.dtd" [
<!ENTITY myaddress "/share/sgml/entities/myaddress.xml">
]>
<letter>
<returnaddress>&myaddress;</returnaddress>
<address>
<name>Leonard Muellner</name>
<company>O'Reilly &amp; Associates</company>
<street>90 Sherman Street</street>
<city>Cambridge</city><state>MA</state><zip>02140</zip>
</address>
<body>
<salutation>Hi Lenny</salutation>
...
</body>
```

The processor extracts the `Returnaddress` and `Address` elements and their children and prints the envelope accordingly.

Now suppose that a colleague from payroll comes by and asks you to adapt the application to print envelopes for mailing checks, using the information in the payroll database, which has a different DTD. And a week later, someone from sales comes by and asks if you can modify the application to use the contact information DTD. After a while, you would have 11 versions of this program to maintain.

Suppose that instead of using the actual element names to locate the addresses in the documents, you asked each person to add a few attributes to their DTD. By forcing the attributes to have fixed values, they'd automatically be present in each document, but authors would never have to worry about them.

For example, the address part of the letter DTD might look like this:

```

```

Effectively, each address in a letter would look like this:

```

```

In practice, the author would not include the ADDRESS attributes; they are automatically provided by the DTD because they are `#FIXED`.[3]

Now the address portion of the payroll DTD might look like this:

```

```

The employee records will look like this:

```

```

Your application no longer cares about the actual element names. It simply looks for the elements with the correct attributes and uses them. This is the power of an architecture: it provides a level of abstraction that processing applications can use to their advantage. In practice, architectural forms are a bit more complex to set up because they have facilities for dealing with attribute name conflicts, among other things.

Why have we told you all this? Because DSSSL is an architecture. This means you can modify the stylesheet DTD and still run your stylesheets through Jade.

Consider the case presented earlier in Example 4.3.4. In order to use this stylesheet, you must specify three things: the backend you want to use, the stylesheet you want to use, and the style specification you want to use. If you mismatch any of the parameters, you'll get the wrong results. In practice, the problem is compounded further:

- Some stylesheets support several backends (RTF, TeX, and SGML).
- Some stylesheets support only some backends (RTF and SGML, but not TeX or MIF).
- Some stylesheets support multiple outputs using the same backend (several kinds of HTML output, for example, using the SGML backend: HTML, HTMLHelp, JavaHelp, and so on).

[3]The use of uppercase names here is intentional. These are not attributes that an author is ever expected to type. In XML, which is case-sensitive, using uppercase for things like this reduces the likelihood of collision with "real" attribute names in the DTD.

- If you have complex stylesheets, some backends may require additional options to define parameter entities or stylesheet options.

None of this complexity is really necessary, after all, the options don't change—you just have to use the correct combinations. The mental model is really something like this: "I want a certain kind of output, TeX say, so I have to use this combination of parameters."

You can summarize this information in a table to help keep track of it:

Desired Output	Backend	Style specification	Options	Supported?
rtf	rtf	print	-V rtf-backend	yes
tex	tex	print	-V tex-backend -i tex	yes
html	sgml	htmlweb	-i html	yes
javahelp	sgml	help	-i help	yes
htmlhelp				no

Putting this information in a table will help you keep track of it, but it's not the best solution. The ideal solution is to keep this information on your system, and let the software figure it all out. You'd like to be able to run a command, tell it what output you want from what stylesheet, what file you want to process, and then let it figure everything else out. For example:

```
format html mybook.dsl mydoc.sgm
```

One way to do this is to put the configuration data in a separate file, and have the **format** command load it out of this other file. The disadvantage of this solution is that it introduces another file that you have to maintain and it's independent from the stylesheet so it isn't easy to keep it up-to-date.

In the DSSSL case, a better alternative is to modify the stylesheet DTD so you can store the configuration data *in the stylesheet*. Using this alternate DTD, your `mybook.dsl` stylesheets might look like this:

```

```

In this example, the stylesheet has been annotated with a title, a list of the public IDs to which it is applicable, and a table that provides information about the output formats that it supports.

Using this information, the **format** command can get all the information it needs to construct the appropriate call to Jade. To make HTML from `myfile.sgm`, **format** would run the following:

```
jade -t sgml -d mybook.dsl#htmlweb -i html myfile.sgm
```

The additional information, titles and public IDs, can be used as part of a GUI interface to simplify the selection of stylesheets for an author.

The complete annotated stylesheet DTD, and an example of the **format** command script, are provided on the CD-ROM.

4.4 A Brief Introduction to XSL

4.4.1 Using XSL tools to publish DocBook documents

There is a growing list of tools to process DocBook documents using XSL stylesheets. Each tool implements parts or all of the XSL standard, which actually has several components:

Extensible Stylesheet Language (XSL) A language for expressing stylesheets written in XML. It includes the formatting object language, but refers to separate documents for the transformation language and the path language.

XSL Transformation (XSLT) The part of XSL for transforming XML documents into other XML documents, HTML, or text. It can be used to rearrange the content and generate new content.

XML Path Language (XPath) A language for addressing parts of an XML document. It is used to find the parts of your document to apply different styles to. All XSL processors use this component.

To publish HTML from your XML documents, you just need an XSLT engine. To get to print, you need an XSLT engine to produce formatting objects (FO), which then must be processed with a formatting object processor to produce PostScript or PDF output.

James Clark's XT was the first useful XSLT engine, and it is still in wide use. It is written in Java, so it runs on many platforms, and it is free (`http://www.jclark.com`). XT comes with James Clark's nonvalidating parser XP, but you can substitute a different Java parser. Here is a simple example of using XT from the Unix command line to produce HTML: You'll need to alter your *`CLASSPATH`* environment variable to include the path to where you put the `.jar` files from the XT distribution.

```
CLASSPATH=xt.jar:xp.jar:sax.jar
export CLASSPATH
java  com.jclark.xsl.sax.Driver filename.xml docbook/html/docbook.xsl > output.html
```

If you replace the HTML stylesheet with a formatting object stylesheet, XT will produce a formatting object file. Then you can convert that to PDF using FOP, a formatting object processor available for free from the Apache XML Project (`http://xml.apache.org`). Here is an example of that two stage processing:

```
CLASSPATH=xt.jar:xp.jar:sax.jar:fop.jar
export CLASSPATH
java  com.jclark.xsl.sax.Driver filename.xml docbook/fo/docbook.xsl > output.fo
java  org.apache.fop.apps.CommandLine output.fo output.pdf
```

As of this writing, some other XSLT processors to choose from include:

- 4XSLT, written in Python, from FourThought LLC (`http://www.fourthought.com`)
- Sablotron, written in C++, from Ginger Alliance (`http://www.gingerall.com`)
- Saxon, written in Java, from Michael Kay (`http://users.iclway.co.uk/mhkay/saxon`)
- Xalan, written in Java, from the Apache XML Project (`http://xml.apache.org`)
- XML::XSLT,written in Perl, from Geert Josten and Egon Willighagen (`http://www.cpan.org`)

For print output, these additional tools are available for processing formatting objects:

- XEP (written in Java) from RenderX (`http://www.renderx.com`).
- PassiveTeX from Sebastian Rahtz (`http://users.ox.ac.uk/~rahtz/passivetex/`).

4.4.2 A brief introduction to XSL

XSL is both a transformation language and a formatting language. The XSLT transformation part lets you scan through a document's structure and rearrange its content any way you like. You can write out the content using a different set of XML tags, and generate text as needed. For example, you can scan through a document to locate all headings and then insert a generated table of contents at the beginning of the document, at the same time writing out the content marked up as HTML. XSL is also a rich formatting language, letting you apply typesetting controls to all components of your output. With a good formatting backend, it is capable of producing high quality printed pages.

An XSL stylesheet is written using XML syntax, and is itself a well-formed XML document. That makes the basic syntax familiar, and enables an XML processor to check for basic syntax errors. The stylesheet instructions use special element names, which typically begin with `xsl:` to distinguish them from any XML tags you want to appear in the output. The XSL namespace is identified at the top of the stylesheet file. As with other XML, any XSL elements that are not empty will require a closing tag. And some XSL elements have specific attributes that control their behavior. It helps to keep a good XSL reference book handy.

Here is an example of a simple XSL stylesheet applied to a simple XML file to generate HTML output.

Example 4.4.1: Simple XML file

```
<?xml version="1.0"?>
<document>
<title>Using a mouse</title>
<para>It's easy to use a mouse. Just roll it
around and click the buttons.</para>
</document>
```

Example 4.4.2: Simple XSL stylesheet

```
<?xml version='1.0'?>
<xsl:stylesheet
          xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version='1.0'>
<xsl:output method="html"/>

<xsl:template match="document">
  <HTML><HEAD><TITLE>
    <xsl:value-of select="./title"/>
  </TITLE>
  </HEAD>
  <BODY>
    <xsl:apply-templates/>
  </BODY>
  </HTML>
</xsl:template>

<xsl:template match="title">
  <H1><xsl:apply-templates/></H1>
</xsl:template>

<xsl:template match="para">
  <P><xsl:apply-templates/></P>
</xsl:template>

</xsl:stylesheet>
```

4.4.3 XSL processing model

XSL is a template language, not a procedural language. That means a stylesheet specifies a sample of the output, not a sequence of programming steps to generate it. A stylesheet consists of a mixture of output samples with instructions of what to put in each sample. Each bit of output sample and instructions is called a *template*.

In general, you write a template for each element type in your document. That lets you concentrate on handling just one element at a time, and keeps a stylesheet modular. The power of XSL comes from processing the templates recursively. That is, each template handles the processing of its own element, and then calls other templates to process its children, and so on. Since an XML document is always a single root element at the top level that contains all of the nested descendent elements, the XSL templates also start at the top and work their way down through the hierarchy of elements.

Take the DocBook `<para>` paragraph element as an example. To convert this to HTML, you want to wrap the paragraph content with the HTML tags «p» and `</</p>`». But a DocBook `<para>` can contain any number of in-line DocBook elements marking up the text. Fortunately, you can let other templates take care of those elements, so your XSL template for `<para>` can be quite simple:

```
<xsl:template match="para">
  <p>
    <xsl:apply-templates/>
  </p>
```

Example 4.4.3: HTML output

```
<HTML>
<HEAD>
<TITLE>Using a mouse</TITLE>
</HEAD>
<BODY>
<H1>Using a mouse</H1>
<P>It's easy to use a mouse. Just roll it
around and click the buttons.</P>
</BODY>
</HTML>
```

```
</xsl:template>
```

The <xsl:template> element starts a new template, and its match attribute indicates where to apply the template, in this case to any <para> elements. The template says to output a literal «p» string and then execute the <xsl:apply-templates/> instruction. This tells the XSL processor to look among all the templates in the stylesheet for any that should be applied to the content of the paragraph. If each template in the stylesheet includes an <xsl:apply-templates/> instruction, then all descendents will eventually be processed. When it is through recursively applying templates to the paragraph content, it outputs the «</p» closing tag.

4.4.3.1 Context is important

Since you aren't writing a linear procedure to process your document, the context of where and how to apply each modular template is important. The match attribute of <xsl:template> provides that context for most templates. There is an entire expression language, XPath, for identifying what parts of your document should be handled by each template. The simplest context is just an element name, as in the example above. But you can also specify elements as children of other elements, elements with certain attribute values, the first or last elements in a sequence, and so on. Here is how the DocBook <formalpara> element is handled:

```
<xsl:template match="formalpara">
  <p>
    <xsl:apply-templates/>
  </p>
</xsl:template>

<xsl:template match="formalpara/title">
  <b><xsl:apply-templates/></b>
  <xsl:text> </xsl:text>
</xsl:template>

<xsl:template match="formalpara/para">
  <xsl:apply-templates/>
</xsl:template>
```

There are three templates defined, one for the <formalpara> element itself, and one for each of its children elements. The match attribute value formalpara/title in the second template is an XPath expression indicating a <title> element that is an immediate child of a <formalpara> element. This distinguishes such titles from other <title> elements used in DocBook. XPath expressions are the key to controlling how your templates are applied.

In general, the XSL processor has internal rules that apply templates that are more specific before templates that are less specific. That lets you control the details, but also provides a fallback mechanism to a less specific template when you don't supply the full context for every combination of elements. This feature is illustrated by the third template, for formalpara/para. By including this template, the stylesheet processes a <para>

within <formalpara> in a special way, in this case by not outputting the HTML <p> tags already output by its parent. If this template had not been included, then the processor would have fallen back to the template specified by match="para" described above, which would have output a second set of <p> tags.

You can also control template context with XSL *modes*, which are used extensively in the DocBook stylesheets. Modes let you process the same input more than once in different ways. A mode attribute in an <xsl:template> definition adds a specific mode name to that template. When the same mode name is used in <xsl:apply-templates/>, it acts as a filter to narrow the selection of templates to only those selected by the match expression *and* that have that mode name. This lets you define two different templates for the same element match that are applied under different contexts. For example, there are two templates defined for DocBook <listitem> elements:

```
<xsl:template match="listitem">
  <li><xsl:apply-templates/></li>
</xsl:template>

<xsl:template match="listitem" mode="xref">
  <xsl:number format="1"/>
</xsl:template>
```

The first template is for the normal list item context where you want to output the HTML <li> tags. The second template is called with <xsl:apply-templates select="$target" mode="xref"/> in the context of processing <xref> elements. In this case the select attribute locates the ID of the specific list item and the mode attribute selects the second template, whose effect is to output its item number when it is in an ordered list. Because there are many such special needs when processing <xref> elements, it is convenient to define a mode name xref to handle them all. Keep in mind that mode settings do *not* automatically get passed down to other templates through <xsl:apply-templates/>.

4.4.3.2 Programming features

Although XSL is template-driven, it also has some features of traditional programming languages. Here are some examples from the DocBook stylesheets.

```
Assign a value to a variable:
<xsl:variable name="refelem" select="name($target)"/>

If statement:
<xsl:if test="$show.comments">
    <i><xsl:call-template name="inline.charseq"/></i>
</xsl:if>

Case statement:
<xsl:choose>
    <xsl:when test="@columns">
        <xsl:value-of select="@columns"/>
    </xsl:when>
    <xsl:otherwise>1</xsl:otherwise>
</xsl:choose>

Call a template by name like a subroutine, passing parameter values and accepting a return
<xsl:call-template name="xref.xreflabel">
   <xsl:with-param name="target" select="$target"/>
</xsl:call-template>
```

However, you can't always use these constructs as you do in other programming languages. Variables in particular have very different behavior.

Using variables and parameters XSL provides two elements that let you assign a value to a name: <xsl:variable> and <xsl:param>. These share the same name space and syntax for assigning names and values. Both can be referred to using the $name syntax. The main difference between these two elements is that a param's value acts

as a default value that can be overridden when a template is called using a `<xsl:with-param>` element as in the last example above.

Here are two examples from DocBook:

```
<xsl:param name="cols">1</xsl:param>
<xsl:variable name="segnum" select="position()"/>
```

In both elements, the name of the parameter or variable is specified with the name attribute. So the name of the `param` here is `cols` and the name of the `variable` is `segnum`. The value of either can be supplied in two ways. The value of the first example is the text node "1" and is supplied as the content of the element. The value of the second example is supplied as the result of the expression in its select attribute, and the element itself has no content.

The feature of XSL variables that is odd to new users is that once you assign a value to a variable, you cannot assign a new value within the same scope. Doing so will generate an error. So variables are not used as dynamic storage bins they way they are in other languages. They hold a fixed value within their scope of application, and then disappear when the scope is exited. This feature is a result of the design of XSL, which is template-driven and not procedural. This means there is no definite order of processing, so you can't rely on the values of changing variables. To use variables in XSL, you need to understand how their scope is defined.

Variables defined outside of all templates are considered global variables, and they are readable within all templates. The value of a global variable is fixed, and its global value can't be altered from within any template. However, a template can create a local variable of the same name and give it a different value. That local value remains in effect only within the scope of the local variable.

Variables defined within a template remain in effect only within their permitted scope, which is defined as all following siblings and their descendants. To understand such a scope, you have to remember that XSL instructions are true XML elements that are embedded in an XML family hierarchy of XSL elements, often referred to as parents, children, siblings, ancestors and descendants. Taking the family analogy a step further, think of a variable assignment as a piece of advice that you are allowed to give to certain family members. You can give your advice only to your younger siblings (those that follow you) and their descendents. Your older siblings won't listen, neither will your parents or any of your ancestors. To stretch the analogy a bit, it is an error to try to give different advice under the same name to the same group of listeners (in other words, to redefine the variable). Keep in mind that this family is not the elements of your document, but just the XSL instructions in your stylesheet. To help you keep track of such scopes in hand-written stylesheets, it helps to indent nested XSL elements. Here is an edited snippet from the DocBook stylesheet file `pi.xsl` that illustrates different scopes for two variables:

```
 1 <xsl:template name="dbhtml-attribute">
 2 ...
 3    <xsl:choose>
 4       <xsl:when test="$count>count($pis)">
 5          <!-- not found -->
 6       </xsl:when>
 7       <xsl:otherwise>
 8          <xsl:variable name="pi">
 9             <xsl:value-of select="$pis[$count]"/>
10          </xsl:variable>
11          <xsl:choose>
12             <xsl:when test="contains($pi,concat($attribute, '='))">
13                <xsl:variable name="rest" select="substring-after($pi,concat($attrib
14                <xsl:variable name="quote" select="substring($rest,1,1)"/>
15                <xsl:value-of select="substring-before(substring($rest,2),$quote)"/>
16             </xsl:when>
17             <xsl:otherwise>
18             ...
19             </xsl:otherwise>
20          </xsl:choose>
21       </xsl:otherwise>
22    </xsl:choose>
23 </xsl:template>
```

The scope of the variable `pi` begins on line 8 where it is defined in this template, and ends on line 20 when its last sibling ends.[4] The scope of the variable `rest` begins on line 13 and ends on line 15. Fortunately, line 15 outputs an expression using the value before it goes out of scope.

What happens when an `<xsl:apply-templates/>` element is used within the scope of a local variable? Do the templates that are applied to the document children get the variable? The answer is no. The templates that are applied are not actually within the scope of the variable. They exist elsewhere in the stylesheet and are not following siblings or their descendants.

To pass a value to another template, you pass a parameter using the `<xsl:with-param>` element. This parameter passing is usually done with calls to a specific named template using `<xsl:call-template>`, although it works with `<xsl:apply-templates>` too. That's because the called template must be expecting the parameter by defining it using a `<xsl:param>` element with the same parameter name. Any passed parameters whose names are not defined in the called template are ignored.

Here is an example of parameter passing from `docbook.xsl`:

```
<xsl:call-template name="head.content">
   <xsl:with-param name="node" select="$doc"/>
</xsl:call-template>
```

Here a template named `head.content` is being called and passed a parameter named *`node`* whose content is the value of the `$doc` variable in the current context. The top of that template looks like this:

```
<xsl:template name="head.content">
   <xsl:param name="node" select="."/>
```

The template is expecting the parameter because it has a `<xsl:param>` defined with the same name. The value in this definition is the default value. This would be the parameter value used in the template if the template was called without passing that parameter.

4.4.3.3 Generating HTML output.

You generate HTML from your DocBook XML files by applying the HTML version of the stylesheets. This is done by using the HTML driver file `docbook/html/docbook.xsl` as your stylesheet. That is the master stylesheet file that uses `<xsl:include>` to pull in the component files it needs to assemble a complete stylesheet for producing HTML.

The way the DocBook stylesheet generates HTML is to apply templates that output a mix of text content and HTML elements. Starting at the top level in the main file `docbook.xsl`:

```
<xsl:template match="/">
  <xsl:variable name="doc" select="*[1]"/>
  <html>
  <head>
    <xsl:call-template name="head.content">
      <xsl:with-param name="node" select="$doc"/>
    </xsl:call-template>
  </head>
  <body>
    <xsl:apply-templates/>
  </body>
  </html>
</xsl:template>
```

This template matches the root element of your input document, and starts the process of recursively applying templates. It first defines a variable named `doc` and then outputs two literal HTML elements `<html>` and

[4] Technically, the scope extends to the end tag of the parent of the `<xsl:variable>` element. That is effectively the last sibling.

<head>. Then it calls a named template head.content to process the content of the HTML <head>, closes the <head> and starts the <body>. There it uses «xsl:apply-templates/>/> to recursively process the entire input document. Then it just closes out the HTML file.

Simple HTML elements can generated as literal elements as shown here. But if the HTML being output depends on the context, you need something more powerful to select the element name and possibly add attributes and their values. Here is a fragment from sections.xsl that shows how a heading tag is generated using the <xsl:element> and <xsl:attribute> elements:

```
 1 <xsl:element name="h{$level}">
 2   <xsl:attribute name="class">title</xsl:attribute>
 3   <xsl:if test="$level<3">
 4     <xsl:attribute name="style">clear: all</xsl:attribute>
 5   </xsl:if>
 6   <a>
 7     <xsl:attribute name="name">
 8       <xsl:call-template name="object.id"/>
 9     </xsl:attribute>
10     <b><xsl:copy-of select="$title"/></b>
11   </a>
12 </xsl:element>
```

This whole example is generating a single HTML heading element. Line 1 begins the HTML element definition by identifying the name of the element. In this case, the name is an expression that includes the variable *$level* passed as a parameter to this template. Thus a single template can generate <h1>, <h2>, etc. depending on the context in which it is called. Line 2 defines a class="title" attribute that is added to this element. Lines 3 to 5 add a style="clear all" attribute, but only if the heading level is less than 3. Line 6 opens an <a> anchor element. Although this looks like a literal output string, it is actually modified by lines 7 to 9 that insert the name attribute into the <a> element. This illustrates that XSL is managing output elements as active element nodes, not just text strings. Line 10 outputs the text of the heading title, also passed as a parameter to the template, enclosed in HTML boldface tags. Line 11 closes the anchor tag with the literal </</a» syntax, while line 12 closes the heading tag by closing the element definition. Since the actual element name is a variable, it couldn't use the literal syntax.

As you follow the sequence of nested templates processing elements, you might be wondering how the ordinary text of your input document gets to the output. In the file docbook.xsl you will find this template that handles any text not processed by any other template:

```
<xsl:template match="text()">
  <xsl:value-of select="."/>
</xsl:template>
```

This template's body consists of the "value" of the text node, which is just its text. In general, all XSL processors have some built-in templates to handle any content for which your stylesheet doesn't supply a matching template. This template serves the same function but appears explicitly in the stylesheet.

4.4.3.4 Generating formatting objects.

You generate formatting objects from your DocBook XML files by applying the fo version of the stylesheets. This is done by using the fo driver file docbook/fo/docbook.xsl as your stylesheet. That is the master stylesheet file that uses <xsl:include> to pull in the component files it needs to assemble a complete stylesheet for producing formatting objects. Generating a formatting objects file is only half the process of producing typeset output. You also need a formatting object processor such as the Apache XML Project's FOP as described in an earlier section.

The DocBook fo stylesheet works in a similar manner to the HTML stylesheet. Instead of outputting HTML tags, it outputs text marked up with <fo:*something*> tags. For example, to indicate that some text should be kept in-line and typeset with a monospace font, it might look like this:

```
<fo:inline-sequence font-family="monospace">/usr/man</fo:inline-sequence>
```

The templates in `docbook/fo/inline.xsl` that produce this output for a DocBook `<filename>` element look like this:

```
<xsl:template match="filename">
  <xsl:call-template name="inline.monoseq"/>
</xsl:template>

<xsl:template name="inline.monoseq">
  <xsl:param name="content">
    <xsl:apply-templates/>
  </xsl:param>
  <fo:inline-sequence font-family="monospace">
    <xsl:copy-of select="$content"/>
  </fo:inline-sequence>
</xsl:template>
```

There are dozens of fo tags and attributes specified in the XSL standard. It is beyond the scope of this document to cover how all of them are used in the DocBook stylesheets. Fortunately, this is only an intermediate format that you probably won't have to deal with very much directly unless you are writing your own stylesheets.

4.4.4 Customizing DocBook XSL stylesheets

The DocBook XSL stylesheets are written in a modular fashion. Each of the HTML and FO stylesheets starts with a driver file that assembles a collection of component files into a complete stylesheet. This modular design puts similar things together into smaller files that are easier to write and maintain than one big stylesheet. The modular stylesheet files are distributed among four directories:

common/ contains code common to both stylesheets, including localization data

fo/ a stylesheet that produces XSL FO result trees

html/ a stylesheet that produces HTML/XHTML result trees

lib/ contains schema-independent functions

The driver files for each of HTML and FO stylesheets are `html/docbook.xsl` and `fo/docbook.xsl`, respectively. A driver file consists mostly of a bunch of `<xsl:include>` instructions to pull in the component templates, and then defines some top-level templates. For example:

```
<xsl:include href="../VERSION"/>
<xsl:include href="../lib/lib.xsl"/>
<xsl:include href="../common/l10n.xsl"/>
<xsl:include href="../common/common.xsl"/>
<xsl:include href="autotoc.xsl"/>
<xsl:include href="lists.xsl"/>
<xsl:include href="callout.xsl"/>
...
<xsl:include href="param.xsl"/>
<xsl:include href="pi.xsl"/>
```

The first four modules are shared with the FO stylesheet and are referenced using relative pathnames to the common directories. Then the long list of component stylesheets starts. Pathnames in include statements are always taken to be relative to the including file. Each included file must be a valid XSL stylesheet, which means its root element must be `<xsl:stylesheet>`.

4.4.4.1 Stylesheet inclusion vs. importing

XSL actually provides two inclusion mechanisms: `<xsl:include>` and `<xsl:import>`. Of the two, `<xsl:include>` is the simpler. It treats the included content as if it were actually typed into the file at that point, and doesn't give it any more or less precedence relative to the surrounding text. It is best used when assembling dissimilar templates that don't overlap what they match. The DocBook driver files use this instruction to assemble a set of modules into a stylesheet.

In contrast, `<xsl:import>` lets you manage the precedence of templates and variables. It is the preferred mode of customizing another stylesheet because it lets you override definitions in the distributed stylesheet with your own, without altering the distribution files at all. You simply import the whole stylesheet and add whatever changes you want.

The precedence rules for import are detailed and rigorously defined in the XSL standard. The basic rule is that any templates and variables in the importing stylesheet have precedence over equivalent templates and variables in the imported stylesheet. Think of the imported stylesheet elements as a fallback collection, to be used only if a match is not found in the current stylesheet. You can customize the templates you want to change in your stylesheet file, and let the imported stylesheet handle the rest.

NOTE

Customizing a DocBook XSL stylesheet is the opposite of customizing a DocBook DTD. When you customize a DocBook DTD, the rules of XML and SGML dictate that the *first* of any duplicate declarations wins. Any subsequent declarations of the same element or entity are ignored. The architecture of the DTD provides slots for inserting your own custom declarations early enough in the DTD for them to override the standard declarations. In contrast, customizing an XSL stylesheet is simpler because your definitions have precedence over imported ones.

You can carry modularization to deeper levels because module files can also include or import other modules. You'll need to be careful to maintain the precedence that you want as the modules get rolled up into a complete stylesheet.

4.4.4.2 Customizing with `<xsl:import>`

There is currently one example of customizing with `<xsl:import>` in the HTML version of the DocBook stylesheets. The `xtchunk.xsl` stylesheet modifies the HTML processing to output many smaller HTML files rather than a single large file per input document. It uses XSL extensions defined only in the XSL processor **XT**. In the driver file `xtchunk.xsl`, the first instruction is `<xsl:import href="docbook.xsl"/>`. That instruction imports the original driver file, which in turn uses many `<xsl:include>` instructions to include all the modules. That single import instruction gives the new stylesheet the complete set of DocBook templates to start with.

After the import, `xtchunk.xsl` redefines some of the templates and adds some new ones. Here is one example of a redefined template:

```
Original template in autotoc.xsl
<xsl:template name="href.target">
  <xsl:param name="object" select="."/>
  <xsl:text>#</xsl:text>
  <xsl:call-template name="object.id">
    <xsl:with-param name="object" select="$object"/>
  </xsl:call-template>
</xsl:template>

New template in xtchunk.xsl
<xsl:template name="href.target">
  <xsl:param name="object" select="."/>
  <xsl:variable name="ischunk">
    <xsl:call-template name="chunk">
      <xsl:with-param name="node" select="$object"/>
```

```
    </xsl:call-template>
  </xsl:variable>

  <xsl:apply-templates mode="chunk-filename" select="$object"/>

  <xsl:if test="$ischunk='0'">
    <xsl:text>#</xsl:text>
    <xsl:call-template name="object.id">
      <xsl:with-param name="object" select="$object"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>
```

The new template handles the more complex processing of HREFs when the output is split into many HTML files. Where the old template could simply output *#object.id*, the new one outputs *filename#object.id*.

4.4.4.3 Setting stylesheet variables

You may not have to define any new templates, however. The DocBook stylesheets are parameterized using XSL variables rather than hard-coded values for many of the formatting features. Since the `<xsl:import>` mechanism also lets you redefine global variables, this gives you an easy way to customize many features of the DocBook stylesheets. Over time, more features will be parameterized to permit customization. If you find hardcoded values in the stylesheets that would be useful to customize, please let the maintainer know.

Near the end of the list of includes in the main DocBook driver file is the instruction `<xsl:include href="param.xsl"/>`. The `param.xsl` file is the most important module for customizing a DocBook XSL stylesheet. This module contains no templates, only definitions of stylesheet variables. Since these variables are defined outside of any template, they are global variables and apply to the entire stylesheet. By redefining these variables in an importing stylesheet, you can change the behavior of the stylesheet.

To create a customized DocBook stylesheet, you simply create a new stylesheet file such as `mystyle.xsl` that imports the standard stylesheet and adds your own new variable definitions. Here is an example of a complete custom stylesheet that changes the depth of sections listed in the table of contents from two to three:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version='1.0'
                xmlns="http://www.w3.org/TR/xhtml1/transitional"
                exclude-result-prefixes="#default">

<xsl:import href="docbook.xsl"/>

<xsl:variable name="toc.section.depth">3</xsl:variable>
<!-- Add other variable definitions here -->

</xsl:stylesheet>
```

Following the opening stylesheet element are the import instruction and one variable definition. The variable `toc.section.depth` was defined in `param.xsl` with value "2", and here it is defined as "3". Since the importing stylesheet takes precedence, this new value is used. Thus documents processed with `mystyle.xsl` instead of `docbook.xsl` will have three levels of sections in the tables of contents, and all other processing will be the same.

Use the list of variables in `param.xsl` as your guide for creating a custom stylesheet. If the changes you want are controlled by a variable there, then customizing is easy.

4.4.4.4 Writing your own templates

If the changes you want are more extensive than what is supported by variables, you can write new templates. You can put your new templates directly in your importing stylesheet, or you can modularize your importing

stylesheet as well. You can write your own stylesheet module containing a collection of templates for processing lists, for example, and put them in a file named `mylists.xsl`. Then your importing stylesheet can pull in your list templates with a `<xsl:include href="mylists.xsl"/>` instruction. Since your included template definitions appear after the main import instruction, your templates will take precedence.

You'll need to make sure your new templates are compatible with the remaining modules, which means:

- Any named templates should use the same name so calling templates in other modules can find them.
- Your template set should process the same elements matched by templates in the original module, to ensure complete coverage.
- Include the same set of `<xsl:param>` elements in each template to interface properly with any calling templates, although you can set different values for your parameters.
- Any templates that are used like subroutines to return a value should return the same data type.

4.4.4.5 Writing your own driver

Another approach to customizing the stylesheets is to write your own driver file. Instead of using `<xsl:import href="docbook.xsl"/>`, you copy that file to a new name and rewrite any of the `<xsl:include/>` instructions to assemble a custom collection of stylesheet modules. One reason to do this is to speed up processing by reducing the size of the stylesheet. If you are using a customized DocBook DTD that omits many elements you never use, you might be able to omit those modules of the stylesheet.

4.4.4.6 Localization

The DocBook stylesheets include features for localizing generated text, that is, printing any generated text in a language other than the default English. In general, the stylesheets will switch to the language identified by a lang attribute when processing elements in your documents. If your documents use the lang attribute, then you don't need to customize the stylesheets at all for localization.

As far as the stylesheets go, a lang attribute is inherited by the descendents of a document element. The stylesheet searches for a lang attribute using this XPath expression:

```
<xsl:variable name="lang-attr"
         select="($target/ancestor-or-self::*/@lang
                  |$target/ancestor-or-self::*/@xml:lang)[last()]"/>
```

This locates the attribute on the current element or its most recent ancestor. Thus a lang attribute is in effect for an element and all of its descendents, unless it is reset in one of those descendents. If you define it in only your document root element, then it applies to the whole document:

```
<?xml version="1.0"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.0//EN" "docbook.dtd">
<book lang="fr">
...
</book>
```

When text is being generated, the stylesheet checks the most recent lang attribute and looks up the generated text strings for that language in a localization XML file. These are located in the `common` directory of the stylesheets, one file per language. Here is the top of the file `fr.xml`:

```
<localization language="fr">

<gentext key="abstract"                 text="R&#x00E9;sum&#x00E9;"/>
<gentext key="answer"                   text="R:"/>
<gentext key="appendix"                 text="Annexe"/>
<gentext key="article"                  text="Article"/>
<gentext key="bibliography"             text="Bibliographie"/>
...
```

The stylesheet templates use the gentext key names, and then the stylesheet looks up the associated text value when the document is processed with that lang setting. The file `l10n.xml` (note the `.xml` suffix) lists the filenames of all the supported languages.

You can also create a custom stylesheet that sets the language. That might be useful if your documents don't make appropriate use of the lang attribute. The module `l10n.xsl` defines two global variables that can be overridden with an importing stylesheet as described above. Here are their default definitions:

```
<xsl:variable name="l10n.gentext.language"></xsl:variable>
<xsl:variable name="l10n.gentext.default.language">en</xsl:variable>
```

The first one sets the language for all elements, regardless of an element's lang attribute value. The second just sets a default language for any elements that haven't got a lang setting of their own (or their ancestors).

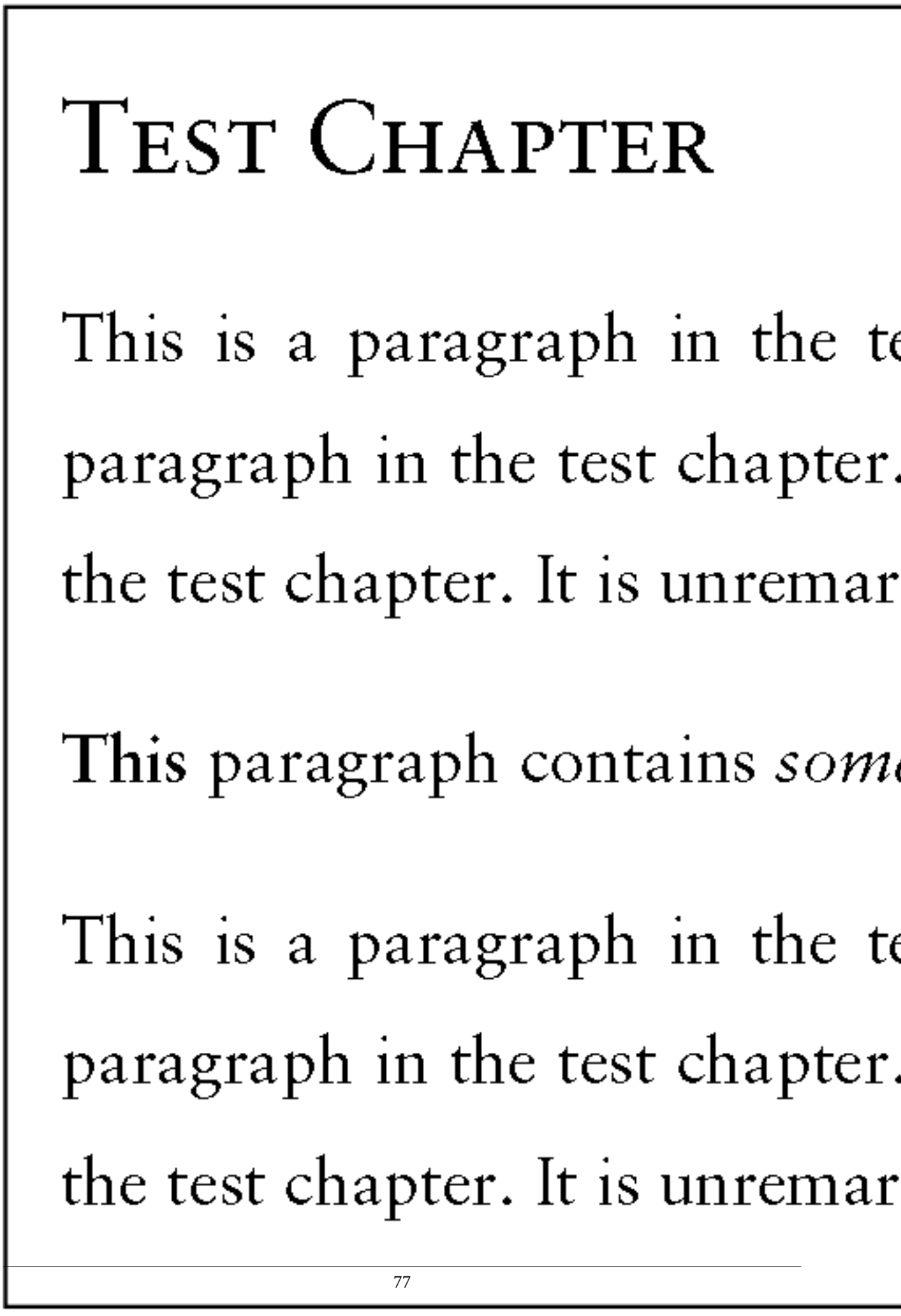


Figure 4.1: The formatted simple document

Chapter 5

Customizing DocBook

For the applications you have in mind, DocBook "out of the box" may not be exactly what you need. Perhaps you need additional inline elements or perhaps you want to remove elements that you never want your authors to use. By design, DocBook makes this sort of customization easy.

This chapter explains how to make your own *customization layer*. You might do this in order to:

- Add new elements
- Remove elements
- Change the structure of existing elements
- Add new attributes
- Remove attributes
- Broaden the range of values allowed in an attribute
- Narrow the range of values in an attribute to a specific list or a fixed value

You can use customization layers to extend DocBook or subset it. Creating a DTD that is a strict subset of DocBook means that all of your instances are still completely valid DocBook instances, which may be important to your tools and stylesheets, and to other people with whom you share documents. An *extension* adds new structures, or changes the DTD in a way that is not compatible with DocBook. Extensions can be very useful, but might have a great impact on your environment.

Customization layers can be as small as restricting an attribute value or as large as adding an entirely different hierarchy on top of the inline elements.

5.1 Should You Do This?

Changing a DTD can have a wide-ranging impact on the tools and stylesheets that you use. It can have an impact on your authors and on your legacy documents. This is especially true if you make an extension. If you rely on your support staff to install and maintain your authoring and publishing tools, check with them before you invest a lot of time modifying the DTD. There may be additional issues that are outside your immediate control. Proceed with caution.

That said, DocBook is designed to be easy to modify. This chapter assumes that you are comfortable with SGML/XML DTD syntax, but the examples presented should be a good springboard to learning the syntax if it's not already familiar to you.

5.2 If You Change DocBook, It's Not DocBook Anymore!

The DocBook DTD is usually referenced by its public identifier:

```
-//OASIS//DTD DocBook V3.1//EN
```

Previous versions of DocBook, V3.0 and the V2 variants, used the owner identifier Davenport, rather than OASIS.

If you make any changes to the structure of the DTD, it is imperative that you alter the public identifier that you use for the DTD and the modules you changed. The license agreement under which DocBook is distributed gives you complete freedom to change, modify, reuse, and generally hack the DTD in any way you want, except that you must not call your alterations "DocBook."

You should change both the owner identifier and the description. The original DocBook formal public identifiers use the following syntax:

```
-//OASIS//text-class DocBook description Vversion//EN
```

Your own formal public identifiers should use the following syntax in order to record their DocBook derivation:

```
-//your-owner-ID//text-class DocBook Vversion-Based Subset|Extension|Variant your-descrip-
```

For example:

```
-//O'Reilly//DTD DocBook V3.0-Based Subset V1.1//EN
```

If your DTD is a proper subset, you can advertise this status by using the `Subset` keyword in the description. If your DTD contains any markup model extensions, you can advertise this status by using the `Extension` keyword. If you'd rather not characterize your variant specifically as a subset or an extension, you can leave out this field entirely, or, if you prefer, use the `Variant` keyword.

There is only one file that you may change without changing the public identifier: `dbgenent.mod`. And you can add only entity and notation declarations to that file. (You can add anything you want, naturally, but if you add anything other than entity and notation declarations, you must change the public identifier!)

5.3 Customization Layers

SGML and XML DTDs are really just collections of declarations. These declarations are stored in one or more files. A complete DTD is formed by combining these files together logically. Parameter entities are used for this purpose. Consider the following fragment:

```
<!ENTITY % dbpool SYSTEM "dbpool.mod">
<!ENTITY % dbhier SYSTEM "dbhier.mod">
%dbpool;
%dbhier;
```

cl-dbpooldcl This line declares the parameter entity `dbpool` and associates it with the file `dbpool.mod`. cl-dbhierdcl This line declares the parameter entity `dbhier` and associates it with the file `dbhier.mod`. cl-dbpooluse This line references `dbpool`, which loads the file `dbpool.mod` and inserts its content here. cl-dbhieruse Similarly, this line loads `dbhier.mod`.

It is an important feature of DTD parsing that entity declarations can be repeated. If an entity is declared more than once, then the *first* declaration is used. Given this fragment:

```
<!ENTITY foo "Lenny">
<!ENTITY foo "Norm">
```

The replacement text for `&foo;` is "Lenny."

These two notions, that you can break a DTD into modules referenced with parameter entities and that the first entity declaration is the one that counts, are used to build "customization layers." With customization layers you can write a DTD that references some or all of DocBook, but adds your own modifications. Modifying the DTD this way means that you never have to edit the DocBook modules directly, which is a tremendous boon to maintaining your modules. When the next release of DocBook comes out, you usually only have to make changes to your customization layer and your modification will be back in sync with the new version.

Customization layers work particularly well in DocBook because the base DTD makes extensive use of parameter entities that can be redefined.

5.4 Understanding DocBook Structure

DocBook is a large and, at first glance, fairly complex DTD. Much of the apparent complexity is caused by the prolific use of parameter entities. This was an intentional choice on the part of the maintainers, who traded "raw

readability" for customizability. This section provides a general overview of the structure of the DTD. After you understand it, DocBook will probably seem much less complicated.

5.4.1 DocBook Modules

DocBook is composed of seven primary modules. These modules decompose the DTD into large, related chunks. Most modifications are restricted to a single chunk.

Figure 5.1 shows the module structure of DocBook as a flowchart.

The modules are:

`docbook.dtd` The main driver file. This module declares and references the other top-level modules.

`dbhier.mod` The hierarchy. This module declares the elements that provide the hierarchical structure of DocBook (sets, books, chapters, articles, and so on).

Changes to this module alter the top-level structure of the DTD. If you want to write a DocBook-derived DTD with a different structure (something other than a book), but with the same paragraph and inline-level elements, you make most of your changes in this module.

`dbpool.mod` The information pool. This module declares the elements that describe content (inline elements, bibliographic data, block quotes, sidebars, and so on) but are not part of the large-scale hierarchy of a document. You can incorporate these elements into an entirely different element hierarchy.

The most common reason for changing this module is to add or remove inline elements.

`dbnotn.mod` The notation declarations. This module declares the notations used by DocBook.

This module can be changed to add or remove notations.

`dbcent.mod` The character entities. This module declares and references the ISO entity sets used by DocBook.

Changes to this module can add or remove entity sets.

`dbgenent.mod` The general entities. This is a place where you can customize the general entities available in DocBook instances.

This is the place to add, for example, boiler plate text, logos for institutional identity, or additional notations understood by your local processing system.

`cals-tbl.dtd` The CALS Table Model. CALS is an initiative by the United States Department of Defense to standardize the document types used across branches of the military. The CALS table model, published in MIL-HDBK-28001, was for a long time the most widely supported SGML table model (one might now argue that the HTML table model is more widely supported by some definitions of "widely supported"). In any event, it is the table model used by DocBook.

DocBook predates the publication of the OASIS Technical Resolution TR 9503:1995 `http://www.oasis-open.org/html/a503.htm`, which defines an industry standard exchange table model and thus incorporates the *full* CALS Table Model.

Most changes to the CALS table model can be accomplished by modifying parameter entities in `dbpool.mod`; changing this DTD fragment is strongly discouraged. If you want to use a different table model, remove this one and add your own.

`*.gml` The ISO standard character entity sets. These entity sets are not actually part of the official DocBook distribution, but are referenced by default.

There are some additional modules, initially undefined, that can be inserted at several places for "redeclaration." This is described in more detail in Section 5.8.5.

5.4.2 DocBook Parameterization

Customization layers are possible because DocBook has been extensively parameterized so that it is possible to make any changes that might be desired without ever editing the actual distributed modules. The parameter entities come in several flavors:

`%*.class;` Classes group elements of a similar type: for example all the lists are in the `%list.class;`.

If you want to add a new kind of something (a new kind of list or a new kind of verbatim environment, for example), you generally want to add the name of the new element to the appropriate class.

`%*.mix;` Mixtures are collections of classes that appear in content models. For example, the content model of the `Example` element includes `%example.mix;`. Not every element's content model is a single mixture, but elements in the same class tend to have the same mixture in their content model.

If you want to change the content model of some class of elements (lists or admonitions, perhaps), you generally want to change the definition of the appropriate mixture.

`%*.module;` The `%*.module;` parameter entities control marked sections around individual elements and their attribute lists. For example, the element and attribute declarations for `Abbrev` occur within a marked section delimited by `%abbrev.module;`.

If you want to remove or redefine an element or its attribute list, you generally want to change its module marked section to `IGNORE` and possibly add a new definition for it in your customization layer.

`%*.element;` The `%*.element;` parameter entities were introduced in DocBook V3.1; they control marked sections around individual element declarations.

`%*.attlist;` The `%*.attlist;` parameter entities were introduced in DocBook V3.1; they control marked sections around individual attribute list declarations.

`%*.inclusion;`,`%*.exclusion;` These parameter entities control the inclusion and exclusion markup in element declarations.

Changing these declarations allows you to make global changes to the inclusions and exclusions in the DTD.

`%local.*;` The `%local.*;` parameter entities are a local extension mechanism. You can add markup to most entity declarations simply by declaring the appropriate local parameter entity.

5.5 The General Structure of Customization Layers

Although customization layers vary in complexity, most of them have the same general structure as other customization layers of similar complexity.

In the most common case, you probably want to include the entire DTD, but you want to make some small changes. These customization layers tend to look like this:

```
Overrides of Entity Declarations Here

<!ENTITY % orig-docbook "-//OASIS//DTD DocBook V3.1//EN">
%orig-docbook;

New/Modified Element and Attribute Declarations Here
```

callout ??? Declare new values for parameter entities (`%local.*;`, `%*.element;`, `%*.attlist;`) that you wish to modify. callout ??? Include the entire DocBook DTD by parameter entity reference. callout ??? Add new element and attribute declarations for any elements that you added to the DTD.

In slightly more complex customization layers, the changes that you want to make are influenced by the interactions between modules. In these cases, rather than including the whole DTD at once, you include each of the modules separately, perhaps with entity or element declarations between them:

```
Overrides of Most Entity Declarations Here

<!ENTITY % orig-pool "-//OASIS//ELEMENTS DocBook Information Pool V3.1//EN">
%orig-pool;

Overrides of Document Hierarchy Entities Here

<!ENTITY % orig-hier "-//OASIS//ELEMENTS DocBook Document Hierarchy V3.1//EN">
%orig-hier;

New/Modified Element and Attribute Declarations Here

<!ENTITY % orig-notn "-//OASIS//ENTITIES DocBook Notations V3.1//EN">
%orig-notn;

<!ENTITY % orig-cent "-//OASIS//ENTITIES DocBook Character Entities V3.1//EN">
%orig-cent;

<!ENTITY % orig-gen  "-//OASIS//ENTITIES DocBook Additional General Entities V3.1//EN
%orig-gen;
```

Finally, it's worth noting that in the rare case in which you need certain kinds of very simple, "one-off" customizations, you can do them in the document subset:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
Overrides of Entity Declarations Here
New/Modified Element and Attribute Declarations Here
]>
<book>...</book>
```

5.6 Writing, Testing, and Using a Customization Layer

The procedure for creating, testing, and using a customization layer is always about the same. In this section, we'll go through the process in some detail. The rest of the sections in this chapter describe a range of useful customization layers.

5.6.1 Deciding What to Change

If you're considering writing a customization layer, there must be something that you want to change. Perhaps you want to add an element or attribute, remove one, or change some other aspect of the DTD.

Adding an element, particularly an inline element, is one possibility. If you're writing documentation about an object-oriented system, you may have noticed that DocBook provides `ClassName` but not `MethodName`. Suppose you want to add `MethodName`?

5.6.2 Deciding How to Change a Customization Layer

Figuring out what to change may be the hardest part of the process. The organization of the parameter entities is quite logical, and, bearing in mind the organization described in Section 5.4, finding something similar usually provides a good model for new changes.

`MethodName` is similar to `ClassName`, so `ClassName` is probably a good model. `ClassName` is an inline element, not a hierarchy element, so it's in `dbpool.mod`. Searching for "classname" in `dbpool.mod` reveals:

```
<!ENTITY % local.tech.char.class "">
<!ENTITY % tech.char.class
        "Action|Application|ClassName|Command|ComputerOutput
        |Database|Email|EnVar|ErrorCode|ErrorName|ErrorType|Filename
        |Function|GUIButton|GUIIcon|GUILabel|GUIMenu|GUIMenuItem
        |GUISubmenu|Hardware|Interface|InterfaceDefinition|KeyCap
        |KeyCode|KeyCombo|KeySym|Literal|Constant|Markup|MediaLabel
        |MenuChoice|MouseButton|MsgText|Option|Optional|Parameter
        |Prompt|Property|Replaceable|ReturnValue|SGMLTag|StructField
        |StructName|Symbol|SystemItem|Token|Type|UserInput|VarName
        %local.tech.char.class;">
```

Searching further reveals the element and attribute declarations for `ClassName`.

It would seem (and, in fact, it is the case) that adding `MethodName` can be accomplished by adding it to the local extension mechanism for `%tech.char.class;`, namely `%local.tech.char.class;`, and adding element and attribute declarations for it. A customization layer that does this can be seen in Example 5.6.1.

Example 5.6.1: Adding MethodName with a Customization Layer

```
<!ENTITY % local.tech.char.class "|MethodName">

<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;

<!ELEMENT MethodName - - ((%smallcptr.char.mix;)+)  >
<!ATTLIST MethodName
        %common.attrib;
        %classname.role.attrib;
        %local.classname.attrib;
>
```

add.methname.1 Declare the appropriate parameter entity (these are described in Section 5.4.2). The declaration in your customization layer is encountered first, so it overrides the definition in the DocBook DTD (all the local classes are defined as empty in the DTD). add.methname.2 Use a parameter entity to load the entire DocBook DTD. add.methname.3 Add an element declaration for the new element. The content model for this element is taken directly from the content model of `ClassName`. add.methname.4 Add an attribute list declaration for the new element. These are the same attributes as `ClassName`.

5.6.3 Using Your Customization Layer

In order to use the new customization layer, you must save it in a file, for example `mydocbk.dtd`, and then you must use the new DTD in your document.

The simplest way to use the new DTD is to point to it with a system identifier:

```
<!DOCTYPE chapter SYSTEM "/path/to/mydocbk.dtd">
<chapter><title>My Chapter</title>
<para>
The Java <classname>Math</classname> class provides a
<methodname>abs</methodname> method to compute absolute value of a number.
</para>
</chapter>
```

If you plan to use your customization layer in many documents, or exchange it with interchange partners, consider giving your DTD its own public identifier, as described in Section 5.2

In order to use the new public identifier, you must add it to your catalog:

```
PUBLIC "-//Your Organization//DTD DocBook V3.1-Based Extension V1.0//EN"
       "/share/sgml/mydocbk.dtd"
```

and use that public identifier in your documents:

```
<!DOCTYPE chapter
  PUBLIC "-//Your Organization//DTD DocBook V3.1-Based Extension V1.0//EN">
<chapter><title>My Chapter</title>
<para>
The Java <classname>Math</classname> class provides a
<methodname>abs</methodname> method to compute absolute value of a number.
</para>
</chapter>
```

If you're using XML, remember that you must provide a system identifier that satisfies the requirements of a Uniform Resource Identifier (URI).

5.7 Testing Your Work

DTDs, by their nature, contain many complex, interrelated elements. Whenever you make a change to the DTD, it's always wise to use a validating parser to double-check your work. A parser like **nsgmls** from James Clark's SP can identify elements (attributes, parameter entities) that are declared but unused, as well as ones that are used but undeclared.

A comprehensive test can be accomplished with **nsgmls** using the `-wall` option. Create a simple test document and run:

```
nsgmls -sv -wall test.sgm
```

nsgmls.opt.1

The `-s` option tells **nsgmls** to suppress its normal output (it will still show errors, if there are any). The `-v` option tells **nsgmls** to print its version number; this ensures that you always get *some* output, even if there are no errors. nsgmls.opt.2

The `-wall` option tells **nsgmls** to provide a comprehensive list of all errors and warnings. You can use less verbose, and more specific options instead; for example, `-wundefined` to flag undefined elements or `-wunused-param` to warn you about unused parameter entities. The **nsgmls** documentation provides a complete list of warning types.

5.7.1 DocBook V3.1 Warnings

If you run the preceding command over DocBook V3.1, you'll discover one warning generated by the DTD:

```
nsgmls:I: SP version "1.3"
nsgmls:cals-tbl.dtd:314:37:W: content model is mixed but does not allow #PCDATA every
```

This is not truly an error in the DTD, and can safely be ignored. The warning is caused by "pernicious mixed content" in the content model of DocBook's `Entry` element. See the `Entry` reference page for a complete discussion.

5.8 Removing Elements

DocBook has a large number of elements. In some authoring environments, it may be useful or necessary to remove some of these elements.

5.8.1 Removing MsgSet

`MsgSet` is a favorite target. It has a complex internal structure designed for describing interrelated error messages, especially on systems that may exhibit messages from several different components. Many technical documents can do without it, and removing it leaves one less complexity to explain to your authors.

Example 5.8.1 shows a customization layer that removes the `MsgSet` element from DocBook:

Example 5.8.1: Removing MsgSet

```
<!ENTITY % compound.class "Procedure|SideBar">
<!ENTITY % msgset.content.module "IGNORE">
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

rmv.msgset.1 Remove `MsgSet` from the `%compound.class;`. This is the only place in the DTD where `MsgSet` is referenced. rmv.msgset.2 Exclude the definition of `MsgSet` and all of its subelements from the DTD.

5.8.2 Removing Computer Inlines

DocBook contains a large number of computer inlines. The DocBook inlines define a domain-specific vocabulary. If you're working in another domain, many of them may be unnecessary. You can remove a bunch of them by redefining the `%tech.char.class;` parameter entity and then excluding the declarations for the elements removed. The initial definition of `%tech.char.class;` is:

```
<!ENTITY % tech.char.class
    "Action|Application|ClassName|Command|ComputerOutput
    |Database|Email|EnVar|ErrorCode|ErrorName|ErrorType|Filename
    |Function|GUIButton|GUIIcon|GUILabel|GUIMenu|GUIMenuItem
    |GUISubmenu|Hardware|Interface|InterfaceDefinition|KeyCap
    |KeyCode|KeyCombo|KeySym|Literal|Markup|MediaLabel|MenuChoice
    |MouseButton|MsgText|Option|Optional|Parameter|Prompt|Property
    |Replaceable|ReturnValue|SGMLTag|StructField|StructName
    |Symbol|SystemItem|Token|Type|UserInput
    %local.tech.char.class;">
```

When examining this list, it seems that you can delete all of the inlines except, perhaps, `Application`, `Command`, `Email`, `Filename`, `Literal`, `Replaceable`, `Symbol`, and `SystemItem`. The following customization layer removes them.

Initially we removed several more elements from `%tech.char.class;` (`%function.module;`, `%keycap.module;`), but using the testing procedure described in Section 5.7, we discovered that these elements are used in other content models. Because they are used in other content modules, they cannot simply be removed from the DTD by deleting them from `%tech.char.class;`. Even though they can't be deleted outright, we've taken them out of most inline contexts.

It's likely that a customization layer that removed this many technical inlines would also remove some larger technical structures (`MsgSet`, `FuncSynopsis`), which allows you to remove additional elements from the DTD.

5.8.3 Removing Synopsis Elements

Another possibility is removing the complex Synopsis elements. The customization layer in Example 5.8.3 removes `CmdSynopsis` and `FuncSynopsis`.

Completely removing all Synopsis elements would require a more extensive customization. You can't make any of the `%*.class;` parameter entities completely empty without changing all of the parameter entities that use them. See Section 5.8.6.

5.8.4 Removing Sectioning Elements

Perhaps you want to restrict your authors to only three levels of sectioning. To do that, you must remove the `Sect4` and `Sect5` elements, as shown in Example 5.8.4.

In order to completely remove an element that isn't in the information pool, it is usually necessary to redefine the elements that include it. In this case, because we're removing the `Sect4` element, we must redefine the `Sect3` element that uses it.

5.8.5 Removing Admonitions from Table Entries

All of the customization layers that we've examined so far have been fairly straightforward. This section describes a much more complex customization layer. Back in Section 5.4.1 we mentioned that several additional modules existed for "redeclaration." The customization layer developed in this section cannot be written without them.

The goal is to remove admonitions (`Warning`, `Caution`, `Note`) from table entries.

Example 5.8.5 is a straightforward, and incorrect, attempt.

Because the parameter entity `%tabentry.mix;` defines the mixture of elements allowed in table entries, you should remove admonitions.

If you attempt to parse this DTD, you'll find that the declaration of `%tabentry.mix;` contains errors. While you can redefine parameter entities, you cannot make reference to entities that have not been defined yet, so the use of `%list.class;`, `%linespecific.class;`, and so on, aren't allowed.

Your second attempt might look like Example 5.8.6.

Declaring `%tabentry.mix;` after the DTD has been loaded removes the errors.

This example contains no errors, but it also doesn't have any effect. Remember, only the first entity declaration counts, so the declaration of `%tabentry.mix;` in `dbpool.mod` is the one used, not your redeclaration.

The only way to fix this problem is to make use of one of the redeclaration placeholders in DocBook.

Redeclaration placeholders are spots in which you can insert definitions into the middle of the DTD. There are four redeclaration placeholders in DocBook:

`%rdbmods;` Inserted in `docbook.dtd`, between `dbpool.mod` and `dbhier.mod`. This placeholder is controlled by the `%intermod.redecl.module;` marked section.

`%rdbpool;` Inserted in the middle of `dbpool.mod`, between the `%*.class;` and `%*.mix;` entity declarations. This placeholder is controlled by the `%dbpool.redecl.module;` marked section.

`%rdbhier;` Inserted in the middle of `dbhier.mod`, between the `%*.class;` and `%*.mix;` entity declarations. This placeholder is controlled by the `%dbhier.redecl.module;` marked section.

`%rdbhier2;` Also inserted into `dbhier.mod`, after the `%*.mix;` entity declarations. This placeholder is controlled by the `%dbhier.redecl2.module;` marked section.

Use the redeclaration placeholder that it occurs nearest to, but before the entity that you want to redeclare. In our case, this is `%rdbpool;`, as seen in Example 5.8.7.

Example 5.8.7 uses numeric character entity references to escape the % signs in the entity declarations and nests an entity declaration in another parameter entity. All of this is perfectly legal, but a bit confusing. A clearer solution, and the only practical solution if you're doing anything more than a single redeclaration, is to place the new declarations in another file and include them in your customization layer by reference, like this:

5.8.6 Removing an Entire Class

Perhaps the modification that you want to make is to completely remove an entire class of elements. (If you have no need for synopsis elements of any sort, why not remove them?) In order to remove an entire class of elements, you must not only redefine the class as empty, but you must also redefine all of the parameter entities that use that class. The customization layer below completely removes the `%synop.class;` from DocBook. It requires a customization layer, shown in Example 5.8.9, that includes both a redeclaration module in `dbpool.mod` and a redeclaration module in `dbhier.mod`.

5.9 Removing Attributes

Just as there may be more elements than you need, there may be more attributes.

5.9.1 Removing an Attribute

Suppose you want to remove the RenderAs attribute from the `Sect1` element. RenderAs allows the author to "cheat" in the presentation of hierarchy by specifying that the stylesheet should render a `Sect1` as something else: a `Sect3`, perhaps. Example 5.9.1 details the removal of RenderAs.

remvattr.1 Turn off the `Sect1` module so that the element and attribute declarations in the DTD will be ignored. remvattr.2 Include the DocBook DTD. remvattr.3 By keeping the local attribute declaration, we leave open the possibility of a simple customization layer on top of our customization layer. remvattr.4 Similarly, we keep the parameterized definition of the Role attribute. remvattr.5 We're changing the attribute list, not the element, so we've simply copied the `Sect1` element declaration from the DocBook DTD. remvattr.6 Finally, we declare the attribute list, leaving out the RenderAs.

5.9.2 Subsetting the Common Attributes

DocBook defines eleven common attributes; these attributes appear on *every* element. Depending on how you're processing your documents, removing some of them can both simplify the authoring task and improve processing speed.

Some obvious candidates are:

Effectivity attributes (Arch, OS,...) If you're not using all of the effectivity attributes in your documents, you can get rid of up to seven attributes in one fell swoop.

Lang If you're not producing multilingual documents, you can remove Lang.

Remap The Remap attribute is designed to hold the name of a semantically equivalent construct from a previous markup scheme (for example, a Microsoft Word style template name, if you're converting from Word). If you're authoring from scratch, or not preserving previous constructs with Remap, you can get rid of it.

XrefLabel If your processing system isn't using XrefLabel, it's a candidate as well.

The customization layer in Example 5.9.2 reduces the common attributes to just ID and Lang.

By definition, whatever attributes you define in the `%common.attrib;` and `%idreq.common.attrib;` parameter entities are the common attributes. In `dbpool.mod`, these parameter entities are defined in terms of other parameter entities, but there's no way to preserve that structure in your customization layer.

5.10 Adding Elements: Adding a Sect6

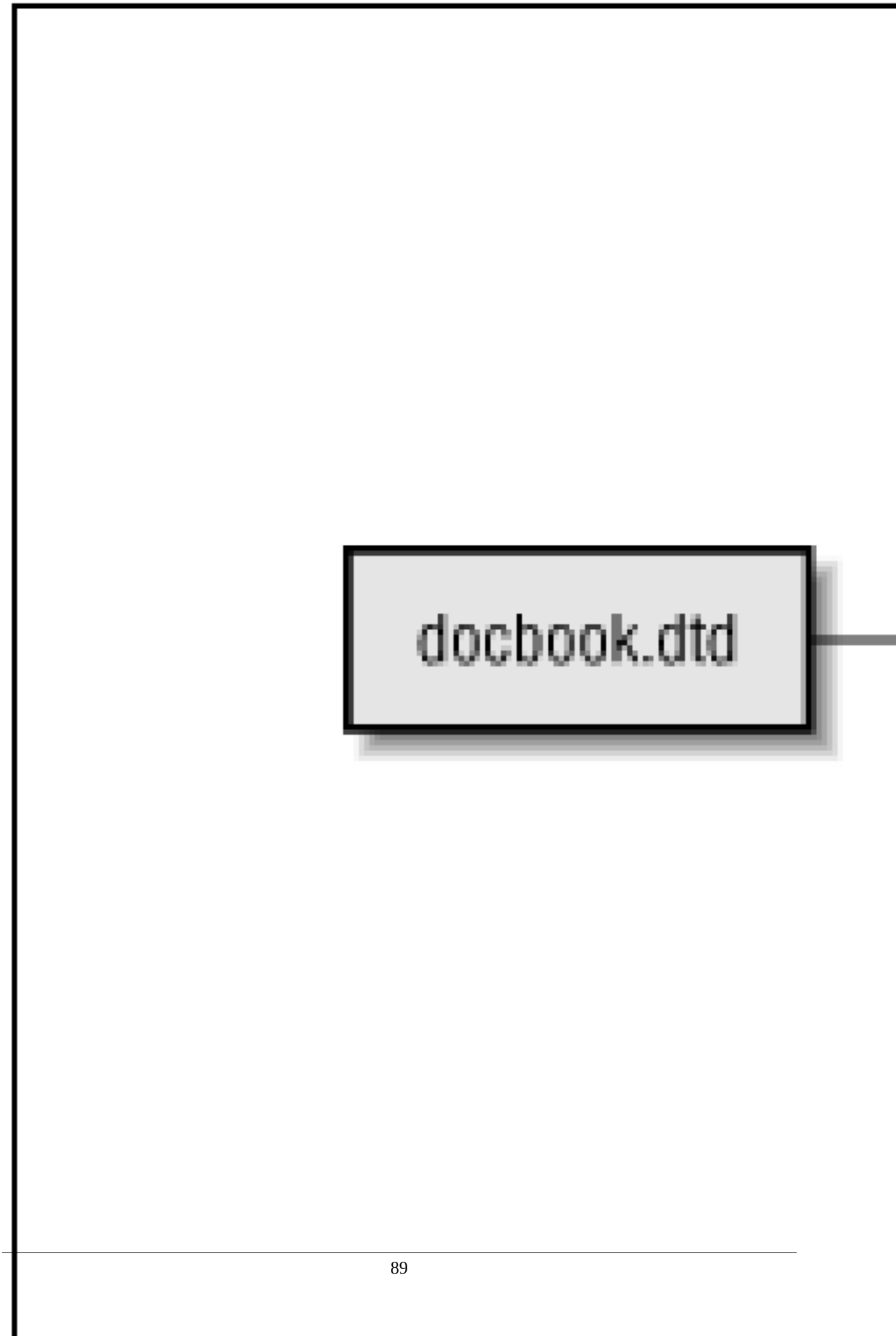
Adding a structural (as opposed to information pool) element generally requires adding its name to a class and then providing the appropriate definitions. Example 5.10.1 extends DocBook by adding a `Sect6` element.

Here we've redefined `Sect5` to include `Sect6` and provided a declaration for `Sect6`. Note that we didn't bother to provide RenderAs attributes in our redefinitions. To properly support `Sect6`, you might want to redefine all of the sectioning elements so that `Sect6` is a legal attribute value for RenderAs.

5.11 Other Modifications: Classifying a Role

The Role attribute, found on almost all of the elements in DocBook, is a `CDATA` attribute that can be used to subclass an element. In some applications, it may be useful to modify the definition of Role so that authors must choose one of a specific set of possible values.

In Example 5.11.1, Role on the `Procedure` element is constrained to the values `Required` or `Optional`.


docbook.dtd

Example 5.8.2: Removing Computer Inlines

```
<!ENTITY % tech.char.class
        "Application|Command|Email|Filename|Literal
        |Replaceable|Symbol|SystemItem">
<!ENTITY % action.module "IGNORE">
<!ENTITY % classname.module "IGNORE">
<!ENTITY % computeroutput.module "IGNORE">
<!ENTITY % database.module "IGNORE">
<!ENTITY % envar.module "IGNORE">
<!ENTITY % errorcode.module "IGNORE">
<!ENTITY % errorname.module "IGNORE">
<!ENTITY % errortype.module "IGNORE">
<!--<!ENTITY % function.module "IGNORE">-->
<!ENTITY % guibutton.module "IGNORE">
<!ENTITY % guiicon.module "IGNORE">
<!ENTITY % guilabel.module "IGNORE">
<!ENTITY % guimenu.module "IGNORE">
<!ENTITY % guimenuitem.module "IGNORE">
<!ENTITY % guisubmenu.module "IGNORE">
<!ENTITY % hardware.module "IGNORE">
<!ENTITY % interface.module "IGNORE">
<!ENTITY % interfacedefinition.module "IGNORE">
<!--<!ENTITY % keycap.module "IGNORE">-->
<!ENTITY % keycode.module "IGNORE">
<!--<!ENTITY % keycombo.module "IGNORE">-->
<!--<!ENTITY % keysym.module "IGNORE">-->
<!ENTITY % markup.module "IGNORE">
<!ENTITY % medialabel.module "IGNORE">
<!ENTITY % menuchoice.module "IGNORE">
<!--<!ENTITY % mousebutton.module "IGNORE">-->
<!--<!ENTITY % msgtext.module "IGNORE">-->
<!--<!ENTITY % option.module "IGNORE">-->
<!--<!ENTITY % optional.module "IGNORE">-->
<!--<!ENTITY % parameter.module "IGNORE">-->
<!ENTITY % prompt.module "IGNORE">
<!ENTITY % property.module "IGNORE">
<!ENTITY % returnvalue.module "IGNORE">
<!ENTITY % sgmltag.module "IGNORE">
<!ENTITY % structfield.module "IGNORE">
<!ENTITY % structname.module "IGNORE">
<!ENTITY % token.module "IGNORE">
<!ENTITY % type.module "IGNORE">
<!ENTITY % userinput.module "IGNORE">
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Example 5.8.3: Removing CmdSynopsis and FuncSynopsis

```
<!ENTITY % synop.class "Synopsis">
<!-- Instead of "Synopsis|CmdSynopsis|FuncSynopsis %local.synop.class;" -->

<!ENTITY % funcsynopsis.content.module "IGNORE">
<!ENTITY % cmdsynsynopsis.content.module "IGNORE">

<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Example 5.8.4: Removing Sect4 and Sect5 Elements

Example 5.8.5: Removing Admonitions (First Attempt: Incorrect)

```
<!-- THIS CUSTOMIZATION LAYER CONTAINS ERRORS -->
<!ENTITY % tabentry.mix
        "%list.class;
        |%linespecific.class;
        |%para.class;        |Graphic
        %local.tabentry.mix;">
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Example 5.8.6: Removing Admonitions (Second Attempt: Incorrect)

```
<!-- THIS CUSTOMIZATION LAYER DOESN'T WORK -->
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
<!ENTITY % tabentry.mix
        "%list.class;
        |%linespecific.class;
        |%para.class;        |Graphic
        %local.tabentry.mix;">
```

Example 5.8.7: Removing Admonitions (Third Attempt: Correct, if confusing)

```
<!ENTITY % dbpool.redecl.module "INCLUDE">
<!ENTITY % rdbpool
'<!ENTITY % local.tabentry.mix "">
<!ENTITY % tabentry.mix
        "&#37;list.class;
        |&#37;linespecific.class;
        |&#37;para.class;        |Graphic
        &#37;local.tabentry.mix;">'>

<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Example 5.8.8: Removing Admonitions (Fourth Attempt: Correct)

In your customization layer:

```
<!ENTITY % dbpool.redecl.module "INCLUDE">
<!ENTITY % rdbpool SYSTEM "rdbpool.mod">

<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

In `rdbpool.mod`:

```
<!ENTITY % local.tabentry.mix "">
<!ENTITY % tabentry.mix
        "%list.class;
        |%linespecific.class;
        |%para.class;        |Graphic
        %local.tabentry.mix;">
```

Example 5.8.9: Removing synop.class

In the customization layer:

In `remv.synop.class.rdbpool.mod`:

In `remv.synop.class.rdbhier.mod`:

Example 5.9.1: Removing RenderAs from Sect1

```
<!ENTITY % sect1.module "IGNORE">

<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;

<!ENTITY % local.sect1.attrib "">
<!ENTITY % sect1.role.attrib "%role.attrib;">
<!ELEMENT Sect1 - O (Sect1Info?, (%sect.title.content;), (%nav.class;)*,
        (((%divcomponent.mix;)+,
        ((%refentry.class;)* | Sect2* | SimpleSect*))
        | (%refentry.class;)+ | Sect2+ | SimpleSect+), (%nav.class;)*)
        +(%ubiq.mix;)>
<!ATTLIST Sect1
        %label.attrib;
        %status.attrib;
        %common.attrib;
        %sect1.role.attrib;
        %local.sect1.attrib;
>
```

Example 5.9.2: Removing Common Attributes

```
<!ENTITY % common.attrib
"ID   ID    #IMPLIED
 Lang CDATA #IMPLIED"
>
<!ENTITY % idreq.common.attrib
"ID   ID    #REQUIRED
 Lang CDATA #IMPLIED"
>
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Example 5.10.1: Adding a Sect6 Element

```
<!ENTITY % sect5.module "IGNORE">
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
<!-- Add Sect6 to content model of Sect5 -->
<!ENTITY % sect5.role.attrib "%role.attrib;">
<!ELEMENT Sect5 - O (Sect5Info?, (%sect.title.content;), (%nav.class;)*,
        (((%divcomponent.mix;)+,
                ((%refentry.class;)* | Sect6* | SimpleSect*))
        | (%refentry.class;)+ | Sect6+ | SimpleSect+), (%nav.class;)*)>
<!ATTLIST Sect5
        %label.attrib;
        %status.attrib;
        %common.attrib;
        %sect5.role.attrib;
>
<!ENTITY % sect6.role.attrib "%role.attrib;">
<!ELEMENT Sect6 - O (Sect6Info?, (%sect.title.content;), (%nav.class;)*,
        (((%divcomponent.mix;)+, ((%refentry.class;)* | SimpleSect*))
        | (%refentry.class;)+ | SimpleSect+), (%nav.class;)*)>
<!ATTLIST Sect6
        %label.attrib;
        %status.attrib;
        %common.attrib;
        %sect6.role.attrib;
>
```

Example 5.11.1: Changing Role on Procedure

```
<!ENTITY % procedure.role.attrib "Role (Required|Optional) #IMPLIED">
<!-- load DocBook -->
<!ENTITY % DocBookDTD PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
%DocBookDTD;
```

Part II

Reference

Chapter 6

DocBook Element Reference

■ abbrev

Name

abbrev – An abbreviation, especially one followed by a period

Synopsis

Mixed Content Model

abbrev ::= (#PCDATA|`acronym`|`emphasis`|`trademark`|`link`|`olink`|`ulink`|`anchor`| `remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`| `indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%gen.char.class;` `%info.class;`

`%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;`

`%tbl.entry.mdl;` `%title.char.mix;`

Description

An abbreviation, especially one followed by a period.

Processing expectations

Formatted inline.

Parents

These elements contain abbrev: `appendixinfo`, `application`, `articleinfo`, `attribution`, `biblioentry`, `bibliographyinfo`, `bibliomisc`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `bridgehead`, `chapterinfo`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glossaryinfo`, `glossentry`, `glosssee`, `glossseealso`, `glossterm`, `indexinfo`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `objectinfo`, `olink`, `para`, `partinfo`, `phrase`, `prefaceinfo`, `primary`, `primaryie`, `productname`, `programlisting`, `quote`, `refentryinfo`, `refentrytitle`, `referenceinfo`, `refpurpose`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `setindexinfo`, `setinfo`, `sidebarinfo`, `simpara`, `subtitle`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in abbrev: `acronym`, `anchor`, `beginpage`, `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

acronym, emphasis, foreignphrase, phrase, quote, wordasword.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <abbrev>Assn.</abbrev> of Computing Machinery would probably never
abbreviate &ldquo;Association&rdquo; like this.
</para>
```

The Assnof Computing Machinery would probably never abbreviate “Association” like this. For additional examples, see also bibliography, biblioset, footnote, glossary.

■ abstract

Name

abstract – A summary

Synopsis

Content Model

abstract ::= (title?, (formalpara|para|simpara)+)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %bookcomponent.content; %component.mix;

%descobj.class; %divcomponent.mix; %info.class;

%listpreamble.mix; %refcomponent.mix;

Description

An abstract can occur in most components of DocBook. It is expected to contain some sort of summary of the content with which it is associated (by containment).

Processing expectations

Formatted as a displayed block. Sometimes suppressed. Often presented in alternate outputs.

Parents

These elements contain abstract: appendix, appendixinfo, article, articleinfo, bibliodiv, biblioentry, bibliography, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, blockquote, bookinfo, callout, chapter, chapterinfo, glossary, glossaryinfo, glossdiv, index, indexinfo, itemizedlist, listitem, msgexplan, msgtext, objectinfo, orderedlist, partinfo, partintro, preface, prefaceinfo, procedure, refentryinfo, referenceinfo, refsect1, refsect1info, refsect2, refsect2info, refsect3, refsect3info, refsection, refsectioninfo, refsynopsisdiv, refsynopsisdivinfo, sect1, sect1info, sect2, sect2info, sect3, sect3info, sect4, sect4info, sect5, sect5info, section, sectioninfo, setindex, setindexinfo, setinfo, sidebarinfo, simplesect, step, variablelist.

Children

The following elements occur in abstract: formalpara, para, simpara, title.

See Also

`blockquote`, `epigraph`, `highlights`, `sidebar`.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Some Chapter</title>
<abstract>
<para>
In brief, &hellip;
</para>
</abstract>
<para>
In this chapter, &hellip;
</para>
</chapter>
```

For additional examples, see also `section`.

■ accel

Name

accel – A graphical user interface (GUI) keyboard shortcut

Synopsis

Mixed Content Model

accel ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

An accelerator is usually a letter used with a meta key (such as control or alt) to activate some element of a GUI without using the mouse to point and click at it.

Processing expectations

Formatted inline. Often underlined.

Parents

These elements contain accel: `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `interface`.

Children

The following elements occur in accel: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

See Also

`guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `keycap`, `keycode`, `keycombo`, `keysym`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para os="windows">
You can exit most Windows applications by selecting
<guimenuitem><accel>Q</accel>uit</guimenuitem> or
<guimenuitem>E<accel>x</accel>it</guimenuitem> from
the <guimenu><accel>F</accel>ile</guimenu> menu.
</para>
```

You can exit most Windows applications by selecting **Quit** or **Exit** from the **File** menu.

■ ackno

Name

ackno – Acknowledgements in an `Article`

Synopsis

Mixed Content Model

ackno ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

Acknowledgements in an `Article`.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain ackno: `article`.

Children

The following elements occur in ackno: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<article>
<articleinfo>
<author><firstname>Norman</firstname><surname>Walsh</surname></author>
</articleinfo>
<para>
This is an odd, minimalist sort of article example.
</para>
<ackno>
First, I'd like to thank the members of the academy, &hellip;
</ackno>
</article>
```

■ acronym

Name

acronym – An often pronounceable word made from the initial (or selected) letters of a name or phrase

Synopsis

Mixed Content Model

acronym ::= (#PCDATA|acronym|emphasis|trademark|link|olink|ulink|anchor| remark| subscript|superscript|inlinegraphic|inlinemediaobject| indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

%word.char.mix;

Description

A pronounceable contraction of initials. An acronym is often printed in all capitals or small capitals, although this is sometimes incorrect (consider dpi or bps).

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain acronym: abbrev, acronym, application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, firstterm, foreignphrase, glossentry, glosssee, glossseealso, glossterm, label, lineannotation, link, literallayout, lotentry, manvolnum, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink, wordasword.

Children

The following elements occur in acronym: acronym, anchor, beginpage, emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, subscript, superscript, trademark, ulink.

See Also

abbrev, emphasis, foreignphrase, phrase, quote, wordasword.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In the United States, <acronym>NASA</acronym> stands for the
National Aeronautics and Space Administration.
</para>
```

In the United States, NASA stands for the National Aeronautics and Space Administration. For additional examples, see also application, command, glossary, hardware, option, variablelist.

■ action

Name

action – A response to a user event

Synopsis

Mixed Content Model

action ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry "none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

Actions are usually associated with GUIs. An event might be movement or clicking of the mouse, a change in focus, or any number of other occurrences.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of Action will be constrained to (#PCDATA | Replaceable | InlineGraphic) in DocBook V4.0.

Parents

These elements contain action: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in action: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Action.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Selecting <guimenuitem>Close</guimenuitem> closes the file and automatically
runs <action>end of job cleanup</action>.
</para>
```

Selecting **Close** closes the file and automatically runs end of job cleanup.

■ address

Name

address – A real-world address, generally a postal address

Synopsis

Mixed Content Model

address ::= (#PCDATA|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|street|pob|postcode|city|state|country|phone|fax|email|otheraddr)*

Attributes Common attributes

Name Type Default

format linespecific "linespecific"

linenumbering numbered | unnumbered

None

Parameter Entities

%admon.mix; %bibliocomponent.mix; %bookcomponent.content;

%component.mix; %divcomponent.mix; %example.mix;

%figure.mix; %footnote.mix; %glossdef.mix;

%indexdivcomponent.mix; %info.class; %informal.class;

%listpreamble.mix; %para.mix; %qandaset.mix;

%refcomponent.mix; %revdescription.mix; %sidebar.mix;

Description

An address is generally a postal address, although it does contain elements for FAX and Email addresses as well as the catch-all OtherAddr.

The linespecific notation on the Format attribute makes line breaks and other spaces significant in an Address.

Processing expectations

Formatted as a displayed block. This element is displayed "verbatim"; whitespace and linebreaks within this element are significant.

Future Changes

The xml:space attribute is automatically provided in the XML DTD.

Parents

These elements contain address: affiliation, answer, appendix, appendixinfo, article, articleinfo, author, bibliodiv, biblioentry, bibliography, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, blockquote, bookinfo, callout, caution, chapter, chapterinfo, confgroup, constraintdef, editor, example, figure, footnote, glossary, glossaryinfo, glossdef, glossdiv, important, index, indexdiv, indexinfo, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, objectinfo, orderedlist, othercredit, para, partinfo, partintro, preface, prefaceinfo, procedure, publisher, qandadiv, qandaset, question, refentryinfo, referenceinfo, refsect1, refsect1info, refsect2, refsect2info, refsect3, refsect3info, refsection, refsectioninfo, refsynopsisdiv, refsynopsisdivinfo, revdescription, sect1, sect1info, sect2, sect2info, sect3, sect3info, sect4, sect4info, sect5, sect5info, section, sectioninfo, setindex, setindexinfo, setinfo, sidebar, sidebarinfo, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in address: affiliation, authorblurb, city, contrib, country, email, fax, firstname, honorific, lineage, otheraddr, othername, personname, phone, pob, postcode, state, street, surname.

Attributes

format The Format attribute applies the linespecific notation to all Addresses. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of an Address are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

See Also

city, country, email, fax, otheraddr, phone, pob, postcode, state, street.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>An example of a postal mail address in the United States:
<address>
John and Jane Doe
<street>100 Main Street</street>
```

```
<city>Anytown</city>, <state>NY</state> <postcode>12345</postcode>
<country>USA</country>
</address>
</para>
```

An example of a postal mail address in the United States:

```
John and Jane Doe
100 Main Street
Anytown, NY 12345
USA
```

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>An example of a post office box address in the United States:
<address>
<pob>P.O. Box 1234</pob>
<city>Anytown</city>, <state>MA</state> <postcode>12345</postcode>
<country>USA</country>
</address>
</para>
```

An example of a post office box address in the United States:

```
P.O. Box 1234
Anytown, MA 12345
USA
```

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>Addresses can also include phone numbers:
<address>
John Smith
<street>100 Pine Blvd</street>
<city>Anytown</city>, <state>NY</state> <postcode>12345</postcode>
<country>USA</country>
<phone>914.555.1212</phone>
<fax>914.555.1212</fax>
</address>
</para>
```

Addresses can also include phone numbers:

```
John Smith
100 Pine Blvd
Anytown, NY 12345
USA
914.555.1212
914.555.1212
```

For additional examples, see also `confgroup`, `otheraddr`, `publisher`.

■ affiliation

Name

affiliation – The institutional affiliation of an individual

Synopsis

Content Model

affiliation ::= (`shortaffil`?,`jobtitle`*,`orgname`?,`orgdiv`*,`address`*)

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;` `%person.ident.mix;`

Description

The institutional affiliation of an author, contributor, or other individual.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Future Changes

In DocBook V4.0, `Affiliation` will be removed from some of the places in which it now occurs. Instead of appearing inside `Author`, for example, a new wrapper element will be created to hold `Author`, `AuthorBlurb`, and `Affiliation`.

Parents

These elements contain affiliation: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `collab`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in affiliation: `address`, `jobtitle`, `orgdiv`, `orgname`, `shortaffil`.

See Also

`corpname`, `firstname`, `honorific`, `jobtitle`, `lineage`, `orgdiv`, `orgname`, `othername`, `shortaffil`, `surname`.

Examples

For examples, see `author`, `authorgroup`, `bookinfo`, `contractsponsor`.

■ alt

Name

alt – Text representation for a graphical element

Synopsis

Mixed Content Model

alt ::= (#PCDATA)

Attributes Common attributes

Parameter Entities

`%equation.content; %inlineequation.content;`

Description

A text (or other nonvisual) description of a graphical element. This is intended to be an alternative to the graphical presentation.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain alt: `equation`, `informalequation`, `inlineequation`.

See Also

`audioobject`, `caption`, `graphic`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

For examples, see `equation`, `informalequation`, `inlineequation`.

■ anchor

Name

anchor – A spot in the document

Synopsis

Content Model

anchor ::= EMPTY

Attributes

Name	Type	Default
revisionflag	added changed deleted off	*None*
revision	CDATA	*None*
arch	CDATA	*None*
vendor	CDATA	*None*
security	CDATA	*None*
pagenum	CDATA	*None*
remap	CDATA	*None*
xreflabel	CDATA	*None*

conformance NMTOKENS *None*

os CDATA *None*

userlevel CDATA *None*

id ID *Required*

role CDATA *None*

condition CDATA *None*

Parameter Entities

`%admon.mix;` `%base.char.class;` `%bookcomponent.content;`
`%component.mix;` `%cptr.char.mix;` `%divcomponent.mix;`
`%genobj.class;` `%indexdivcomponent.mix;` `%listpreamble.mix;`
`%ndxterm.char.mix;` `%para.char.mix;` `%qandaset.mix;`
`%refcomponent.mix;` `%refinline.char.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%tbl.entry.mdl;` `%title.char.mix;`
`%word.char.mix;`

Description

An anchor identifies a single spot in the content. This may serve as the target for a cross reference, for example, from a `Link`. The `Anchor` element may occur almost anywhere.

`Anchor` has the Role attribute and all of the common attributes except Lang.

Processing expectations

`Anchor` has no content and generally produces no output. It is a link target.

Parents

These elements contain anchor: `abbrev`, `acronym`, `action`, `answer`, `appendix`, `application`, `article`, `attribution`, `bibliodiv`, `bibliography`, `bibliomisc`, `blockquote`, `bridgehead`, `callout`, `caution`, `chapter`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `constraintdef`, `database`, `emphasis`, `entry`, `filename`, `firstterm`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glossary`, `glossdiv`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `important`, `index`, `indexdiv`, `interfacename`, `itemizedlist`, `keycap`, `label`, `lineannotation`, `link`, `listitem`, `literal`, `literallayout`, `lotentry`, `manvolnum`, `member`, `msgaud`, `msgexplan`, `msgtext`, `note`, `olink`, `option`, `optional`, `orderedlist`, `para`, `parameter`, `partintro`, `phrase`, `preface`, `primary`, `primaryie`, `procedure`, `productname`, `programlisting`, `property`, `qandadiv`, `qandaset`, `question`, `quote`, `refentrytitle`, `refpurpose`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `remark`, `replaceable`, `revdescription`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `setindex`, `sidebar`, `simpara`, `simplesect`, `step`, `subscript`, `subtitle`, `superscript`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `tip`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`, `variablelist`, `warning`, `wordasword`.

Attributes

pagenum PageNum indicates the page on which the anchor occurs in some printed version of the document.

The PageNum attribute does not influence the pagination or page numbering of an SGML application processing the document; it is informative, not declarative.

See Also

`link`, `olink`, `ulink`, `xref`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The anchor element<anchor id="example.anchor.1"/> is empty and contributes
nothing to the flow of the content in which it occurs.  It is only useful
as a target.
</para>
```

The anchor element is empty and contributes nothing to the flow of the content in which it occurs. It is only useful as a target.

■ answer

Name

answer – An answer to a question posed in a `QandASet`

Synopsis

Content Model

answer ::= (`label`?, (`calloutlist`|`glosslist`|`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`)*, `qandaentry`*)

Attributes Common attributes

Description

Within a `QandAEntry`, a `Question` may have an `Answer`. An `Answer` is optional (some questions have no answers) and may be repeated (some questions have more than one answer).

Processing expectations

`Answer`s are frequently introduced with a label, such as “A:”. If an `Answer` has a `Label` child element, the content of that `Label` is used as the label for the `Answer`. The DefaultLabel attribute on the nearest ancestor `QandASet` of an `Answer` can be used to indicate that a processing application should automatically generate a label for the `Answer`.

Parents

These elements contain answer: `qandaentry`.

Children

The following elements occur in answer: `address`, `anchor`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `label`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis` `note`, `orderedlist`, `para`, `procedure`, `programlisting`, `programlistingco`, `qandaentry`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `synopsis`, `table`, `tip`, `variablelist`, `warning`.

Examples

For examples, see qandaset.

■ appendix

Name

appendix – An appendix in a Book or Article

Synopsis

Content Model

appendix ::= (beginpage?,appendixinfo?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary|bibliography)*, tocchap?, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset|constraintdef|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)+, (sect1*|(refentry)*|simplesect*|(section)*))|(sect1+|(refentry)+|simplesect+|(section)+)), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

status CDATA *None*

label CDATA *None*

Parameter Entities

%appendix.class; %partcontent.mix;

Description

Appendixes usually occur at the end of a document.

Processing expectations

Formatted as a displayed block. Usually introduces a forced page break and often starts on the next recto page. Sometimes restarts page numbering. Typically, appendixes are lettered rather than numbered, and appear in the table of contents.

Parents

These elements contain appendix: article, book, part.

Children

The following elements occur in appendix: abstract, address, anchor, appendixinfo, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, sect1, section, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, tocchap, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes. The second `Appendix` might be labeled "B", for example.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the `Appendix`.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

`article`, `book`, `chapter`, `colophon`, `dedication`, `part`, `partintro`, `preface`, `set`.

Examples

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book>
<chapter><title>Required Chapter</title>
<para>
At least one chapter, reference, part, or article is required in a book.
</para>
</chapter>
<appendix><title>Demonstration Appendix</title>
<para>
This appendix demonstrates an appendix in a book.  It has the
same broad content model as a chapter.
</para>
</appendix>
</book>
```

For additional examples, see also `book`.

■ appendixinfo

Name

appendixinfo – Meta-information for an `Appendix`

Synopsis

Content Model

appendixinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblierelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The AppendixInfo element is a wrapper for a large collection of meta-information about a Appendix. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain appendixinfo: appendix.

Children

The following elements occur in appendixinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

■ application

Name

application – The name of a software program

Synopsis

Mixed Content Model

application ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none
refentry
"none"

class hardware
software
None

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refclass.char.mix;` `%refinline.char.mix;` `%refname.char.mix;`
`%tbl.entry.mdl;` `%tech.char.class;` `%title.char.mix;`

Description

The appelation “application” is usually reserved for larger software packages—WordPerfect, for example, but not **grep**. In some domains, `Application` may also apply to a piece of hardware.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain application: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refclass`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in application: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

class Class identifies an application as either a hardware application or a software application.

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Application`.

See Also

`database`, `filename`, `hardware`, `medialabel`, `productname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
<application>Word Perfect</application> is one of several word
processors to claim support for <acronym>SGML</acronym>.
</para>
```

Word Perfect is one of several word processors to claim support for SGML. For additional examples, see also `envar`, `medialabel`, `systemitem`.

■ area

Name

area – A region defined for a `Callout` in a graphic or code example

Synopsis

Content Model

area ::= EMPTY

Attributes Common attributes

Name Type Default

units (calspair | linecolumn | linecolumnpair | linerange | other) *None*

label CDATA *None*

otherunits NMTOKEN *None*

linkends IDREFS *None*

coords CDATA *Required*

Description

An `Area` is an empty element holding information about a region in a graphic, program listing, or screen.

The region is generally decorated with a number, symbol, or other distinctive mark. The mark is usually used as the label for the `Callout` in a `CalloutList`, which allows the reader to identify which callouts are associated with which regions. The marks may be generated by the processing application from the `Area`s, or it may be added by some other process. (This is an interchange issue. See Appendix F.)

For a complete description of callouts, see `Callout`.

Processing expectations

Suppressed. This element provides data for processing but it is not expected to be rendered directly.

The processing expectations of `Callouts` are likely to deserve special consideration for interchange. See Appendix F.

The Coords, which are required, identify the location of the region. The coordinates are CDATA; how they are interpreted depends on the Units specified:

`CALSPair` The coordinates are expressed using the semantics of the CALS graphic attributes. The format of the coordinates is "x1,y1 x2,y2". This identifies a rectangle with the lower-left corner at (x1,y1) and the upper-right corner at (x2,y2). The X and Y coordinates are integers in the range 0 to 10000; they express a percentage of the total distance from 0.00 to 100.00%.

`LineColumn` The coordinates are expressed using lines and columns. The format of the coordinates is "line column." In a graphic context, the meaning of this unit is unspecified.

`LineRange` The coordinates are expressed using lines. The format of the coordinates is "startingline endingline." In a graphic context, the meaning of this unit is unspecified.

`LineColumnPair` The coordinates are expressed as a continuous flow of characters. The format of the coordinates is "line1 col1 line2 col2". This identifies a flow of characters that begins at col1 of line1 and extends to col2 of line2. If line1 and line2 are different, then the region includes all of the intervening lines (including text that occurs before col1 and after col2). In other words, this unit does not specify a rectangle. In a graphic context, the meaning of this unit is unspecified.

`OtherUnits` If specified, then the OtherUnits attribute is expected to identify the units in some implementation-specific way.

The Units attribute is not required, if it is not specified, the semantics of the coordinates must be inherited from the surrounding `AreaSpec` or `AreaSet` element or implied in some implementation-specific manner.

In processing systems in which the mark is inserted automatically, the Label attribute is provided as a mechanism for specifying what the mark should be.

The author may point to any relevant information with Linkends. DocBook does not specify a semantic for these links. One possible use would be for providing a link back to the appropriate `Callout` in an online environment.

Parents

These elements contain area: `areaset`, `areaspec`.

Attributes

coords Coords provides the coordinates of the `Area`. The coordinates should be interpreted using the Units (or OtherUnits) specified.

label Label specifies an identifying number or string that may be used in presentation. The `Area` label might be drawn on top of the figure, for example, at the position indicated by Coords.

linkends Linkends points to the `Callout`(s) which refer to this `Area`. (This provides bidirectional linking, which may be useful in online presentation.)

otherunits If none of the Units are applicable, set Units to `Other` and set OtherUnits to some application-specific description of the desired units.

units Units indicate how the specified Coords are to be interpreted. The default units vary according to the type of callout specified; `CALSPair` for graphics and `LineColumn` for line-oriented elements.

Examples

For examples, see `graphicco`, `mediaobjectco`, `programlistingco`.

areaset

Name

areaset – A set of related areas in a graphic or code example

Synopsis

Content Model

areaset ::= (`area`+)

Attributes Common attributes

Name Type Default

units	calspair linecolumn linecolumnpair linerange other	*None*

label CDATA *None*

otherunits NMTOKEN *None*

coords CDATA *Required*

Description

An `AreaSet` contains one or more `Area`s. These areas are bound in a set in order to associate them with a single `Callout` description. See `Area` for a more complete description of the areas.

For a complete description of callouts, see `Callout`.

Processing expectations

Suppressed.

Future Changes

The Coords attribute will be removed in DocBook V5.0.

Parents

These elements contain areaset: `areaspec`.

Children

The following elements occur in areaset: `area`.

Attributes

coords Coords provides the coordinates of the `AreaSet`. The coordinates should be interpreted using the Units (or OtherUnits) specified. *This attribute is erroneous since the enclosed `Area`s must also provide Coords. It will go away in DocBook 4.0.*

label Label specifies an identifying number or string that may be used in presentation.

otherunits If none of the Units are applicable, set Units to `Other` and set OtherUnits to some application-specific description of the desired units.

units Units indicate how Coords are to be interpreted.

Examples

For examples, see `graphicco`, `mediaobjectco`, `programlistingco`.

■ areaspec

Name

areaspec – A collection of regions in a graphic or code example

Synopsis

Content Model

areaspec ::= ((`area`|`areaset`)+)

Attributes Common attributes

Name Type Default

otherunits NMTOKEN *None*

units
- calspair
- linecolumn
- linecolumnpair
- linerange
- other

None

Description

An `AreaSpec` holds a collection of regions and/or region sets in a graphic, program listing, or screen that are associated with `Callout` descriptions. See `Area` for a description of the attributes.

Processing expectations

Suppressed. This element provides data for processing but it is not expected to be rendered directly.

Parents

These elements contain areaspec: `graphicco`, `imageobjectco`, `programlistingco`, `screenco`.

Children

The following elements occur in areaspec: `area`, `areaset`.

Attributes

otherunits If none of the Units are applicable, set Units to `Other` and set OtherUnits to some application-specific description of the desired units.

units Units indicate how the specified Coords are to be interpreted. The default units vary according to the type of callout specified; `CALSPair` for graphics and `LineColumn` for line-oriented elements.

See Also

`calloutlist`, `co`, `coref`, `graphicco`, `imageobjectco`, `mediaobjectco`, `programlistingco`, `screenco`.

Examples

For examples, see `graphicco`, `mediaobjectco`, `programlistingco`.

■ arg

Name

arg – An argument in a `CmdSynopsis`

Synopsis

Mixed Content Model

arg ::= (#PCDATA|`arg`|`group`|`option`|`synopfragmentref`|`replaceable`|`sbr`)*

Attributes Common attributes

Name Type Default

rep norepeat | repeat "norepeat"

choice opt | plain | req "opt"

Description

See `CmdSynopsis` for more information.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

Parents

These elements contain arg: `arg`, `cmdsynopsis`, `group`, `synopfragment`.

Children

The following elements occur in arg: `arg`, `group`, `option`, `replaceable`, `sbr`, `synopfragmentref`.

Attributes

choice Choice indicates whether the `Arg` is required (`Req` or `Plain`) or optional (`Opt`). Arguments identified as `Plain` are required, but are shown without additional decoration.

rep A Rep value of `Repeat` indicates that the `Arg` is reapeatable. This is frequently rendered with an ellipsis.

See Also

`cmdsynopsis`, `group`, `refsynopsisdiv`, `sbr`, `synopfragment`, `synopfragmentref`.

Examples

For examples, see `cmdsynopsis`, `refentry`, `synopfragment`.

■ article

Name

article – An article

Synopsis

Content Model

article ::= ((`title`,`subtitle`?,`titleabbrev`?)?, `articleinfo`?,`tocchap`?,`lot`*, (((`calloutlist`| `glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`| `constraintdef`| `anchor`|`bridgehead`|`remark`|`highlights`|`abstract`| `authorblurb`|`epigraph`|`indexterm`| `beginpage`)+, (`sect1`*| (`refentry`)*| `simplesect`*| (`section`)*))| (`sect1`+| (`refentry`)+| `simplesect`+| (`section`)+)), ((`toc`|`lot`|`index`|`glossary`|`bibliography`)| (`appendix`)| `ackno`)*)

Attributes Common attributes

Name Type Default

parentbook IDREF *None*

status CDATA *None*

class
- faq
- journalarticle
- productsheet
- specification
- techreport
- whitepaper

None

Parameter Entities

`%article.class;` `%partcontent.mix;`

Description

The Article element is a general-purpose container for articles. The content model is both quite complex and rather loose in order to accommodate the wide range of possible Article structures. Although changes to the Article element have been discussed on several occasions, no better model has been presented.

An Article is composed of a header and a body. The body may include a table of contents and multiple lists of tables, figures, and so on, before the main text of the article and may include a number of common end-matter components at the end.

Processing expectations

Formatted as a displayed block. Frequently causes a forced page break in print media. May be numbered separately and presented in the table of contents.

Future Changes

The ToC element in the content model may be replaced by TocChap. This change may be delayed if the DocBook technical committee decides to review the whole ToC/LoT apparatus.

Parents

These elements contain article: book, part.

Children

The following elements occur in article: abstract, ackno, address, anchor, appendix, articleinfo, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, sect1, section, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, tocchap, variablelist, warning.

Attributes

class Class identifies the type of article.

parentbook ParentBook holds the ID of an enclosing Book, if applicable.

status Status identifies the editorial or publication status of the Article.

> Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

appendix, book, chapter, colophon, dedication, part, partintro, preface, set.

Examples

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<article>
<articleinfo>
  <author><firstname>Norman</firstname><surname>Walsh</surname></author>
  <authorinitials>ndw</authorinitials>
  <artpagenums>339-343</artpagenums>
  <volumenum>15</volumenum>
  <issuenum>3</issuenum>
  <publisher><publishername>The TeX User's Group</publishername></publisher>
  <pubdate>1994</pubdate>
  <title>A World Wide Web Interface to CTAN</title>
  <titleabbrev>CTAN-Web</titleabbrev>
  <revhistory>
     <revision>
        <revnumber>1.0</revnumber>
        <date>28 Mar 1994</date>
        <revremark>Submitted.</revremark>
     </revision>
     <revision>
        <revnumber>0.5</revnumber>
        <date>15 Feb 1994</date>
        <revremark>First draft for review.</revremark>
     </revision>
  </revhistory>
</articleinfo>
<para>
The body of the article &hellip;
</para>
</article>
```

For additional examples, see also `ackno`, `contractsponsor`, `qandaset`.

■ articleinfo

Name

articleinfo – Meta-information for an `Article`

Synopsis

Content Model

articleinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblierelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `ArticleInfo` element is a wrapper for a large collection of meta-information about a `Article`. Much of this data is bibliographic in nature.

Prior to version 4.0 of DocBook, this element was named `ArtHeader`.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain articleinfo: `article`, `biblioentry`.

Children

The following elements occur in articleinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

`appendixinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `prefaceinfo`, `refsynopsisdivinfo`, `screeninfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setinfo`.

■ artpagenums

Name

artpagenums – The page numbers of an article as published

Synopsis

Mixed Content Model

artpagenums ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

This element holds the page numbers of an article as published. Its content is not intended to influence the page numbers used by a presentation system formatting the parent `Article`.

Processing expectations

Formatted inline. Sometimes suppressed. Although it appears at the beginning of the content model for `BlockQuote` and `Epigraph`, it is often output at the end.

Parents

These elements contain artpagenums: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in artpagenums: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

Examples

For examples, see article.

■ attribution

Name

attribution – The source of a block quote or epigraph

Synopsis

Mixed Content Model

attribution ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Description

An Attribution identifies the source to whom a BlockQuote or Epigraph is ascribed.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain attribution: blockquote, epigraph.

Children

The following elements occur in attribution: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

blockquote, epigraph.

Examples

```
<!DOCTYPE blockquote PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<blockquote>
<attribution>William Shakespeare</attribution>
<literallayout>
What say you?  Will you yield, and this avoid?
Or, guilty in defense, be thus destroyed?
</literallayout>
</blockquote>
```

> What say you? Will you yield, and this avoid?
> Or, guilty in defense, be thus destroyed?
>
> —William Shakespeare

For additional examples, see also blockquote, chapter, literallayout.

■ audiodata

Name

audiodata – Pointer to external audio data

Synopsis

Content Model

audiodata ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

BMP
CGM-BINARY
CGM-CHAR
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
IGES
format JPEG
JPG
linespecific
PCX
PIC
PNG
PS
SGML
SVG
TBL
TEX
TIFF
WMF
WPG
None

entityref ENTITY *None*

fileref CDATA *None*

Description

This empty element points to external audio data.

Processing expectations

There are two ways to provide content for `AudioData`: EntityRef or FileRef. It is best to use only one of these methods. However, if multiple sources are provided, EntityRef will be used in favor of FileRef.

Parents

These elements contain audiodata: `audioobject`.

Attributes

entityref EntityRef identifies the general entity which points to the content of the audio data.

fileref FileRef specifies the name of the file which contains the content of the audio data.

format Format identifies the format of the audio data. The Format must be a defined notation.

srccredit SrcCredit contains details about the source of the audio data.

Examples

For examples, see `audioobject`.

■ audioobject

Name

audioobject – A wrapper for audio data and its associated meta-information

Synopsis

Content Model

audioobject ::= (`objectinfo`?,`audiodata`)

Attributes Common attributes

Parameter Entities

`%mediaobject.mix;`

Description

`AudioObject` is a wrapper for `AudioData`.

Processing expectations

Its content is rendered aurally or not at all. It might not be rendered, depending on its placement within a `MediaObject` or `InlineMediaObject` and the constraints on the publishing system. For a more detailed description of the semantics involved, see `MediaObject`.

Parents

These elements contain audioobject: `inlinemediaobject`, `mediaobject`.

Children

The following elements occur in audioobject: `audiodata`, `objectinfo`.

See Also

`alt`, `caption`, `graphic`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

```
<!DOCTYPE mediaobject PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<mediaobject>
<audioobject>
  <objectinfo>
    <title>Phaser sound effect</title>
  </objectinfo>
  <audiodata fileref="phaser.wav"/>
</audioobject>
<textobject>
<phrase>A <trademark>Star Trek</trademark> phaser sound effect</phrase>
</textobject>
</mediaobject>
```

A Star Trek™ phaser sound effect

■ author

Name

author – The name of an individual author

Synopsis

Content Model

author ::= ((personname| (honorific|firstname|surname|lineage|othername|affiliation| authorblurb|contrib)+), (personblurb|email|address)*)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %docinfo.char.class; %info.class;

%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix;

Description

The Author element holds information about the author of the document in which it occurs; it is meta-information about the current document or document section, not a reference to the author of an external document.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain author: appendixinfo, application, articleinfo, attribution, authorgroup, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayou lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in author: address, affiliation, authorblurb, contrib, email, firstname, honorific, lineage, othername, personblurb, personname, surname.

See Also

authorblurb, authorgroup, collab, collabname, contrib, corpauthor, editor, othercredit, personblurb, personname.

Examples

```
<!DOCTYPE author PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<author>
  <honorific>Mr</honorific>
  <firstname>Norman</firstname>
  <surname>Walsh</surname>
```

```
  <othername role='mi'>D</othername>
  <affiliation>
    <shortaffil>ATI</shortaffil>
    <jobtitle>Senior Application Analyst</jobtitle>
    <orgname>ArborText, Inc.</orgname>
    <orgdiv>Application Developement</orgdiv>
  </affiliation>
</author>
```

For additional examples, see also ackno, article, authorgroup, bibliography, biblioset, book, bookinfo, othercredit.

■ authorblurb

Name

authorblurb – A short description or note about an author

Synopsis

Content Model

authorblurb ::= (title?, (formalpara|para|simpara)+)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %bookcomponent.content; %component.mix;

%descobj.class; %divcomponent.mix; %info.class;

%listpreamble.mix; %person.ident.mix; %refcomponent.mix;

Description

A short description of an author.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.

Parents

These elements contain authorblurb: address, appendix, appendixinfo, article, articleinfo, author, bibliodiv, biblioentry, bibliography, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, blockquote, bookinfo, callout, chapter, chapterinfo, editor, glossary, glossaryinfo, glossdiv, index, indexinfo, itemizedlist, listitem, msgexplan, msgtext, objectinfo, orderedlist, othercredit, partinfo, partintro, preface, prefaceinfo, procedure, refentryinfo, referenceinfo, refsect1, refsect1info, refsect2, refsect2info, refsect3, refsect3info, refsection, refsectioninfo, refsynopsisdiv, refsynopsisdivinfo, sect1, sect1info, sect2, sect2info, sect3, sect3info, sect4, sect4info, sect5, sect5info, section, sectioninfo, setindex, setindexinfo, setinfo, sidebarinfo, simplesect, step, variablelist.

Children

The following elements occur in authorblurb: formalpara, para, simpara, title.

See Also

author, authorgroup, collab, collabname, contrib, corpauthor, editor, othercredit, personblurb, personname.

Examples

For examples, see `authorgroup`.

■ authorgroup

Name

authorgroup – Wrapper for author information when a document has multiple authors or collabarators

Synopsis

Content Model

authorgroup ::= ((`author`|`editor`|`collab`|`corpauthor`|`othercredit`)+)

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The `AuthorGroup` element is a wrapper around multiple authors or other collaborators.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes given very special treatment, especially on title pages or other displayed areas. Sometimes suppressed.

Parents

These elements contain authorgroup: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in authorgroup: `author`, `collab`, `corpauthor`, `editor`, `othercredit`.

See Also

`author`, `authorblurb`, `collab`, `collabname`, `contrib`, `corpauthor`, `editor`, `othercredit`, `personblurb`, `personname`.

Examples

The example below demonstrates `AuthorGroup` and many of the elements of `Author`.

```
<!DOCTYPE authorgroup PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<authorgroup>
  <author>
    <honorific>Dr.</honorific>
    <firstname>Lois</firstname>
    <surname>Common-Demoninator</surname>
    <affiliation>
      <shortaffil>Director, M. Behn School of Coop. Eng.</shortaffil>
      <jobtitle>Director of Cooperative Efforts</jobtitle>
      <orgname>The Marguerite Behn International School of
```

```
                Cooperative Engineering</orgname>
    </affiliation>
  </author>

  <editor>
    <firstname>Peter</firstname>
    <surname>Parker</surname>
    <lineage>Sr.</lineage>
    <othername>Spiderman</othername>
    <authorblurb>
      <para>
      Peter's a super hero in his spare time.
      </para>
    </authorblurb>
  </editor>
</authorgroup>
```

For additional examples, see also `bibliography`, `bookinfo`, `collab`.

■ authorinitials

Name

authorinitials – The initials or other short identifier for an author

Synopsis

Mixed Content Model

authorinitials ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%docinfo.char.class;` `%info.class;`

`%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;`

`%tbl.entry.mdl;` `%title.char.mix;`

Description

Author initials occur most frequently in a `Revision` or `Comment`.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain authorinitials: `appendixinfo`, `application`, `articleinfo`, `attribution`, `biblioentry`, `bibliographyinfo`, `bibliomisc`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `bridgehead`, `chapterinfo`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glossaryinfo`, `glosssee`, `glossseealso`, `glossterm`, `indexinfo`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `objectinfo`, `olink`, `para`, `partinfo`, `phrase`, `prefaceinfo`, `primary`, `primaryie`, `productname`, `programlisting`, `quote`, `refentryinfo`, `refentrytitle`, `referenceinfo`, `refpurpose`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `remark`, `revision`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `setindexinfo`, `setinfo`, `sidebarinfo`, `simpara`, `subtitle`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in authorinitials: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

Examples

For examples, see article, revhistory.

■ beginpage

Name

beginpage – The location of a page break in a print version of the document

Synopsis

Content Model

beginpage ::= EMPTY

Attributes Common attributes

Name Type Default

pagenum CDATA *None*

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%cptr.char.mix; %divcomponent.mix; %example.mix;
%figure.mix; %glossdef.mix; %indexdivcomponent.mix;
%legalnotice.mix; %listpreamble.mix; %para.char.mix;
%refcomponent.mix; %refinline.char.mix; %sidebar.mix;
%smallcptr.char.mix; %tbl.entry.mdl; %word.char.mix;

Description

The BeginPage element marks the location of an actual page break in a print version of the document, as opposed to where a page break might appear in a further rendition of the document. This information may be used, for example, to allow support staff using an online system to coordinate with a user referring to a page number in a printed manual.

Processing expectations

The break identified by BeginPage may be displayed in an online version of the document or used for legacy purposes, but it is not expected to cause a page break when the document is processed by an SGML system.

Parents

These elements contain beginpage: abbrev, accel, acronym, action, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, classname, classsynopsisinfo, command, computeroutput, constant, constraintdef, database, dedication, emphasis, entry, envar, errorcode, errorname, errortext, errortype, example, exceptionname, figure, filename, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, important, index, indexdiv, informalexample, informalfigure, initializer, interface, interfacename, itemizedlist, keycap, keycode, keysym, label, legalnotice, lineannotation, link, listitem, literal, literallayout, lot, lotentry, manvolnum, markup, medialabel, member, methodname, modifier, mousebutton, msgaud, msgexplan, msglevel, msgorig, msgtext, note, olink, option, optional, orderedlist, para, parameter, part, partintro, phrase,

preface, procedure, productname, programlisting, prompt, property, quote, refentry, refentrytitle, reference, refpurpose, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, returnvalue, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sgmltag, sidebar, simpara, simplesect, step, structfield, structname, symbol, synopsis, systemitem, term, tip, toc, tocback, tocentry, tocfront, token, type, ulink, userinput, variablelist, varname, warning, wordasword.

Attributes

pagenum PageNum contains the page number of the page in the printed document which begins at the location of the BeginPage element.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter id="ch-publish">
<chapterinfo>
<beginpage pagenum="64"/> <!-- first edition -->
</chapterinfo>
<title>Publishing DocBook Documents</title>

<para>Creating and editing &SGML;/&XML; documents is usually only half the
battle. After you've composed your document, you'll want to publish
it. Publishing, for our purposes, means either print or web
publishing. For &SGML; and &XML; documents, this is usually
accomplished with some kind of <glossterm>stylesheet</glossterm>.  In
the (not too distant) future, you may be able to publish an &XML;
document on the Web by simply putting it online with a stylesheet, but
for now you'll probably have to translate your document into &HTML;.
</para>

<!-- ... -->
</chapter>
```

■ bibliocoverage

Name

bibliocoverage – The spatial or temporal coverage of a document

Synopsis

Mixed Content Model

bibliocoverage ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Name	Type	Default
otherspatial	CDATA	*None*
spatial	dcmibox dcmipoint iso3166 otherspatial tgn	*None*

dcmiperiod
temporal othertemporal
w3c-dtf
None

othertemporal CDATA *None*

Parameter Entities

`%bibliocomponent.mix; %info.class;`

Description

The `bibliocoverage` element satisfies coverage element of the Dublin Core Metadata Initiative `http://dublincore.org/`.

The Dublin Core defines coverage as "the extent or scope of the content of the resource." It goes on to say:

> Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity).
>
> Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names `http://www.getty.edu/research/tools/vocabulary/tgn/` [TGN]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges.

DocBook 4.2 added `bibliocoverage`, `bibliorelation`, and `bibliosource` to make the DocBook meta-information wrappers a complete superset of the Dublin Core.

Processing expectations

Formatted inline. Sometimes suppressed.

This element is used for both spatial and temporal coverage, but the intent is that only one is specified at a time. In other words, on any given instance of the `bibliocoverage` element, specify either a spatial or a temporal encoding, but not both.

Parents

These elements contain bibliocoverage: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in bibliocoverage: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Attributes

spatial Identifies the encoding scheme for the supplied spatial identifier. See the Dublin Core Qualifiers `http://dublincore.org/documents/dcmes-qualifiers/` for details about the meaning of each encoding.

otherspatial Used to identify the encoding scheme when spatial is set to `otherspatial`.

temporal Identifies the encoding scheme for the supplied temporal identifier. See the Dublin Core Qualifiers `http://dublincore.org/documents/dcmes-qualifiers/` for details about the meaning of each encoding.

othertemporal Used to identify the encoding scheme when temporal is set to `othertemporal`.

■ bibliodiv

Name

bibliodiv – A section of a `Bibliography`

Synopsis

Content Model

bibliodiv ::= ((`title`,`subtitle`?,`titleabbrev`?)?, (`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)*, (`biblioentry`|`bibliomixed`)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

`BiblioDiv` is a section of a `Bibliography`. A bibliography might be divided into sections in order to group different types of sources together, like books, journal articles, web sites, and so on.

A bibliography may contain any number of `BiblioEntry` or `BiblioMixed` elements or any number of `BiblioDiv`s, but it cannot contain a mixture of both at the same level.

Processing expectations

Formatted as a displayed block. Some systems may display only those entries within a `BiblioDiv` that are cited in the containing document. This may be an interchange issue. See Appendix F.

Parents

These elements contain bibliodiv: `bibliography`.

Children

The following elements occur in bibliodiv: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `biblioentry`, `bibliomixed`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `BiblioDiv`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

Examples

For examples, see bibliography.

■ biblioentry

Name

biblioentry – An entry in a Bibliography

Synopsis

Content Model

biblioentry ::= ((articleinfo| (abbrev|abstract|address|artpagenums|author|authorgroup| authorinitials|bibliomisc|biblioset|collab|confgroup| contractnum|contractsponsor| copyright|corpauthor|corpname|date| edition|editor|invpartnumber|isbn|issn| issuenum|orgname| biblioid|citebiblioid|bibliosource|bibliorelation| bibliocoverage| othercredit|pagenums|printhistory|productname| productnumber|pubdate|publisher| publishername|pubsnumber| releaseinfo|revhistory|seriesvolnums|subtitle| title| titleabbrev|volumenum|citetitle|personname|honorific|firstname| surname| lineage|othername|affiliation|authorblurb|contrib| indexterm))+)

Attributes Common attributes

Description

A BiblioEntry is an entry in a Bibliography. The contents of BiblioEntry is a "database" of named fields. Presentation systems frequently suppress some elements in a BiblioEntry.

Processing expectations

Formatted as a displayed block.

BiblioEntrys are "raw." They contain a database-like collection of named fields. It is the responsibility of the processing system to select elements from within a BiblioEntry, present them in the correct order, and add all punctuation.

There is no expectation that a system will present all of the fields in a BiblioEntry or that they will be displayed in the order in which they occur.

Correct formatting of BiblioEntrys is an interchange issue. See Appendix F.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of BiblioEntry in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

BookBiblio will be discarded.

Future Changes

ArticleInfo will be dropped from the content model of BiblioEntry.

Parents

These elements contain biblioentry: bibliodiv, bibliography.

Children

The following elements occur in biblioentry: abbrev, abstract, address, affiliation, articleinfo, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, honorific, indexterm, invpartnumber, isbn, issn, issuenum, lineage, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subtitle, surname, title, titleabbrev, volumenum.

See Also

bibliomisc, bibliomixed, bibliomset, biblioset.

Examples

For examples, see bibliography, biblioset.

■ bibliography

Name

bibliography – A bibliography

Synopsis

Content Model

bibliography ::= (bibliographyinfo?, (title,subtitle?,titleabbrev?)?, (calloutlist| glosslist|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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)*, (bibliodiv+| (biblioentry|bibliomixed)+))

Attributes Common attributes

Name Type Default

status CDATA *None*

Parameter Entities

%nav.class; %partcontent.mix;

Description

A bibliography. A DocBook bibliography may contain some preferatory matter, but its main content is a set of bibliography entries (either BiblioEntry or BiblioMixed). These may occur inside BiblioDivs, instead of appearing directly in the Bibliograpy.

Processing expectations

Formatted as a displayed block. A Bibliography in a Book frequently causes a forced page break in print media.

Some systems may display only those entries within a Bibliography that are cited in the containing document. This may be an interchange issue. See Appendix F.

The two styles of bibliography entry have quite different processing expectations. BiblioEntrys are "raw;" they contain a database-like collection of named fields. BiblioMixed entries are "cooked;" the fields occur in the order in which they will be presented and additional punctuation may be sprinkled between the fields.

See BiblioEntry and BiblioMixed for further discussion.

Parents

These elements contain bibliography: appendix, article, book, chapter, glossary, part, preface, sect1, sect2, sect3, sect4, sect5, section.

Children

The following elements occur in bibliography: abstract, address, anchor, authorblurb, beginpage, bibliodiv, biblioentry, bibliographyinfo, bibliomixed, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

status Status identifies the editorial or publication status of the Bibliography.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

Examples

The entries in a Bibliography come in two general forms, "raw" and "cooked". A raw entry is a database-like collection of named fields:

```
<!DOCTYPE bibliography PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<bibliography>
<title>A Test Bibliography</title>

<bibliodiv><title>Books</title>

<biblioentry>
  <abbrev>AhoSethiUllman96</abbrev>
  <authorgroup>
    <author><firstname>Alfred V.</firstname><surname>Aho</surname></author>
    <author><firstname>Ravi</firstname><surname>Sethi</surname></author>
    <author><firstname>Jeffrey D.</firstname><surname>Ullman</surname></author>
  </authorgroup>
  <copyright><year>1996</year>
             <holder>Bell Telephone Laboratories, Inc.</holder></copyright>
  <editor><firstname>James T.</firstname><surname>DeWolf</surname></editor>
  <isbn>0-201-10088-6</isbn>
  <publisher>
    <publishername>Addison-Wesley Publishing Company</publishername>
  </publisher>
  <title>Compilers, Principles, Techniques, and Tools</title>
</biblioentry>
```

```
<biblioentry xreflabel="Kites75">
  <authorgroup>
    <author><firstname>Andrea</firstname><surname>Bahadur</surname></author>
    <author><firstname>Mark</firstname><surname>Shwarek</surname></author>
  </authorgroup>
  <copyright><year>1974</year><year>1975</year>
     <holder>Product Development International Holding N. V.</holder>
     </copyright>
  <isbn>0-88459-021-6</isbn>
  <publisher>
    <publishername>Plenary Publications International, Inc.</publishername>
  </publisher>
  <title>Kites</title>
  <subtitle>Ancient Craft to Modern Sport</subtitle>
  <pagenums>988-999</pagenums>
</biblioentry>

</bibliodiv>
<bibliodiv><title>Periodicals</title>

<biblioentry>
  <abbrev>Walsh97</abbrev>
  <biblioset relation='journal'>
    <title>XML: Principles, Tools, and Techniques</title>
    <publisher>
      <publishername>O'Reilly &amp;amp; Associates, Inc.</publishername>
    </publisher>
    <issn>1085-2301</issn>
    <editor><firstname>Dan</firstname><surname>Connolly</surname></editor>
  </biblioset>
  <biblioset relation='article'>
    <title>A Guide to XML</title>
    <author><surname>Walsh</surname><firstname>Norman</firstname></author>
    <copyright><year>1997</year><holder>ArborText, Inc.</holder></copyright>
    <pagenums>97-108</pagenums>
  </biblioset>
</biblioentry>

</bibliodiv>

</bibliography>
```

A cooked entry is formatted, including additional #PCDATA, so that it is easy to render.

```
<!DOCTYPE bibliography PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<bibliography><title>References</title>

<bibliomixed>
  <bibliomset relation='article'>
    <surname>Walsh</surname>, <firstname>Norman</firstname>.
    <title role='article'>Introduction to Cascading Style Sheets</title>.
  </bibliomset>
  <bibliomset relation='journal'>
    <title>The World Wide Web Journal</title>
    <volumenum>2</volumenum><issuenum>1</issuenum>.
    <publishername>O'Reilly &amp;amp; Associates, Inc.</publishername> and
    <corpname>The World Wide Web Consortium</corpname>.
    <pubdate>Winter, 1996</pubdate></bibliomset>.
```

```
</bibliomixed>

</bibliography>
```

■ bibliographyinfo

Name

bibliographyinfo – Meta-information for a `Bibliography`

Synopsis

Content Model

bibliographyinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `BibliographyInfo` element is a wrapper for a large collection of meta-information about a `Bibliography`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain bibliographyinfo: `bibliography`.

Children

The following elements occur in bibliographyinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

`appendixinfo`, `articleinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `prefaceinfo`, `refsynopsisdivinfo`, `screeninfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setinfo`.

■ biblioid

Name

biblioid – An identifier for a document

Synopsis

Mixed Content Model

biblioid ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Name Type Default

otherclass CDATA *None*

class
- doi
- isbn
- issn
- libraryofcongress
- other
- pubnumber
- uri

None

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

A bibliographic identifier, such as an ISBN number, Library of Congress identifier, or URI.
This element supercedes the `isbn`, `issn`, and `pubsnumber` elements.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain biblioid: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in biblioid: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Attributes

class Identifies the encoding scheme of the identifier.

otherclass Identifies the encoding scheme of the identifier when class is `other`.

See Also

`invpartnumber`, `isbn`, `issn`, `issuenum`, `productnumber`, `pubsnumber`, `seriesvolnums`, `volumenum`.

■ bibliomisc

Name

bibliomisc – Untyped bibliographic information

Synopsis

Mixed Content Model

bibliomisc ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The BiblioMisc element is a wrapper for bibliographic information that does not fit neatly into the other bibliographic fields (such as Author and Publisher).

Processing expectations

Formatted inline. It is recommended that the Role attribute be used to identify the kind of information that this element contains.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain bibliomisc: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in bibliomisc: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject,

interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

biblioentry, bibliomixed, bibliomset, biblioset.

■ bibliomixed

Name

bibliomixed – An entry in a Bibliography

Synopsis

Mixed Content Model

bibliomixed ::= (#PCDATA|abbrev|abstract|address|artpagenums|author|authorgroup| authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum| contractsponsor| copyright|corpauthor|corpname|date|edition| editor|invpartnumber|isbn|issn| issuenum|orgname|biblioid| citebiblioid|bibliosource|biblierelation|bibliocoverage| othercredit|pagenums|printhistory|productname|productnumber| pubdate|publisher| publishername|pubsnumber|releaseinfo| revhistory|seriesvolnums|subtitle| title|titleabbrev|volumenum| citetitle|personname|honorific|firstname|surname| lineage| othername|affiliation|authorblurb|contrib|indexterm|bibliomset)*

Attributes Common attributes

Description

BiblioMixed is an entry in a Bibliography. The contents of BiblioMixed includes all necessary punctuation for formatting. Presentation systems usually display all of the elements in a BiblioMixed.

Processing expectations

Formatted as a displayed block.

BiblioMixed entries are "cooked." In addition to named fields, they can contain interspersed #PCDATA to provide punctuation and other formatting information.

The processing system is generally expected to present each and every element in the entry, and all interspersed #PCDATA, in the order in which it occurs.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of BiblioEntry in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Future Changes

BiblioSet will be removed from the content model of BiblioMixed. Allowing a "raw" container inside a "cooked" one confuses processing expectations.

Parents

These elements contain bibliomixed: bibliodiv, bibliography.

Children

The following elements occur in bibliomixed: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliomset, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, honorific, indexterm, invpartnumber, isbn, issn, issuenum, lineage, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subtitle, surname, title, titleabbrev, volumenum.

See Also

biblioentry, bibliomisc, bibliomset, biblioset.

Examples

For examples, see bibliography, bibliomset.

■ bibliomset

Name

bibliomset – A “cooked” container for related bibliographic information

Synopsis

Mixed Content Model

bibliomset ::= (#PCDATA|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm|bibliomset)*

Attributes Common attributes

Name Type Default

relation CDATA *None*

Description

BiblioMSet is a “cooked” wrapper for a collection of bibliographic information.

The purpose of this wrapper is to assert the relationship that binds the collection. For example, in a BiblioMixed entry for an article in a journal, you might use two BiblioMSets to wrap the fields related to the article and the fields related to the journal.

Processing expectations

Formatted as a displayed block.

BiblioMSets are “cooked.” In addition to named fields, they can contain interspersed #PCDATA to provide punctuation and other formatting information.

The processing system is generally expected to present each and every element in the set, all interspersed #PCDATA, in the order in which it occurs.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of BiblioMSet in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Future Changes

`BiblioSet` will be removed from the content model of `BiblioMSet`. Allowing a “raw” container inside a “cooked” one confuses processing expectations.

Parents

These elements contain bibliomset: `bibliomixed`, `bibliomset`.

Children

The following elements occur in bibliomset: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliomset`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `lineage`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

Attributes

relation Relation identifies the relationship between the various elements in the `BiblioMSet`.

See Also

`biblioentry`, `bibliomisc`, `bibliomixed`, `biblioset`.

Examples

```
<!DOCTYPE bibliomixed PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<bibliomixed>
  <bibliomset relation='article'>
    <surname>Walsh</surname>, <firstname>Norman</firstname>.
    <title role='article'>Introduction to Cascading Style Sheets</title>.
  </bibliomset>
  <bibliomset relation='journal'>
    <title>The World Wide Web Journal</title>
    <volumenum>2</volumenum><issuenum>1</issuenum>.
    <publishername>O'Reilly &amp;amp; Associates, Inc.</publishername> and
    <corpname>The World Wide Web Consortium</corpname>.
    <pubdate>Winter, 1996</pubdate></bibliomset>.
</bibliomixed>
```

For additional examples, see also `bibliography`.

■ bibliorelation

Name

bibliorelation – The relationship of a document to another

Synopsis

Mixed Content Model

bibliorelation ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Name	Type	Default
class	doi isbn issn libraryofcongress other pubnumber uri	*None*
type	hasformat haspart hasversion isformatof ispartof isreferencedby isreplacedby isrequiredby isversionof othertype references replaces requires	*None*
otherclass	CDATA	*None*
othertype	CDATA	*None*

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The `bibliorelation` element satisfies the relation element of the Dublin Core Metadata Initiative `http://dublincore.org/`.

The Dublin Core defines relation as "a reference to a related resource." It goes on to note that "recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system."

DocBook 4.2 added `bibliocoverage`, `bibliorelation`, and `bibliosource` to make the DocBook meta-information wrappers a complete superset of the Dublin Core.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain bibliorelation: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in bibliorelation: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Attributes

class Identifies the encoding scheme of the identifier.

otherclass Identifies the encoding scheme of the identifier when class is `other`.

type Identifies the relationship type of the identifier. See the Dublin Core Qualifiers `http://dublincore.org/documents/dcmes-qualifiers/` for details about the relationship types.

othertype Identifies the relationship type of the identifier when type is `othertype`.

■ biblioset

Name

biblioset – A "raw" container for related bibliographic information

Synopsis

Content Model

biblioset ::= ((`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Name Type Default

relation CDATA *None*

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

`BiblioSet` is a "raw" wrapper for a collection of bibliographic information.

The purpose of this wrapper is to assert the relationship that binds the collection. For example, in a `BiblioEntry` for an article in a journal, you might use two `BiblioSet`s to wrap the fields related to the article and the fields related to the journal.

Processing expectations

Formatted as a displayed block.

`BiblioSet`s are "raw." They contain a database-like collection of named fields. It is the responsibility of the processing system to select elements from within a `BiblioSet`, present them in the correct order, and add all punctuation.

There is no expectation that a system will present all of the fields in a `BiblioSet` or that they will be displayed in the order in which they occur.

Correct formatting of `BiblioSet`s is an interchange issue. See Appendix F.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of BiblioSet in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain biblioset: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in biblioset: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, biblioorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, honorific, indexterm, invpartnumber, isbn, issn, issuenum, lineage, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumbe pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subtitle, surname, title, titleabbrev, volumenum.

Attributes

relation Relation identifies the relationship between the various elements in the BiblioSet.

See Also

biblioentry, bibliomisc, bibliomixed, bibliomset.

Examples

```
<!DOCTYPE biblioentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<biblioentry>
  <abbrev>Walsh97</abbrev>
  <biblioset relation='journal'>
    <title>XML: Principles, Tools, and Techniques</title>
    <publisher>
      <publishername>O'Reilly &amp;amp; Associates, Inc.</publishername>
    </publisher>
    <issn>1085-2301</issn>
    <editor><firstname>Dan</firstname><surname>Connolly</surname></editor>
  </biblioset>
  <biblioset relation='article'>
    <title>A Guide to XML</title>
    <author><surname>Walsh</surname><firstname>Norman</firstname></author>
    <copyright><year>1997</year><holder>ArborText, Inc.</holder></copyright>
    <pagenums>97-108</pagenums>
  </biblioset>
</biblioentry>
```

For additional examples, see also bibliography.

■ bibliosource

Name

bibliosource – The source of a document

Synopsis

Mixed Content Model

bibliosource ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Name Type Default

otherclass CDATA *None*

class
- doi
- isbn
- issn
- libraryofcongress
- other
- pubnumber
- uri

None

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The `bibliosource` element satisfies the source element of the Dublin Core Metadata Initiative http://dublincore.org/.

The Dublin Core defines source as "a reference to a resource from which the present resource is derived." It goes on to note that "the present resource may be derived from the source resource in whole or in part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system."

DocBook 4.2 added `bibliocoverage`, `biblioreleation`, and `bibliosource` to make the DocBook meta-information wrappers a complete superset of the Dublin Core.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain bibliosource: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in bibliosource: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Attributes

class Identifies the encoding scheme of the identifier.

otherclass Identifies the encoding scheme of the identifier when class is `other`.

■ blockinfo

Name

blockinfo – Meta-information for a block element

Synopsis

Content Model

blockinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Parameter Entities

`%tbl.table.mdl;`

Description

The `blockinfo` element is a wrapper for a large collection of meta-information about a block element. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain blockinfo: `equation`, `example`, `figure`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `legalnotice`, `msgset`, `orderedlist`, `procedure`, `qandadiv`, `qandaentry`, `qandaset`, `table`, `variablelist`.

Children

The following elements occur in blockinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

appendixinfo, articleinfo, bibliographyinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

■ blockquote

Name

blockquote – A quotation set off from the main text

Synopsis

Content Model

blockquote ::= (title?,attribution?, (calloutlist|glosslist|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|msgset|procedure| sidebar|qandaset|productionset| constraintdef|anchor|bridgehead|remark|highlights| abstract| authorblurb|epigraph|indexterm|beginpage)+)

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%divcomponent.mix; %example.mix; %figure.mix;

%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;

%informal.class; %legalnotice.mix; %listpreamble.mix;

%para.mix; %qandaset.mix; %refcomponent.mix;

%revdescription.mix; %sidebar.mix; %textobject.mix;

Description

Block quotations are set off from the main text, as opposed to occurring inline.

Processing expectations

Formatted as a displayed block.

Future Changes

Epigraph will not be allowed in BlockQuote in DocBook V4.0.

Parents

These elements contain blockquote: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in blockquote: abstract, address, anchor, attribution, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, tip, title, variablelist, warning.

See Also

abstract, attribution, epigraph, highlights, sidebar.

Examples

```
<!DOCTYPE blockquote PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<blockquote><attribution>Richard Dawkins</attribution>
<para>
The universe that we observe has precisely the properties we should
expect if there is, at bottom, no design, no purpose, no evil and
no good, nothing but pitiless indifference.
</para>
</blockquote>
```

> The universe that we observe has precisely the properties we should expect if there is, at bottom, no design, no purpose, no evil and no good, nothing but pitiless indifference.
>
> —Richard Dawkins

For additional examples, see also attribution, literallayout.

■ book

Name

book – A book

Synopsis

Content Model

book ::= ((title,subtitle?,titleabbrev?)?, bookinfo?, (dedication|toc|lot|glossary|bibliography|preface|chapter|reference|part|article|appendix|index|setindex|colophon)*)

Attributes Common attributes

Name Type Default

fpi CDATA *None*

status CDATA *None*

label CDATA *None*

Parameter Entities

%book.class;

Description

A complete book. This is probably the most common document starting point in DocBook documents. The content model of Book was made dramatically less restrictive in DocBook V3.1.

Processing expectations

Formatted as a displayed block. Generally causes a forced page break, restarts page numbering, and may generate additional front and back matter (tables of contents and indexes, for example) automatically. In a `Set`, `Book` almost always begins on the next available recto page.

The input order of major components is taken to be the desired output order.

Parents

These elements contain book: `set`.

Children

The following elements occur in book: `appendix`, `article`, `bibliography`, `bookinfo`, `chapter`, `colophon`, `dedication`, `glossary`, `index`, `lot`, `part`, `preface`, `reference`, `setindex`, `subtitle`, `title`, `titleabbrev`, `toc`.

Attributes

fpi FPI holds the Formal Public Identifier for the `Book`.

label Label specifies an identifying string for presentation purposes. The third `Book` in a `Set` might be labeled "Volume III", for example.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the `Book`.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

`appendix`, `article`, `chapter`, `colophon`, `dedication`, `part`, `partintro`, `preface`, `set`.

Examples

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book>
<title>An Example Book</title>
<titleabbrev>Example</titleabbrev>
<bookinfo>
  <legalnotice><para>No notice is required.</para></legalnotice>
  <author><firstname>Norman</firstname><surname>Walsh</surname></author>
</bookinfo>
<dedication>
<para>
This book is dedicated to you.
</para>
</dedication>
<preface><title>Foreword</title>
<para>
Some content is always required.
</para>
```

```
</preface>
<chapter><title>Required Chapter</title>
<para>
At least one chapter, reference, part, or article is required in a book.
</para>
</chapter>
<appendix><title>Optional Appendix</title>
<para>
Appendixes are optional.
</para>
</appendix>
</book>
```

For additional examples, see also `appendix`, `collab`, `set`, `xref`.

■ bookinfo

Name

bookinfo – Meta-information for a Book

Synopsis

Content Model

bookinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Name Type Default

contents IDREFS *None*

Description

Meta-information for a Book.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `BookInfo` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors. `BookBiblio` will be discarded.

Future Changes

The contents attribute will be removed.

Parents

These elements contain bookinfo: `book`.

Children

The following elements occur in bookinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

Attributes

contents Contents, if specified, should contain a list of all the IDs of the chapter-level subelements of the Book, in their natural order.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

Examples

```
<!DOCTYPE bookinfo PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<bookinfo>
  <title>User's Guide for the DocBook DTD</title>
  <authorgroup>
    <author><firstname>Terry</firstname><surname>Allen</surname></author>
    <author><firstname>Eve</firstname><surname>Maler</surname>
      <affiliation><orgname>Arbortext, Inc.</orgname></affiliation>
    </author>
    <author><firstname>Norman</firstname><surname>Walsh</surname>
      <affiliation><orgname>Arbortext, Inc.</orgname></affiliation>
    </author>
  </authorgroup>
  <edition>User's Guide version 1.0 for DocBook V3.0</edition>
  <pubdate>1997</pubdate>
  <copyright><year>1992</year>
    <year>1993</year>
    <year>1994</year>
    <year>1995</year>
    <year>1996</year>
    <year>1997</year>
<holder>Arbortext, Inc.,
HaL Computer Systems, Inc.,
Fujitsu Software Corp., and
O'Reilly &amp;amp; Associates, Inc.
</holder>
</copyright>

<legalnotice>
<para>Permission to use, copy, modify and distribute
the DocBook DTD and its accompanying documentation for any purpose and
without fee is hereby granted in perpetuity, provided that the above
copyright notice and this paragraph appear in all copies.</para>
```

```
</legalnotice>

<legalnotice>
<para>The copyright holders make no representation about the suitability of
this DTD for any purpose. It is provided
<quote>as is</quote> without expressed
or implied warranty.  If you modify the DocBook DTD in any way, except for
declaring and referencing additional general entities and declaring additional
notations, identify your DTD as a variant of DocBook.</para>
</legalnotice>

</bookinfo>
```

For additional examples, see also book, collab.

■ bridgehead

Name

bridgehead – A free-floating heading

Synopsis

Mixed Content Model

bridgehead ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|indexterm)*

Attributes Common attributes

Name Type Default

renderas	other sect1 sect2 sect3 sect4 sect5	*None*

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %genobj.class; %listpreamble.mix;
%qandaset.mix; %refcomponent.mix; %revdescription.mix;
%sidebar.mix;

Description

Some documents, usually legacy documents, use headings that are not tied to the normal sectional hiearchy. These headings may be represented in DocBook with the BridgeHead element.

BridgeHeads may also be useful in fiction or journalistic works that don't have a nested hierarchy.

Processing expectations

A BridgeHead is formatted as a block, using the same display properties as the section heading which it masquerades as. The RenderAs attribute controls which heading it mimics.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain bridgehead: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, glossary, glossdiv, important, index, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in bridgehead: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

renderas The RenderAs attribute identifies how the BridgeHead should be rendered. In this way, a BridgeHead can be made to appear as a Sect1, for example.

See Also

sect1, sect2, sect3, sect4, sect5, section, simplesect.

Examples

For examples, see chapter.

■ callout

Name

callout – A “called out” description of a marked Area

Synopsis

Content Model

callout ::= ((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb| epigraph|indexterm|beginpage)+)

Attributes Common attributes

Name Type Default

arearefs IDREFS *Required*

Description

A "callout" is a visual device for associating annotations with an image, program listing, or similar figure. Each location is identified with a mark, and the annotation is identified with the same mark. This is somewhat analagous to the notion of footnotes in print.

An example will help illustrate the concept. In the following example, the synopsis for the **mv** command is annotated with two marks. Note the location of the old and new filenames.

```
mv oldfile newfile
```

Somewhere else in the document, usually close by, a CalloutList provides a description for each of the callouts: co-oldname The old filename. The **mv** command renames the file currently called *oldfile*, which must exist when **mv** is executed. co-newname The new filename. The **mv** command changes the name of the old file to *newfile*. If *newfile* exists when **mv** is executed, it will be replaced by the old file.

Each Callout contains an annotation for an individual callout or a group of callouts. The Callout points to the areas that it annotates with ID references. The areas are identified by coordinates in an an Area or AreaSet, or by an explicit CO element.

Processing expectations

Formatted as a displayed block.

CallOuts usually generate text that points the reader to the appropriate area on the object being augmented. Often, these are numbered bullets or other distinct visual icons. The same icons should be used in both places. In other words, whatever identifies the callouts on the object should generate the same icons on the respective callouts.

In online environments, it may also be possible to establish a linking relationship between the two elements.

The processing expectations of Callouts are likely to deserve special consideration for interchange. See Appendix F. This is especially true if your interchange partners are producing documentation in a medium that has restricted visual presentation features, such as aural media or Braille.

Parents

These elements contain callout: calloutlist.

Children

The following elements occur in callout: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark,

screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, tip, variablelist, warning.

Attributes

arearefs AreaRefs must point to one or more callouts. Callouts can be identified with Area or AreaSet elements in a GraphicCO, MediaObjectCO, ProgramListingCO, or ScreenCO element or with a simple CO element in a number of other environments. These callouts identify the portions of the object described by this Callout.

Examples

For examples, see programlistingco, screenco.

■ calloutlist

Name

calloutlist – A list of Callouts

Synopsis

Content Model

calloutlist ::= ((title,titleabbrev?)?, callout+)

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %footnote.mix;
%glossdef.mix; %highlights.mix; %legalnotice.mix;
%list.class; %para.mix; %qandaset.mix;
%refcomponent.mix; %revdescription.mix; %sidebar.mix;
%tabentry.mix; %tbl.entry.mdl; %textobject.mix;

Description

A CalloutList is a list of annotations or descriptions. Each Callout points to the area on a Graphic, ProgramListing, or Screen that it augments.

The areas are identified by coordinates in an an Area or AreaSet, or by an explicit CO element.

Processing expectations

Formatted as a displayed block.

Future Changes

Introductory material may appear before the first list item.

Parents

These elements contain calloutlist: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, footnote, glossary, glossdef, glossdiv, graphicco, highlights, imageobjectco, important, index, informalexample, legalnotice, listitem, msgexplan, msgtext, note, para, partintro, preface, procedure, programlistingco, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, screenco, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, warning.

Children

The following elements occur in calloutlist: `callout`, `title`, `titleabbrev`.

See Also

`areaspec`, `co`, `coref`, `graphicco`, `imageobjectco`, `itemizedlist`, `listitem`, `mediaobjectco`, `orderedlist`, `programlistingco`, `screenco`, `segmentedlist`, `simplelist`, `variablelist`.

Examples

For examples, see `programlistingco`, `screenco`.

■ caption

Name

caption – A caption

Synopsis

Content Model

caption ::= (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`| `simplelist`| `variablelist`|`caution`|`important`|`note`|`tip`|`warning`| `literallayout`|`programlisting`| `programlistingco`|`screen`|`screenco`| `screenshot`|`formalpara`|`para`|`simpara`|`blockquote`)*

Attributes Common attributes

Description

A `Caption` is an extended description of a `MediaObject`. Unlike a `TextObject`, which is an alternative to the other elements in the `MediaObject`, the `Caption` augments the object.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain caption: `mediaobject`.

Children

The following elements occur in caption: `blockquote`, `calloutlist`, `caution`, `formalpara`, `glosslist`, `important`, `itemizedlist`, `literallayout`, `note`, `orderedlist`, `para`, `programlisting`, `programlistin` `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `tip`, `variablelist`, `warning`.

See Also

`alt`, `audioobject`, `graphic`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

For examples, see `imageobject`, `informalfigure`.

■ caution

Name

caution – A note of caution

Synopsis

Content Model

caution ::= (`title`?, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`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`|`sidebar`|`anchor`|`bridgehead`|`remark`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Parameter Entities

`%admon.class;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%highlights.mix;` `%legalnotice.mix;`

`%listpreamble.mix;` `%para.mix;` `%qandaset.mix;`

`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`

`%tabentry.mix;` `%tbl.entry.mdl;` `%textobject.mix;`

Description

A `Caution` is an admonition, usually set off from the main text.

In some types of documentation, the semantics of admonitions are clearly defined (`Caution` might imply the possibility of harm to equipment whereas `Warning` might imply harm to a person) However, DocBook makes no such assertions.

Processing expectations

Formatted as a displayed block. Often outputs the generated text “Caution” or some other visible indication of the type of admonition, especially if a `Title` is not present. Sometimes outputs a graphical icon or other symbol as well.

Parents

These elements contain caution: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `chapter`, `colophon`, `dedication`, `entry`, `glossary`, `glossdiv`, `highlights`, `index`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `variablelist`.

Children

The following elements occur in caution: `address`, `anchor`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `orderedlist`, `para`, `procedure`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `synopsis`, `table`, `title`, `variablelist`.

See Also

`important`, `note`, `tip`, `warning`.

Examples

```
<!DOCTYPE caution PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<caution><title>No User Servicable Parts Inside</title>
<para>
Breaking this seal voids all warranties.
</para>
</caution>
```

NO USER SERVICABLE PARTS INSIDE

Breaking this seal voids all warranties.

■ chapter

Name

chapter – A chapter, as of a book

Synopsis

Content Model

chapter ::= (`beginpage`?,`chapterinfo`?, (`title`,`subtitle`?,`titleabbrev`?), (`toc`|`lot`|`index`|`glossary`|`bibliography`)*, `tocchap`?, (((`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)+, (`sect1`*|(`refentry`)*|`simplesect`*|(`section`)*))|(`sect1`+|(`refentry`)+|`simplesect`+|(`section`)+)), (`toc`|`lot`|`index`|`glossary`|`bibliography`)*)

Attributes Common attributes

Name Type Default

status CDATA *None*

label CDATA *None*

Parameter Entities

`%chapter.class;` `%partcontent.mix;`

Description

`Chapter` is a chapter of a `Book`.

Processing expectations

Formatted as a displayed block. Usually introduces a forced page break and often starts on the next recto page. The first chapter of a document usually restarts page numbering. Typically, chapters are numbered and presented in the table of contents.

Parents

These elements contain chapter: `book`, `part`.

Children

The following elements occur in chapter: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `bibliography`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `chapterinfo`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glossary`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `index`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `lot`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refentry`, `remark`, `screen`, `screenco`, `screenshot`, `sect1`, `section`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `simplesect`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `toc`, `tocchap`, `variablelist`, `warning`.

Attributes

label Label specifies an identifying string for presentation purposes. The fourth `Chapter` in a `Book` might be labeled “4”, for example.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the `Chapter`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`appendix`, `article`, `book`, `colophon`, `dedication`, `part`, `partintro`, `preface`, `set`.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter label="6" id="figures">
<!--beginpage pagenum='129'-->
<chapterinfo>
<keywordset>
  <keyword>images</keyword>
  <keyword>illustrations</keyword>
</keywordset>
<itermset>
  <indexterm zone="figures"><primary>Figures</primary></indexterm>
  <indexterm zone="figures"><primary>Pictures</primary></indexterm>
  <indexterm zone="notreal">
    <primary>Sections</primary><secondary>Not Real</secondary>
  </indexterm>
```

```
</itermset>
</chapterinfo>
<title>Pictures and Figures</title>
<epigraph>
<attribution>William Safire</attribution>
<para>
Knowing how things work is the basis for appreciation, and is
thus a source of civilized delight.
</para>
</epigraph>
<para>
Pictures and figures &hellip;
</para>
<sect1><title>Top Level Section</title>
<para>
&hellip;
</para>
<bridgehead id="notreal" renderas='sect3'>Not a Real Section</bridgehead>
<para>
This paragraph appears to be under a Sect3 heading, but it's really
in the same Sect1 as the preceding paragraph.
</para>
</sect1>
</chapter>
```

For additional examples, see also `abstract`, `appendix`, `beginpage`, `book`, `collab`, `highlights`, `indexterm`, `olink`, `part`, `section`, `set`, `titleabbrev`, `xref`.

■ chapterinfo

Name

chapterinfo – Meta-information for a `Chapter`

Synopsis

Content Model

chapterinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblierelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `ChapterInfo` element is a wrapper for a large collection of meta-information about a `Chapter`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain chapterinfo: chapter.

Children

The following elements occur in chapterinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

■ citation

Name

citation – An inline bibliographic reference to another published work

Synopsis

Mixed Content Model

citation ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

The content of a Citation is assumed to be a reference string, perhaps identical to an abbreviation in an entry in a Bibliography.

Processing expectations

Formatted inline.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain citation: `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `para`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `quote`, `refentrytitle`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in citation: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

See Also

`citebiblioid`, `citerefentry`, `citetitle`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Consult <citation>AhoSethiUllman96</citation> for more details on
abstract syntax tree construction.
</para>
```

Consult [**?**] for more details on abstract syntax tree construction.

■ citebiblioid

Name

citebiblioid – A citation of a bibliographic identifier

Synopsis

Mixed Content Model

citebiblioid ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Name Type Default

otherclass CDATA *None*

class
- doi
- isbn
- issn
- libraryofcongress
- other
- pubnumber
- uri

None

Parameter Entities

`%bibliocomponent.mix; %info.class;`

Description

A `citebiblioid` identifies a citation to another work by bibliographic identifier.

Processing expectations

Formatted inline.

Parents

These elements contain citebiblioid: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in citebiblioid: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Attributes

class Identifies the encoding scheme of the identifier.

otherclass Identifies the encoding scheme of the identifier when class is `other`.

See Also

`citation`, `citerefentry`, `citetitle`.

■ citerefentry

Name

citerefentry – A citation to a reference page

Synopsis

Content Model

citerefentry ::= (refentrytitle,manvolnum?)

Attributes Common attributes

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

This element is a citation to a RefEntry. It must include a RefEntryTitle that should exactly match the title of a RefEntry.

Processing expectations

This element implicitly links to the RefEntry with the same RefEntryTitle (in the same volume, as defined by ManVolNum).

Formatted inline. Usually the ManVolNum is put in parentheses.

Parents

These elements contain citerefentry: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in citerefentry: manvolnum, refentrytitle.

See Also

citation, citebiblioid, citetitle.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
For a further description of print formats, consult the
<citerefentry><refentrytitle>printf</refentrytitle>
<manvolnum>3S</manvolnum></citerefentry> manual page.
</para>
```

For a further description of print formats, consult the printf(3S) manual page. For additional examples, see also manvolnum, refentry.

■ citetitle

Name

citetitle – The title of a cited work

Synopsis

Mixed Content Model

citetitle ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

pubwork	article book chapter journal manuscript part refentry section series set	*None*

Parameter Entities

%bibliocomponent.mix; %gen.char.class; %info.class;

%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix;

Description

CiteTitle provides inline markup for the title of a cited work.

Processing expectations

Formatted inline. Often italicized for Books and quoted for Articles.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain citetitle: appendixinfo, application, articleinfo, attribution, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayout, lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo,

remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in citetitle: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

pubwork Pubwork identifies the genre of the cited publication.

See Also

citation, citebiblioid, citerefentry.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
For a complete methodology for DTD creation, see
<citetitle pubwork="book">Developing SGML DTDs: From Text to Model
to Markup</citetitle> by Eve Maler and Jeanne El Andaloussi.
</para>
```

For a complete methodology for DTD creation, see *Developing SGML DTDs: From Text to Model to Markup* by Eve Maler and Jeanne El Andaloussi. For additional examples, see also emphasis, footnote, literallayout, productnumber, ulink.

■ city

Name

city – The name of a city in an address

Synopsis

Mixed Content Model

city ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Description

The name of a city in an `Address`.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain city: `address`.

Children

The following elements occur in city: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `country`, `email`, `fax`, `otheraddr`, `phone`, `pob`, `postcode`, `state`, `street`.

Examples

For examples, see `address`, `otheraddr`.

■ classname

Name

classname – The name of a class, in the object-oriented programming sense

Synopsis

Mixed Content Model

classname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

The `ClassName` tag is used to identify the name of a class. This is likely to occur only in documentation about object-oriented programming systems, languages, and architectures.

DocBook does not contain a complete set of inlines appropriate for describing object-oriented programming environments. (While it has `ClassName`, for example, it has nothing suitable for methods.) This will be addressed in a future version of DocBook.

Processing expectations

Formatted inline.

Parents

These elements contain classname: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, ooclass, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in classname: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

interface, property, structfield, structname, symbol, token, type.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
All user-interface components must be descendants of the
<classname>Widget</classname> class.
</para>
```

All user-interface components must be descendants of the Widget class.

■ classsynopsis

Name

classsynopsis – The syntax summary for a class definition

Synopsis

Content Model

classsynopsis ::= ((ooclass|oointerface|ooexception)+, (classsynopsisinfo|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis)*)

Attributes Common attributes

Name Type Default

language CDATA *None*

class class
interface
"class"

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %figure.mix;
%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;
%listpreamble.mix; %para.char.mix; %qandaset.mix;
%refcomponent.mix; %revdescription.mix; %sidebar.mix;
%synop.class; %tbl.entry.mdl;

Description

A ClassSynopsis contains the syntax summary of a class (generally speaking, a class in the object-oriented programming language sense).

This is one of the few places where DocBook attempts to model as well as describe. Unlike FuncSynopsis which was designed with C language function prototypes in mind, the content model of ClassSynopsis was designed to capture a wide range of object-oriented language semantics.

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis. The exception to this rule is that the spacing and other punctuation inside a parameter that is a pointer to a function must be provided in the source markup.

Parents

These elements contain classsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in classsynopsis: classsynopsisinfo, constructorsynopsis, destructorsynopsis, fieldsynopsis, methodsynopsis, ooclass, ooexception, oointerface.

Attributes

class The Class attribute identifies the whether this is the synopsis of a class or an interface.

language Identifies the language (Java, C++, Perl, etc.) of the class for which this is a synopsis allows stylesheets to select an appropriate rendering.

■ classsynopsisinfo

Name

classsynopsisinfo – Information supplementing the contents of a ClassSynopsis

Synopsis

Mixed Content Model

classsynopsisinfo ::= (#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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|

nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage|textobject|lineannotation)*

Attributes Common attributes

Name Type Default

format linespecific "linespecific"

linenumbering numbered unnumbered

None

Description

Supplementary information in a ClassSynopsis. See ClassSynopsis.

Unlike the other info elements, ClassSynopsisInfo is not a container for meta-information. Instead ClassSynopsisInfo is a verbatim environment for adding additional information to a class synopsis.

Processing expectations

This element is displayed “verbatim”; whitespace and linebreaks within this element are significant.

Parents

These elements contain classsynopsisinfo: classsynopsis.

Children

The following elements occur in classsynopsisinfo: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, lineannotation, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, textobject, token, type, ulink, userinput, varname.

Attributes

format The Format attribute applies the linespecific notation to all ClassSynopsisInfos. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of a ClassSynopsisInfo are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

■ cmdsynopsis

Name

cmdsynopsis – A syntax summary for a software command

Synopsis

Content Model

cmdsynopsis ::= ((`command`|`arg`|`group`|`sbr`)+, `synopfragment`*)

Attributes Common attributes

Name Type Default

sepchar CDATA " "

cmdlength CDATA *None*

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%para.char.mix;` `%qandaset.mix;`
`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`
`%synop.class;` `%tbl.entry.mdl;`

Description

A `CmdSynopsis` summarizes the options and parameters of a command started from a text prompt. This is usually a program started from the DOS, Windows, or UNIX shell prompt.

`CmdSynopsis` operates under the following general model: commands have arguments, that may be grouped; arguments and groups may be required or optional and may be repeated.

Processing expectations

The processing expectations of `CmdSynopsis` are fairly complex.

- Arguments are generally identified with a prefix character.

  In the UNIX world, this character is almost universally the dash or hyphen although plus signs and double dashes have become more common in recent years.

  In the DOS/Windows world, forward slashes are somewhat more common than dashes.

  The DocBook processing expectations on this point are intentionally vague. In some environments it may be most convenient to generate these characters automatically, in other environments it may be more convenient to insert them literally in the content.

  Whichever processing model you choose, note that this will be an interchange issue if you share documents with other users (see Appendix F).

- Brackets are used to distinguish between optional, required, or plain arguments. Usually square brackets are placed around optional arguments, `[-g]`, and curly brackets are placed around required arguments, `{-g}`. Plain arguments are required, but are not decorated with brackets.

- Repeatable arguments are followed by an ellipsis.

- Multiple arguments within a group are considered exclusive and are separated by vertical bars.

- Groups, like arguments, may be optional, required, or plain and may or may not repeat. The same brackets and ellipses that are used to indicate these characteristics on arguments are used on groups.

- Arguments and groups may nest more-or-less arbitrarily.

- Formatted as a displayed block. The processing system is free to introduce line breaks where required, but the `SBR` element may be introduced by the author to provide an explicit break location.

Parents

These elements contain cmdsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in cmdsynopsis: arg, command, group, sbr, synopfragment.

Attributes

cmdlength CmdLength indicates displayed length of the command; this information may be used to intelligently indent command synopses which extend beyond one line.

label Label specifies an identifying number or string that may be used in presentation.

sepchar SepChar specifies the character (a space by default) that should separate the Command and its top-level arguments.

See Also

arg, funcsynopsis, group, refsynopsisdiv, sbr, synopfragment, synopfragmentref, synopsis.

Examples

```
<!DOCTYPE cmdsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<cmdsynopsis>
  <command>cd</command>
  <arg choice='req'><replaceable>directory</replaceable></arg>
</cmdsynopsis>
```

cd *directory*

```
<!DOCTYPE cmdsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<cmdsynopsis>
  <command>cal</command>
  <arg>-j</arg>
  <arg>-y</arg>
  <arg>month <arg>year</arg></arg>
</cmdsynopsis>
```

cal [-j] [-y] [month [year]]

```
<!DOCTYPE cmdsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<cmdsynopsis>
  <command>chgrp</command>
  <arg>-R
    <group>
      <arg>-H</arg>
      <arg>-L</arg>
      <arg>-P</arg>
    </group>
  </arg>
  <arg>-f</arg>
  <arg choice='plain'><replaceable>group</replaceable></arg>
  <arg rep='repeat' choice='plain'><replaceable>file</replaceable></arg>
</cmdsynopsis>
```

```
chgrp [-R [-H | -L | -P]] [-f] group file...
```

```
<!DOCTYPE cmdsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<cmdsynopsis>
  <command>emacs</command>
  <arg>-t <replaceable>file</replaceable></arg>
  <arg>-q</arg>
  <arg>-u <replaceable>user</replaceable></arg>
  <arg>+<replaceable>number</replaceable></arg>
  <arg rep='repeat'>-f <replaceable>function</replaceable></arg>
  <sbr/>
  <arg rep='repeat'>-l <replaceable>file</replaceable></arg>
  <arg rep='repeat' choice='plain'><replaceable>file</replaceable></arg>
</cmdsynopsis>
```

```
emacs [-t file] [-q] [-u user] [+number] [-f function...]  [-l file...]
    file...
```

Note the use of SBR in this example to force line breaks at reasonable places in the synopsis.
For additional examples, see also `refentry`, `synopfragment`.

■ CO

Name

co – The location of a callout embedded in text

Synopsis

Content Model

co ::= EMPTY

Attributes Common attributes

Name Type Default

label CDATA *None*

linkends IDREFS *None*

Description

A `CO` identifies (by its location) a point of reference for a callout. See `Callout`.

Processing expectations

Formatted inline.

Parents

These elements contain co: computeroutput, literallayout, programlisting, prompt, replaceable, screen, synopsis, systemitem, userinput.

Attributes

label Label specifies an identifying number or string that may be used in presentation.

linkends Linkends points to the Callout(s) which refer to this CO. (This provides bidirectional linking, which may be useful in online presentation, for example.)

See Also

areaspec, calloutlist, coref, graphicco, imageobjectco, mediaobjectco, programlistingco, screenco.

Examples

For examples, see screenco.

■ collab

Name

collab – Identifies a collaborator

Synopsis

Content Model

collab ::= (collabname,affiliation*)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

This element identifies a collaborative partner in a document. It associates the name of a collaborator with his or her Affiliation.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain collab: appendixinfo, articleinfo, authorgroup, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in collab: `affiliation`, `collabname`.

See Also

`author`, `authorblurb`, `authorgroup`, `collabname`, `contrib`, `corpauthor`, `editor`, `othercredit`, `personblurb`, `personname`.

Examples

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book>
<bookinfo>
  <title>DocBook: The Definitive Guide</title>
  <authorgroup>
    <collab><collabname>Lenny Muellner</collabname></collab>
    <collab><collabname>Norman Walsh</collabname></collab>
  </authorgroup>
</bookinfo>
<chapter><title>Just an Example</title>
<para>
This is just an example, in real life, Lenny and Norm are both
<sgmltag>Author</sgmltag>s.
</para>
</chapter>
</book>
```

For additional examples, see also `contractsponsor`.

■ collabname

Name

collabname – The name of a collaborator

Synopsis

Mixed Content Model

collabname ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

The name of a collaborator.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain collabname: `collab`.

Children

The following elements occur in collabname: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

author, authorblurb, authorgroup, collab, contrib, corpauthor, editor, othercredit, personblurb, personname.

Examples

For examples, see collab, contractsponsor.

■ colophon

Name

colophon – Text at the back of a book describing facts about its production

Synopsis

Content Model

colophon ::= ((title,subtitle?,titleabbrev?)?, (calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist|simplelist|variablelist|caution|important|note|tip|warning|literallayout|programlisting|programlistingco|screen|screenco|screenshot|formalpara|para|simpara|blockquote)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

A Colophon, if present, almost always occurs at the very end of a book. It contains factual information about the book, especially about its production, and includes details about typographic style, the fonts used, the paper used, and perhaps the binding method of the book.

Font geeks like Norm think every book should have one.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain colophon: book.

Children

The following elements occur in colophon: blockquote, calloutlist, caution, formalpara, glosslist, important, itemizedlist, literallayout, note, orderedlist, para, programlisting, programlistin screen, screenco, screenshot, segmentedlist, simpara, simplelist, subtitle, tip, title, titleabbrev, variablelist, warning.

Attributes

status Status identifies the editorial or publication status of the Colophon.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`appendix`, `article`, `book`, `chapter`, `dedication`, `part`, `partintro`, `preface`, `set`.

Examples

```
<!DOCTYPE colophon PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<colophon>
<para>
Draft versions of this book were produced with the
DocBook DSSSL Stylesheets. Final production was
performed with Troff.
</para>
</colophon>
```

■ colspec

Name

colspec – Specifications for a column in a table

Synopsis

Content Model

colspec ::= EMPTY

Attributes

Name Type Default

colnum CDATA *None*

char CDATA *None*

colsep CDATA *None*

colwidth CDATA *None*

charoff CDATA *None*

colname CDATA *None*

rowsep CDATA *None*

align (center | char | justify | left | right) *None*

Parameter Entities

`%tbl.entrytbl.mdl;` `%tbl.hdft.mdl;` `%tbl.tgroup.mdl;`

Description

The attributes of this empty element specify the presentation characteristics of entries in a column of a table.

Each `ColSpec` refers to a single column. Columns are numbered sequentially from left to right in the table. If the ColNum attribute is not specified, the `ColSpec` is for the next column after the preceeding `ColSpec` or column 1 if it is the first `ColSpec`.

Processing expectations

Suppressed. This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`.

Parents

These elements contain colspec: `entrytbl`, `tfoot`, `tgroup`, `thead`.

Attributes

align Align specifies the horizontal alignment of `Entry`s (or `EntryTbl`s) in the column. The default alignment is inherited from the enclosing `TGroup`. If `Char` is specified, see also Char and CharOff. Individual `Entry`s and `EntryTbl`s can specify an alternate alignment.

char Char specifies the alignment character when the Align attribute is set to `Char`.

charoff CharOff specifies the percentage of the column's total width that should appear to the left of the first occurance of the character identified in Char when the Align attribute is set to `Char`. This attribute is inherited from the enclosing `TGroup`.

colname ColName gives a symbolic name to a column. The symbolic name can then be used in subsequent `Entry`s and `SpanSpec`s to identify the column.

colnum ColNum gives the number of the column defined by this `ColSpec`. If not specified, this `ColSpec` describes the next column to the right of the column defined by the previous `ColSpec` or the first column (column 1) if this is the first `ColSpec`.

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of the column described by this `ColSpec`. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the ColSep of the last column in the table is ignored. If unspecified, this attribute is inherited from enclosing elements. Individual `Entry`s or `EntryTbl`s can override the `ColSpec` setting of this attribute.

colwidth ColWidth specifies the desired width of the relevant column. It can be either a fixed measure using one of the CALS units (36pt, 10pc, etc.) or a proportional measure. Proportional measures have the form "*number*`*`", meaning this column should be *number* times wider than a column with the measure "`1*`" (or just "`*`"). These two forms can be mixed, as in "`3*+1pc`".

rowsep If RowSep has the value 1 (true), then a rule will be drawn below the cells in the specified column. A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements. Individual `Entry`s or `EntryTbl`s can override the `ColSpec` setting of this attribute.

See Also

`entry`, `entrytbl`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

For examples, see `informaltable`, `table`.

■ command

Name

command – The name of an executable program or other software command

Synopsis

Mixed Content Model

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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none
refentry

"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

This element holds the name of an executable program or the text of a command that a user enters to execute a program.

Command is an integral part of the CmdSynopsis environment as well as being a common inline.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of Command will be constrained to (#PCDATA | Replaceable | InlineGraphic) in DocBook V4.0.

Parents

These elements contain command: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, cmdsynopsis, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in command: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag,

structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Command.

See Also

constant, literal, replaceable, varname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In <acronym>UNIX</acronym>,
<command>ls</command> is used to get a directory listing.
</para>
```

In UNIX, **ls** is used to get a directory listing. For additional examples, see also cmdsynopsis, medialabel, option, parameter, refentry, synopfragment.

■ computeroutput

Name

computeroutput – Data, generally text, displayed or presented by a computer

Synopsis

Mixed Content Model

computeroutput ::= (#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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| remark|subscript|superscript|inlinegraphic|inlinemediaobject| indexterm|beginpage|co)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

ComputerOutput identifies lines of text generated by a computer program (messages, results, or other output). Note that ComputerOutput is not a verbatim environment, but an inline.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information. It's often presented in a fixed width font.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain computeroutput: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `ComputerOutput`.

See Also

`constant`, `envar`, `filename`, `lineannotation`, `literal`, `literallayout`, `markup`, `option`, `optional`, `parameter`, `programlisting`, `prompt`, `replaceable`, `screen`, `screenshot`, `sgmltag`, `synopsis`, `systemitem`, `userinput`, `varname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The output from the date command,
<computeroutput>Sun  Nov 16, 1997  21:03:29</computeroutput>,
uses fixed-width fields so that it can easily be parsed.
</para>
```

The output from the date command, `Sun Nov 16, 1997 21:03:29`, uses fixed-width fields so that it can easily be parsed.

■ confdates

Name

confdates – The dates of a conference for which a document was written

Synopsis

Mixed Content Model

confdates ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`ConfDates` holds the dates of a conference for which a document was written or at which it was presented.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain confdates: `confgroup`.

Children

The following elements occur in confdates: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`confgroup`, `confnum`, `confsponsor`, `conftitle`, `contractnum`, `contractsponsor`.

Examples

For examples, see `confgroup`.

■ confgroup

Name

confgroup – A wrapper for document meta-information about a conference

Synopsis

Content Model

confgroup ::= ((`confdates`|`conftitle`|`confnum`|`address`|`confsponsor`)*)

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

If a document, for example an `Article`, is written in connection with a conference, the elements in this wrapper are used to hold information about the conference: titles, sponsors, addresses, dates, etc.

Processing Expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain confgroup: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in confgroup: `address`, `confdates`, `confnum`, `confsponsor`, `conftitle`.

See Also

`confdates`, `confnum`, `confsponsor`, `conftitle`, `contractnum`, `contractsponsor`.

Examples

```
<!DOCTYPE confgroup PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<confgroup>
<confdates>April, 1998</confdates>
<conftitle>The World Wide Web Conference</conftitle>
<confnum>7</confnum>
<address>Brisbane, Australia</address>
<confsponsor>World Wide Web Conference Committee (W3C3)</confsponsor>
</confgroup>
```

■ confnum

Name

confnum – An identifier, frequently numerical, associated with a conference for which a document was written

Synopsis

Mixed Content Model

confnum ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

See `ConfGroup`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain confnum: `confgroup`.

Children

The following elements occur in confnum: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`confdates`, `confgroup`, `confsponsor`, `conftitle`, `contractnum`, `contractsponsor`.

Examples

For examples, see `confgroup`.

■ confsponsor

Name

confsponsor – The sponsor of a conference for which a document was written

Synopsis

Mixed Content Model

confsponsor ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

See `ConfGroup`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain confsponsor: `confgroup`.

Children

The following elements occur in confsponsor: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`confdates`, `confgroup`, `confnum`, `conftitle`, `contractnum`, `contractsponsor`.

Examples

For examples, see `confgroup`.

■ conftitle

Name

conftitle – The title of a conference for which a document was written

Synopsis

Mixed Content Model

conftitle ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

See `ConfGroup`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain conftitle: `confgroup`.

Children

The following elements occur in conftitle: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`confdates`, `confgroup`, `confnum`, `confsponsor`, `contractnum`, `contractsponsor`.

Examples

For examples, see `confgroup`.

■ constant

Name

constant – A programming or system constant

Synopsis

Mixed Content Model

constant ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

class limit
None

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

A `Constant` identifies a value as immutable. It is most often used to identify system limitations or other defined constants.

Processing expectations

Formatted inline.

Parents

These elements contain constant: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in constant: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

class Class identifies constants that are system or application limits (for example, the maximum length of a filename).

See Also

command, computeroutput, literal, markup, option, optional, parameter, prompt, replaceable, sgmltag, userinput, varname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In ACL, <constant>main::PCS</constant> contains the path component
separator character.
</para>
```

In ACL, main::PCS contains the path component separator character.

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The maximum legal length for a path name is
<constant class='limit'>PATH_MAX</constant>, defined in
<filename class='headerfile'>limits.h</filename>.
</para>
```

The maximum legal length for a path name is PATH_MAX, defined in limits.h.

■ constraint

Name

constraint – A constraint in an EBNF production

Synopsis

Content Model

constraint ::= EMPTY

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

Description

This element is only available if you are using the EBNF Module.

A `Constraint` is a cross-reference to a description of a constraint that cannot be expressed in the grammar (generally logical rather than syntactic constraints).

Parents

These elements contain constraint: `production`.

Attributes

linkend FIXME:

Examples

For examples, see `productionset`.

constraintdef

Name

constraintdef – The definition of a constraint in an EBNF production

Synopsis

Content Model

constraintdef ::= (`title`, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`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`|`sidebar`|`anchor`|`bridgehead`|`remark`|`indexterm`|`beginpage`))

Attributes Common attributes

Parameter Entities

`%bookcomponent.content;` `%component.mix;` `%compound.class;`

`%divcomponent.mix;` `%ebnf.block.hook;` `%refcomponent.mix;`

Description

This element is only available if you are using the EBNF Module.

A `ConstraintDef` contains a description of a constraint that cannot be expressed in the grammar (generally logical rather than syntactic constraints).

Processing expectations

Formatted as a displayed block.

Parents

These elements contain constraintdef: `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `chapter`, `glossary`, `glossdiv`, `index`, `listitem`, `msgexplan`, `msgtext`, `partintro`, `preface`, `procedure`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `simplesect`, `step`.

Children

The following elements occur in constraintdef: `address`, `anchor`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `orderedlist`, `para`, `procedure`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `synopsis`, `table`, `title`, `variablelist`.

Examples

For examples, see `productionset`.

■ constructorsynopsis

Name

constructorsynopsis – A syntax summary for a constructor

Synopsis

Content Model

constructorsynopsis ::= (`modifier`*,`methodname`?, (`methodparam`+|`void`), `exceptionname`*)

Attributes Common attributes

Name Type Default

language CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%method.synop.class;` `%para.char.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%synop.class;` `%tbl.entry.mdl;`

Description

A `ConstructorSynopsis` contains the syntax summary of a constructor in an object-oriented programming language. Unlike a `MethodSynopsis`, which it closely resembles, it may not identify a return type and the `MethodName` is optional (in some languages, constructor names can be generated automatically).

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis.

Parents

These elements contain constructorsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, classsynopsis, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in constructorsynopsis: exceptionname, methodname, methodparam, modifier, void.

■ contractnum

Name

contractnum – The contract number of a document

Synopsis

Mixed Content Model

contractnum ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The ContractNum element that occurs in bibliographic metadata contains information about the contract number of a contract under which a document was written.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain contractnum: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in contractnum: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

confdates, confgroup, confnum, confsponsor, conftitle, contractsponsor.

Examples

For examples, see `contractsponsor`.

■ contractsponsor

Name

contractsponsor – The sponsor of a contract

Synopsis

Mixed Content Model

contractsponsor ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The `ContractSponsor` element that occurs in bibliographic metadata contains information about the sponser of a contract under which a document was written.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain contractsponsor: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in contractsponsor: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`confdates`, `confgroup`, `confnum`, `confsponsor`, `conftitle`, `contractnum`.

Examples

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<article>
<articleinfo>
  <title>Retrofitting Class A Widgets</title>
  <contractsponsor>Xyzzy Engineering Resources</contractsponsor>
  <contractnum>314-592-7</contractnum>
  <pubsnumber>XER-314-7A</pubsnumber>
  <corpauthor>Technical Documentation Consultants, Inc.</corpauthor>
  <collab><collabname>John Whorfin</collabname>
    <affiliation><orgname>Yoyodyne Propulsion Systems</orgname></affiliation>
```

```
  </collab>
</articleinfo>
<para>&hellip;</para>
</article>
```

■ contrib

Name

contrib – A summary of the contributions made to a document by a credited source

Synopsis

Mixed Content Model

contrib ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;` `%person.ident.mix;`

Description

The `Contrib` element contains a summary or description of the contributions made by an author, editor, or other credited source.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain contrib: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in contrib: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`author`, `authorblurb`, `authorgroup`, `collab`, `collabname`, `corpauthor`, `editor`, `othercredit`, `personblurb`, `personname`.

Examples

For examples, see `othercredit`.

■ copyright

Name

copyright – Copyright information about a document

Synopsis

Content Model

copyright ::= (year+,holder*)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The Copyright element holds information about the date(s) and holder(s) of a document copyright. If an extended block of text describing the copyright or other legal status is required, use LegalNotice.

The Copyright element is confined to meta-information. For copyright statements in running text, see Trademark.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

A displayed copyright notice usually includes the copyright symbol, ©, as generated text and is formatted with commas separating multiple years. Additional generated text, such as the legend "All rights reserved," may also be generated.

Parents

These elements contain copyright: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in copyright: holder, year.

See Also

legalnotice, productname, trademark.

Examples

```
<!DOCTYPE copyright PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<copyright>
  <year>1996</year>
  <year>1997</year>
  <holder>O'Reilly &amp; Associates, Inc.</holder>
</copyright>
```

For additional examples, see also bibliography, biblioset, bookinfo.

■ coref

Name

coref – A cross reference to a co

Synopsis

Content Model

coref ::= EMPTY

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

label CDATA *None*

Description

The `coref` plays a role for callouts that is analagous to the role of `footnoteref` for footnotes.

Use one `co` and one or more `coref` elements when you want to indicate that the same callout should appear in several places.

NOTE

A `coref` is not a cross-reference to a callout (use `xref` for that), rather, it is an indication that the callout appears semantically in more than one place.

Processing expectations

Formatted inline.

Parents

These elements contain coref: `literallayout`, `programlisting`, `screen`, `synopsis`.

Attributes

label Label specifies an identifying number or string that may be used in presentation.

linkend Linkend points to the `co` referenced.

See Also

`areaspec`, `calloutlist`, `co`, `graphicco`, `imageobjectco`, `mediaobjectco`, `programlistingco`, `screenco`.

■ corpauthor

Name

corpauthor – A corporate author, as opposed to an individual

Synopsis

Mixed Content Model

corpauthor ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %docinfo.char.class; %info.class;

%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix;

Description

In documents that have no specific authors, but are credited as authored by a corporation, the CorpAuthor tag can be used in place of the Author tag to indicate authorship. This element is used in bibliographic metadata.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain corpauthor: appendixinfo, application, articleinfo, attribution, authorgroup, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayou lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in corpauthor: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

author, authorblurb, authorgroup, collab, collabname, contrib, editor, othercredit, personblurb, personname.

Examples

For examples, see contractsponsor, set.

■ corpname

Name

corpname – The name of a corporation

Synopsis

Mixed Content Model

corpname ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The name of a corporation.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain corpname: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in corpname: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `jobtitle`, `orgdiv`, `orgname`, `publishername`, `shortaffil`.

Examples

For examples, see `bibliography`, `bibliomset`.

■ country

Name

country – The name of a country

Synopsis

Mixed Content Model

country ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

The name of a country, typically in an address.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain country: address.

Children

The following elements occur in country: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

address, city, email, fax, otheraddr, phone, pob, postcode, state, street.

Examples

For examples, see address, otheraddr.

■ database

Name

database – The name of a database, or part of a database

Synopsis

Mixed Content Model

database ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name	Type	Default
moreinfo	none refentry	"none"
class	field key1 key2 name record table	*None*

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The name of a database, or part of a database.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of `Database` will be constrained to `(#PCDATA | Replaceable | InlineGraphic)` in DocBook V4.0.

Parents

These elements contain database: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

Attributes

class Class allows the author to identify particular elements of a database.

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Database`.

See Also

`application`, `filename`, `hardware`, `medialabel`, `productname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <database>ProjectStatus</database> database has been updated.
Please note that <database class='field'>Year</database> has been
extended to four digits.
</para>
```

The ProjectStatus database has been updated. Please note that Year has been extended to four digits.

■ date

Name

date – The date of publication or revision of a document

Synopsis

Mixed Content Model

date ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

The `Date` element identifies a date.

Processing expectations

Formatted inline.

DocBook does not specify the format of the date.

Parents

These elements contain date: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `revision`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in date: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`edition`, `printhistory`, `pubdate`, `releaseinfo`, `revhistory`.

Examples

For examples, see `article`, `revhistory`.

■ dedication

Name

dedication – A wrapper for the dedication section of a book

Synopsis

Content Model

dedication ::= ((`title`,`subtitle`?,`titleabbrev`?)?, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`caution`|`important`|`note`|`tip`|`warning`|`literallayout`|`programlisting`|`programlistingco`|`screen`|`screenco`|`screenshot`|`formalpara`|`para`|`simpara`|`blockquote`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

A `Dedication` is a section at the very beginning of a book (before any other body matter) containing a tribute to something (frequently someone) in connection with the writing or publication of the `Book`.

Processing expectations

Formatted as a displayed block. Frequently appears on a page by itself at the beginning of a book.

Parents

These elements contain dedication: `book`.

Children

The following elements occur in dedication: `beginpage`, `blockquote`, `calloutlist`, `caution`, `formalpara`, `glosslist`, `important`, `indexterm`, `itemizedlist`, `literallayout`, `note`, `orderedlist`, `para`, `programlisting`, `programlistingco`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `subtitle`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `Dedication`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`appendix`, `article`, `book`, `chapter`, `colophon`, `part`, `partintro`, `preface`, `set`.

Examples

For examples, see `book`.

■ destructorsynopsis

Name

destructorsynopsis – A syntax summary for a destructor

Synopsis

Content Model

destructorsynopsis ::= (`modifier*`,`methodname`?, (`methodparam`+|`void`), `exceptionname*`)

Attributes Common attributes

Name Type Default

language CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%method.synop.class;` `%para.char.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%synop.class;` `%tbl.entry.mdl;`

Description

A `DestructorSynopsis` contains the syntax summary of a destructor in an object-oriented programming language. Unlike a `MethodSynopsis`, which it closely resembles, it may not identify a return type and the `MethodName` is optional (in some languages, destructors have an immutable name which may be generated automatically).

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis.

Parents

These elements contain destructorsynopsis: `answer`, `appendix`, `application`, `article`, `attribution`, `bibliodiv`, `bibliography`, `bibliomisc`, `blockquote`, `callout`, `caution`, `chapter`, `citation`, `citetitle`, `classsynopsis`, `constraintdef`, `emphasis`, `entry`, `example`, `figure`, `footnote`, `foreignphrase`, `glossary`, `glossdef`, `glossdiv`, `glosssee`, `glossseealso`, `glossterm`, `important`, `index`, `indexdiv`, `informalexample`, `informalfigure`, `itemizedlist`, `lineannotation`, `link`, `listitem`, `literallayout`, `lotentry`, `member`, `msgaud`, `msgexplan`, `msgtext`, `note`, `olink`, `orderedlist`, `para`, `partintro`, `phrase`, `preface`, `procedure`, `productname`, `programlisting`, `qandadiv`, `qandaset`, `question`, `quote`, `refentrytitle`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `remark`, `revdescription`, `screen`, `screeninfo`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `seg`, `setindex`, `sidebar`, `simpara`, `simplesect`, `step`, `synopsis`, `term`, `tip`, `tocback`, `tocentry`, `tocfront`, `ulink`, `variablelist`, `warning`.

Children

The following elements occur in destructorsynopsis: `exceptionname`, `methodname`, `methodparam`, `modifier`, `void`.

■ edition

Name

edition – The name or number of an edition of a document

Synopsis

Mixed Content Model

edition ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The Edition contains the name or number of the edition of the document.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain edition: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in edition: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

date, printhistory, pubdate, releaseinfo, revhistory.

Examples

For examples, see bookinfo.

■ editor

Name

editor – The name of the editor of a document

Synopsis

Content Model

editor ::= ((personname|(honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib)+), (personblurb|email|address)*)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The name of the editor of a document.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `Editor` in DocBook V4.0. A new wrapper element will be created to hold `Editor`, `AuthorBlurb`, and `Affiliation`.

Parents

These elements contain editor: `appendixinfo`, `articleinfo`, `authorgroup`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in editor: `address`, `affiliation`, `authorblurb`, `contrib`, `email`, `firstname`, `honorific`, `lineage`, `othername`, `personblurb`, `personname`, `surname`.

See Also

`author`, `authorblurb`, `authorgroup`, `collab`, `collabname`, `contrib`, `corpauthor`, `othercredit`, `personblurb`, `personname`.

Examples

For examples, see `authorgroup`, `bibliography`, `biblioset`.

■ email

Name

email – An email address

Synopsis

Mixed Content Model

email ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

Inline markup identifying an email address.

Processing expectations

Formatted inline. `Email` may generate surrounding punctuation, such as angle brackets. This is an interchange issue. See Appendix F.

In some processing environments, `Email` may automatically generate a hypertext link (a `mailto:` URL).

In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain email: action, address, application, attribution, author, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, editor, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, othercredit, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in email: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

address, city, country, fax, otheraddr, phone, pob, postcode, state, street.

■ emphasis

Name

emphasis – Emphasized text

Synopsis

Mixed Content Model

emphasis ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%docinfo.char.mix; %gen.char.class; %ndxterm.char.mix;

%para.char.mix; %refinline.char.mix; %tbl.entry.mdl;

%title.char.mix; %word.char.mix;

Description

Emphasis provides a method for indicating that certain text should be stressed in some way.

Processing expectations

Formatted inline. Emphasized text is traditionally presented in italics or boldface. A Role attribute of `bold` or `strong` is often used to generate boldface, if italics is the default presentation.

`Emphasis` is often used wherever its typographic presentation is desired, even when other markup might theoretically be more appropriate.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain emphasis: `abbrev`, `ackno`, `acronym`, `application`, `artpagenums`, `attribution`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `bibliosource`, `bridgehead`, `citation`, `citebiblioid`, `citetitle`, `city`, `collabname`, `confdates`, `confnum`, `confsponsor`, `conftitle`, `contractnum`, `contractsponsor`, `contrib`, `corpauthor`, `corpname`, `country`, `date`, `edition`, `email`, `emphasis`, `entry`, `fax`, `firstname`, `firstterm`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `holder`, `honorific`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `jobtitle`, `label`, `lineage`, `lineannotation`, `link`, `literallayout`, `lotentry`, `manvolnum`, `member`, `modespec`, `msgaud`, `olink`, `orgdiv`, `orgname`, `otheraddr`, `othername`, `pagenums`, `para`, `phone`, `phrase`, `pob`, `postcode`, `primary`, `primaryie`, `productname`, `productnumber`, `programlisting`, `pubdate`, `publishername`, `pubsnumber`, `quote`, `refentrytitle`, `refmiscinfo`, `refpurpose`, `releaseinfo`, `remark`, `revnumber`, `revremark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `seriesvolnums`, `shortaffil`, `simpara`, `state`, `street`, `subscript`, `subtitle`, `superscript`, `surname`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `volumenum`, `wordasword`, `year`.

Children

The following elements occur in emphasis: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

See Also

`abbrev`, `acronym`, `foreignphrase`, `phrase`, `quote`, `wordasword`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <emphasis>most</emphasis> important example of this
phenomenon occurs in A. Nonymous's book
<citetitle>Power Snacking</citetitle>.
</para>
```

The *most* important example of this phenomenon occurs in A. Nonymous's book *Power Snacking*. For additional examples, see also `index`, `programlistingco`, `type`.

■ entry

Name

entry – A cell in a table

Synopsis

Mixed Content Model

entry ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`|`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`caution`|`important`|`note`|`tip`|`warning`|`literallayout`|`programlisting`|`programlistingco`|`screen`|`screenco`|`screenshot`|`formalpara`|`para`|`simpara`|`graphic`|`mediaobject`)*

Attributes Common attributes

Name Type Default

nameend CDATA *None*

valign (bottom | middle | top) *None*

char CDATA *None*

colsep CDATA *None*

charoff CDATA *None*

morerows CDATA *None*

spanname CDATA *None*

namest CDATA *None*

colname CDATA *None*

rowsep CDATA *None*

rotate CDATA *None*

align (center | char | justify | left | right) *None*

Parameter Entities

`%tbl.row.mdl;`

Description

Entry is a cell in a table.

Each Entry may specify its starting column. Entries that do not explicitly specify a starting column begin implicitly in the column that is immediately adjacent to the preceding cell. Note that Entrys with the MoreRows attribute from preceding rows implicitly occupy cells in the succeeding Rows.

Rows are not required to be full. It is legal for some entries to be completely absent (at the beginning, middle, or end of a row). Pernicious Mixed Content

The content model of the Entry element exhibits a nasty peculiarity that we call "pernicious mixed content".[1]

Every other element in DocBook contains either block elements or inline elements (including #PCDATA) unambiguously. In these cases, the meaning of line breaks and spaces are well understood; they are insignificant between block elements and significant (to the SGML parser, anyway) where inline markup can occur.

Table entries are different; they can contain either block or inline elements, but not both at the same time. In other words, one Entry in a table might contain a paragraph or a list while another contains simply #PCDATA or another inline markup, but no single Entry can contain both.

Because the content model of an Entry allows both kinds of markup, each time the SGML parser encounters an Entry, it has to decide what variety of markup it contains. SGML parsers are forbidden to use more than a single token of lookahead to reach this decision. In practical terms, what this means is that a line feed or space after an Entry start tag causes the parser to decide that the cell contains inline markup. Subsequent discovery of a paragraph or another block element causes a parsing error.

All of these are legal:

```
<entry>3.1415927</entry>
<entry>General <emphasis>#PCDATA</emphasis></entry>
<entry><para>
A paragraph of text
</para></entry>
```

However, each of these is an error:

```
<entry>
Error, cannot have a line break before a block element
<para>
A paragraph of text.
</para></entry>

<entry><para>
A paragraph of text.
</para>               Error, cannot have a line break between block elements

<para>
A paragraph of text.
</para></entry>

<entry><para>
A paragraph of text.
</para>               Error, cannot have a line break after a block element

</entry>
```

When designing a DTD, it is wise to avoid pernicious mixed content. Unfortunately, the only way to correct the pernicious mixed content problem that already exists in DocBook is to require some sort of wrapper (a block element, or an inline like Phrase) around #PCDATA within table Entrys. This is annoying and inconvenient in a great many tables in which #PCDATA cells predominate and, in addition, differ from CALS.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* http://www.oasis-open.org/html/a502.htm.

[1] A term coined by Terry Allen.

The content of Entry is formatted to fit within the table cell that it occupies. Horizontal and vertical spanning may allow the content of an Entry to occupy several physical cells.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain entry: row.

Children

The following elements occur in entry: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, calloutlist, caution, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, formalpara, funcsynopsis, function, glosslist, glossterm, graphic, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, important, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, itemizedlist, keycap, keycode, keycombo, keysym, link, literal, literallayout, markup, medialabel, mediaobject, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, note, olink, ooclass, ooexception, oointerface, option, optional, orderedlist, othercredit, para, parameter, personname, phrase, productname, productnumber, programlisting, programlistingco, prompt, property, quote, remark, replaceable, returnvalue, revhistory, screen, screenco, screenshot, segmentedlist, sgmltag, simpara, simplelist, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, tip, token, trademark, type, ulink, userinput, variablelist, varname, warning, wordasword, xref.

Attributes

align Align specifies the horizontal alignment of text (and other elements) within the Entry. If no alignment is specified, it is inherited from the ColSpec for the current column, or the SpanSpec if this entry occurs in a span. If Char is specified, see also Char and CharOff.

char Char specifies the alignment character when the Align attribute is set to Char.

charoff CharOff specifies the percentage of the column's total width that should appear to the left of the first occurance of the character identified in Char when the Align attribute is set to Char. This attribute is inherited from the relevant ColSpec or SpanSpec.

colname ColName identifies the column in which this entry should appear; it must have been previously defined in a ColSpec. Entrys cannot be given out of order, the column referenced must be to the right of the last Entry or EntryTbl placed in the current row. It is an error to specify both a ColName and a SpanName.

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of this Entry. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute of the enclosing Table or InformalTable and the ColSep of an entry in the last column in the table is ignored. If unspecified, this attribute is inherited from the the corresponding ColSpec or SpanSpec and enclosing elements.

morerows MoreRows indicates how many more rows, in addition to the current row, this `Entry` is to occupy. It creates a vertical span. The default of `0` indicates that the `Entry` occupies only a single row.

nameend NameEnd is the name (defined in a `ColSpec`) of the rightmost column of a span. On `Entry`, specifying both NameSt and NameEnd defines a horizontal span for the current `Entry`. (See also SpanName.)

namest NameSt ("name start") is the name (defined in a `ColSpec`) of the leftmost column of a span. On `Entry`, specifying both NameSt and NameEnd defines a horizontal span for the current `Entry`. (See also SpanName.)

rotate If Rotate has the value 1 (true), the `Entry` is to be rotated 90 degrees counterclockwise in the table cell. A value of 0 (false) indicates that no rotation is to occur. If the stylesheet also specifies rotation, the value of Rotate is ignored; they are not additive. Only the values 0 and 1 are legal.

rowsep If RowSep has the value 1 (true), then a rule will be drawn below the `Entry`. A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements.

spanname SpanName is the name (defined in a `SpanSpec`) of a span. This cell will be rendered with the specified horizontal span.

valign VAlign specifies the vertical alignment of text (and other elements) within the `Entry`. If no alignment is specified, it is inherited from enclosing elements.

See Also

`colspec`, `entrytbl`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

For examples, see `entrytbl`, `footnoteref`, `informaltable`, `table`.

■ entrytbl

Name

entrytbl – A subtable appearing in place of an `Entry` in a table

Synopsis

Content Model

entrytbl ::= (`colspec`*,`spanspec`*,`thead`?,`tbody`)

Attributes Common attributes

Name Type Default

nameend CDATA *None*

char CDATA *None*

charoff CDATA *None*

spanname CDATA *None*

namest CDATA *None*

tgroupstyle CDATA *None*

cols CDATA *Required*

colsep CDATA *None*

rowsep CDATA *None*

colname CDATA *None*

align (center | char | justify | left | right) *None*

Parameter Entities

`%tbl.row.mdl;`

Description

The `EntryTbl` element allows for a single level of nesting within tables. This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`.

An entry table may occur in a row instead of an `Entry`. `EntryTbl`s have most of the elements of a table but may not include themselves, thus limiting nesting to a single level.

Processing expectations

The content of `EntryTbl` is formatted, *as a table*, to fit within the table cell that it occupies. Horizontal and vertical spanning may allow an `EntryTbl` to occupy several physical cells in the table that contains it.

If multiple `EntryTbl`s occur in a single row, formatters that support `EntryTbl` are not required to ensure that subrows within the various tables are vertically aligned.

Many formatters are incapable of supporting `EntryTbl`s. This is an interchange issue. See Appendix F.

Parents

These elements contain entrytbl: `row`.

Children

The following elements occur in entrytbl: `colspec`, `spanspec`, `tbody`, `thead`.

Attributes

align Align specifies the horizontal alignment of the `EntryTbl` in the cell in which it occurs. If no alignment is specified, it is inherited from the `ColSpec` for the current column, or the `SpanSpec` if this entry occurs in a span.

char Char specifies the alignment character when the Align attribute is set to `Char`.

charoff CharOff specifies the percentage of the column's total width that should appear to the left of the first occurance of the character identified in Char when the Align attribute is set to `Char`. This attribute is inherited from the relevant `ColSpec` or `SpanSpec`.

colname ColName identifies the column in which this entry table should appear; it must have been previously defined in a `ColSpec`. `EntryTbl`s cannot be given out of order, the column referenced must be to the right of the last `Entry` or `EntryTbl` placed in the current row. It is an error to specify both a ColName and a SpanName.

cols Cols specifies the number of columns in the `EntryTbl`.

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of this `EntryTbl`. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the ColSep of an entry in the last column in the table is ignored. If unspecified, this attribute is inherited from the the corresponding `ColSpec` or `SpanSpec` and enclosing elements.

nameend NameEnd is the name (defined in a `ColSpec`) of the rightmost column of a span. On `EntryTbl`, specifying both NameSt and NameEnd defines a horizontal span for the current `EntryTbl`. (See also SpanName.)

namest NameSt ("name start") is the name (defined in a `ColSpec`) of the leftmost column of a span. On `EntryTbl`, specifying both NameSt and NameEnd defines a horizontal span for the current `EntryTbl`. (See also SpanName.)

rowsep If RowSep has the value 1 (true), then a rule will be drawn below the `EntryTbl`. A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements.

spanname SpanName is the name (defined in a `SpanSpec`) of a span. This cell will be rendered with the specified horizontal span.

tgroupstyle TGroupstyle holds the name of a table group style defined in a stylesheet that will be used to process this document.

See Also

`colspec`, `entry`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

```
<!DOCTYPE informaltable PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<!-- entrytbl not supported, fake it in descrip.1 -->
<informaltable frame='all'>
<tgroup cols='3'>
<tbody>
<row>
  <entry>a1</entry>
  <entry>b1</entry>
  <entry>c1</entry>
</row>
<row>
  <entry>a2</entry>
  <entrytbl cols='3'>
    <tbody>
      <row>
        <entry>b2a1</entry>
        <entry>b2b1</entry>
        <entry>b2c1</entry>
      </row>
      <row>
```

```
        <entry>b2a2</entry>
        <entry>b2b2</entry>
        <entry>b2c2</entry>
      </row>
      <row>
        <entry>b2a3</entry>
        <entry>b2b3</entry>
        <entry>b2c3</entry>
      </row>
    </tbody>
  </entrytbl>
  <entry>c2</entry>
</row>
<row>
  <entry>a3</entry>
  <entry>b3</entry>
  <entry>c3</entry>
</row>
</tbody>
</tgroup>
</informaltable>
```

The preceding table would look something like this when formatted:

a1	b1			c1
a2	b2a1	b2b1	b2c1	c2
b2a2	b2b2	b2c2		
b2a3	b2b3	b2c3		
a3	b3			c3

■ envar

Name

envar – A software environment variable

Synopsis

Mixed Content Model

envar ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`| `beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

`EnVar` is an environment variable used most often for the UNIX, DOS, or Windows environments.

Processing expectations

Formatted inline.

Parents

These elements contain envar: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in envar: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

computeroutput, filename, prompt, systemitem, userinput.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In order to translate public identifiers into local system identifiers,
<application>Jade</application> and <application>SP</application> read the catalog fi
pointed to by <envar>SGML_CATALOG_FILES</envar>.
</para>
```

In order to translate public identifiers into local system identifiers, Jade and SP read the catalog files pointed to by SGML_CATALOG_FILES.

■ epigraph

Name

epigraph – A short inscription at the beginning of a document or component

Synopsis

Content Model

epigraph ::= (attribution?, ((formalpara|para|simpara)| literallayout)+)

Attributes Common attributes

Parameter Entities

%bookcomponent.content; %component.mix; %descobj.class;

%divcomponent.mix; %listpreamble.mix; %refcomponent.mix;

Description

An Epigraph is a short inscription, often a quotation or poem, set at the beginning of a document or component. Epigraphs are usually related somehow to the content that follows them and may help set the tone for the component.

Processing expectations

Formatted as a displayed block.

Future Changes

`Epigraph` will not be allowed in `BlockQuote` in DocBook V4.0.

Parents

These elements contain epigraph: `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `chapter`, `glossary`, `glossdiv`, `index`, `itemizedlist`, `listitem`, `msgexplan`, `msgtext`, `orderedlist`, `partintro`, `preface`, `procedure`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `simplesect`, `step`, `variablelist`.

Children

The following elements occur in epigraph: `attribution`, `formalpara`, `literallayout`, `para`, `simpara`.

See Also

`abstract`, `attribution`, `blockquote`, `highlights`, `sidebar`.

Examples

For examples, see `chapter`.

■ equation

Name

equation – A displayed mathematical equation

Synopsis

Content Model

equation ::= (`blockinfo`?, (`title`,`titleabbrev`?)?, (`informalequation`| (`alt`?, (`graphic`+| `mediaobject`+))))

Attributes Common attributes

Name Type Default

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%formal.class;` `%glossdef.mix;`

`%para.mix;` `%qandaset.mix;` `%refcomponent.mix;`

`%revdescription.mix;` `%sidebar.mix;`

Description

An `Equation` is a formal mathematical equation (with an optional rather than required title).

If the MathML Module is used, `Equation` can also contain the `mml:math` element.

Processing expectations

Formatted as a displayed block. For an inline equation, use `InlineEquation`.

Processing systems that number equations or build a table of equations at the beginning of a document may have difficulty correctly formatting documents that contain both `Equation`s with `Title`s and `Equation`s without `Title`s. You are advised to use `InformalEquation` for equations without titles.

Future Changes

In some future version of DocBook, probably V5.0, even though the change has not yet been announced, the `Title` on `Equation` will be required. For equations without titles, use `InformalEquation`.

Parents

These elements contain equation: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caution`, `chapter`, `constraintdef`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `tip`, `warning`.

Children

The following elements occur in equation: `alt`, `blockinfo`, `graphic`, `informalequation`, `mediaobject`, `title`, `titleabbrev`.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

See Also

`example`, `figure`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `inlineequation`, `subscript`, `superscript`, `table`.

Examples

```
<!DOCTYPE equation PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<equation><title>Fermat's Last Theorum</title>
  <alt>x^n + y^n &ne; z^n &forall; n &ne; 2</alt>
  <graphic fileref="figures/fermat.png"/>
</equation>
```

$$x^n + y^n \neq z^n \forall n \neq 2$$

Equation 6.98.1: Fermat's Last Theorum

■ errorcode

Name

errorcode – An error code

Synopsis

Mixed Content Model

errorcode ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

An error code. Error codes are often numeric, but in some environments they may be symbolic constants.

DocBook provides four elements for identifying the parts of an error message: `ErrorCode`, for the alphanumeric error code (e.g., “-2”); `ErrorName`, for the symbolic name of the error (e.g., “ENOENT”); `ErrorText`, for the text of the error message (e.g., “file not found”); and `ErrorType`, for the error type (e.g., “recoverable”).

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain errorcode: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in errorcode: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `ErrorCode`.

See Also

`errorname`, `errortext`, `errortype`, `msgset`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
On most DOS-derived systems, functions signal a <errorname>File
Not Found</errorname> error by returning
<errorcode>2</errorcode>.  This is usually a
<errortype>recoverable</errortype> (non-fatal) error.
</para>
```

On most DOS-derived systems, functions signal a File Not Found error by returning 2. This is usually a recoverable (non-fatal) error.

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
On most UNIX systems, functions signal a <errorname>File
Not Found</errorname> error by returning
<errorcode>ENOENT</errorcode>, defined in
<filename>errno.h</filename>.  This is usually a
<errortype>recoverable</errortype> (non-fatal) error.
</para>
```

On most UNIX systems, functions signal a File Not Found error by returning ENOENT, defined in `errno.h`. This is usually a recoverable (non-fatal) error.

■ errorname

Name

errorname – An error name

Synopsis

Mixed Content Model

errorname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

`ErrorName` holds the symbolic name of an error.

DocBook provides four elements for identifying the parts of an error message: `ErrorCode`, for the alphanumeric error code (e.g., “-2”); `ErrorName`, for the symbolic name of the error (e.g., “ENOENT”); `ErrorText`, for the text of the error message (e.g., “file not found”); and `ErrorType`, for the error type (e.g., “recoverable”).

Prior to DocBook V4.2, the `ErrorName` element was the recommended element for error *messages*. However, this left no element for symoblic names, so the `ErrorText` element was added and the semantics of the error elements adjusted slightly.

Processing expectations

Formatted inline.

Parents

These elements contain errorname: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in errorname: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

errorcode, errortext, errortype, msgset.

Examples

For examples, see errorcode, returnvalue.

■ errortext

Name

errortext – An error message.

Synopsis

Mixed Content Model

errortext ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

ErrorText holds the text of an error message.

DocBook provides four elements for identifying the parts of an error message: ErrorCode, for the alphanumeric error code (e.g., “-2”); ErrorName, for the symbolic name of the error (e.g., “ENOENT”); ErrorText, for the text of the error message (e.g., “file not found”); and ErrorType, for the error type (e.g., “recoverable”).

Prior to DocBook V4.2, the ErrorName element was the recommended element for error *messages*. However, this left no element for symoblic names, so the ErrorText element was added and the semantics of the error elements adjusted slightly.

Processing expectations

Formatted inline.

Parents

These elements contain errortext: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in errortext: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

errorcode, errorname, errortype, msgset.

■ errortype

Name

errortype – The classification of an error message

Synopsis

Mixed Content Model

errortype ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The ErrorType element identifies a class of error. The exact classifications are naturally going to vary by system, but “recoverable” and “fatal” are two possibilities.

DocBook provides four elements for identifying the parts of an error message: ErrorCode, for the alphanumeric error code (e.g., “-2”); ErrorName, for the symbolic name of the error (e.g., “ENOENT”); ErrorText, for the text of the error message (e.g., “file not found”); and ErrorType, for the error type (e.g., “recoverable”).

Processing expectations

Formatted inline.

Parents

These elements contain errortype: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle,

refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in errortype: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

errorcode, errorname, errortext, msgset.

Examples

For examples, see errorcode.

■ example

Name

example – A formal example, with a title

Synopsis

Content Model

example ::= (blockinfo?, (title,titleabbrev?), (calloutlist|glosslist|itemizedlist| orderedlist|segmentedlist| simplelist|variablelist|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| indexterm|beginpage)+)

Attributes Common attributes

Name Type Default

width CDATA *None*

label CDATA *None*

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%divcomponent.mix; %formal.class; %glossdef.mix;

%para.mix; %qandaset.mix; %refcomponent.mix;

%revdescription.mix; %sidebar.mix;

Description

Example is a formal example with a title. Examples often contain ProgramListings or other large, block elements. Frequently they are given IDs and referenced from the text with XRef or Link.

Processing expectations

Formatted as a displayed block. DocBook does not specify the location of the example within the final displayed flow of text; it may float or remain where it is located.

A list of examples may be generated at the beginning of a document.

Parents

These elements contain example: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, glossary, glossdef, glossdiv, important, index, listitem, msgexplan, msgtext, note, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, warning.

Children

The following elements occur in example: address, beginpage, blockinfo, blockquote, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, fieldsynopsis, formalpara, funcsynopsis, glosslist, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, programlisting, programlistingco, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, title, titleabbrev, variablelist.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

width Width specifies the width (in characters) of the longest line in this Example (formatters may use this value to determine scaling or rotation).

See Also

equation, figure, informalequation, informalexample, informalfigure, informaltable, table.

Examples

```
<!DOCTYPE example PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<example><title>A DSSSL Function</title>
<programlisting>
(define (node-list-filter-by-gi nodelist gilist)
  ;; Returns the node-list that contains every element of the original
  ;; nodelist whose gi is in gilist
  (let loop ((result (empty-node-list)) (nl nodelist))
    (if (node-list-empty? nl)
   result
   (if (member (gi (node-list-first nl)) gilist)
       (loop (node-list result (node-list-first nl))
        (node-list-rest nl))
       (loop result (node-list-rest nl))))))
</programlisting>
</example>
```

For additional examples, see also remark.

Example 6.103.1: A DSSSL Function

```
(define (node-list-filter-by-gi nodelist gilist)
  ;; Returns the node-list that contains every element of the original
  ;; nodelist whose gi is in gilist
  (let loop ((result (empty-node-list)) (nl nodelist))
    (if (node-list-empty? nl)
   result
   (if (member (gi (node-list-first nl)) gilist)
       (loop (node-list result (node-list-first nl))
        (node-list-rest nl))
       (loop result (node-list-rest nl))))))
```

■ exceptionname

Name

exceptionname – The name of an exception

Synopsis

Mixed Content Model

exceptionname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

The `ExceptionName` element is used to identify the name of an interface. This is likely to occur only in documentation about object-oriented programming systems, languages, and architectures.

Processing expectations

Formatted inline.

Parents

These elements contain exceptionname: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `constructorsynopsis`, `database`, `destructorsynopsis`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `methodsynopsis`, `msgaud`, `olink`, `ooexception`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in exceptionname: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

■ fax

Name

fax – A fax number

Synopsis

Mixed Content Model

fax ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`Fax` is a fax number in an address.

Processing expectations

Formatted inline. Sometimes suppressed. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain fax: `address`.

Children

The following elements occur in fax: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `city`, `country`, `email`, `otheraddr`, `phone`, `pob`, `postcode`, `state`, `street`.

Examples

For examples, see `address`.

■ fieldsynopsis

Name

fieldsynopsis – The name of a field in a class definition

Synopsis

Content Model

fieldsynopsis ::= (`modifier`*,`type`?,`varname`,`initializer`?)

Attributes Common attributes

Name Type Default

language CDATA *None*

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %figure.mix;
%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;
%listpreamble.mix; %para.char.mix; %qandaset.mix;
%refcomponent.mix; %revdescription.mix; %sidebar.mix;
%synop.class; %tbl.entry.mdl;

Description

A FieldSynopsis contains the syntax summary of a field (generally speaking, fields in the object-oriented programming language sense).

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis.

Parents

These elements contain fieldsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, classsynopsis, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in fieldsynopsis: initializer, modifier, type, varname.

■ figure

Name

figure – A formal figure, generally an illustration, with a title

Synopsis

Content Model

figure ::= (blockinfo?, (title,titleabbrev?), (literallayout|programlisting|programlistingco| screen|screenco| screenshot|synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis|constructorsynopsis|destructorsynopsis| methodsynopsis|address| blockquote|graphic|graphicco|mediaobject| mediaobjectco|informalequation| informalexample|informalfigure| informaltable|indexterm|beginpage|link|olink| ulink)+)

Attributes Common attributes

Name Type Default

float CDATA "0"

pgwide CDATA *None*

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%formal.class;` `%glossdef.mix;`
`%para.mix;` `%qandaset.mix;` `%refcomponent.mix;`
`%revdescription.mix;` `%sidebar.mix;`

Description

`Figure` is a formal example with a title. Figures often contain `Graphic`s, or other large, display elements. Frequently they are given IDs and referenced from the text with `XRef` or `Link`.

Processing expectations

Formatted as a displayed block.

`Figure`s may contain multiple display elements. DocBook does not specify how these elements are to be presented with respect to one another.

DocBook does not specify the location of the figure within the final displayed flow of text; it may float or remain where it is located.

A list of figures may be generated at the beginning of a document.

Parents

These elements contain figure: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caution`, `chapter`, `constraintdef`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `tip`, `warning`.

Children

The following elements occur in figure: `address`, `beginpage`, `blockinfo`, `blockquote`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `fieldsynopsis`, `funcsynopsis`, `graphic`, `graphicco`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `link`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `olink`, `programlisting`, `programlistingco`, `screen`, `screenco`, `screenshot`, `synopsis`, `title`, `titleabbrev`, `ulink`.

Attributes

float If Float has the value 1 (true), then the processing system is free to move the figure to a convenient location. (Where convenient location may be described in the style sheet or may be application dependent.) A value of 0 (false) indicates that the figure should be placed precisely where it occurs in the flow.

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

pgwide If Pgwide has the value 0 (false), then the `Figure` is rendered in the current text flow (with flow column width). A value of 1 (true) specifies that the figure should be rendered across the full text page.

See Also

`equation`, `example`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `table`.

Examples

```
<!DOCTYPE figure PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<figure><title>The Pythagorean Theorum Illustrated</title>
<graphic fileref="figures/pythag.png"/>
</figure>
```

For additional examples, see also `graphic`.

■ filename

Name

filename – The name of a file

Synopsis

Mixed Content Model

filename ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

path CDATA *None*

moreinfo
- none
- refentry

"none"

class
- devicefile
- directory
- extension
- headerfile
- libraryfile
- partition
- symlink

None

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

A `Filename` is the name of a file on a local or network disk. It may be a simple name or may include a path or other elements specific to the operating system.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of `Filename` will be constrained to `(#PCDATA | Replaceable | InlineGraphic)` in DocBook V4.0.

Future Changes

Filename extensions can be explicitly identified with the `extension` class value.

Parents

These elements contain filename: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

Attributes

class Class allows the author to identify particular kinds of files.

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Filename`.

path Path contains the *search* path, possibly in a system or application specific form, on which the file can be found.

See Also

`application`, `computeroutput`, `database`, `envar`, `hardware`, `medialabel`, `productname`, `prompt`, `systemitem`, `userinput`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The symbolic constants for error numbers are defined in
<filename class='headerfile'>errno.h</filename> in
<filename class='directory'>/usr/include/sys</filename>.
</para>
```

The symbolic constants for error numbers are defined in `errno.h` in `/usr/include/sys`. For additional examples, see also `constant`, `errorcode`, `keycap`, `keycode`, `keysym`, `literal`, `variablelist`.

■ firstname

Name

firstname – The first name of a person

Synopsis

Mixed Content Model

firstname ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;` `%person.ident.mix;`

Description

The Western-style first name of an author, editor, or other individual.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain firstname: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `personname`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in firstname: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `honorific`, `lineage`, `othername`, `surname`.

Examples

For examples, see `ackno`, `article`, `author`, `authorgroup`, `bibliography`, `bibliomset`, `biblioset`, `book`, `bookinfo`, `othercredit`, `personname`.

■ firstterm

Name

firstterm – The first occurrence of a term

Synopsis

Mixed Content Model

firstterm ::= (#PCDATA|acronym|emphasis|trademark|link|olink|ulink|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

linkend IDREF *None*

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

This element marks the first occurrence of a word or term in a given context.

Processing expectations

Formatted inline. FirstTerms are often given special typographic treatment, such as italics.

Parents

These elements contain firstterm: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in firstterm: acronym, anchor, beginpage, emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, subscript, superscript, trademark, ulink.

Attributes

linkend Linkend points to a related element, perhaps the Glossary definition of the term.

See Also

glossterm.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In an <firstterm>Object Oriented</firstterm> programming language,
data and procedures (called <glossterm>methods</glossterm>) are
bound together.
</para>
```

In an *Object Oriented* programming language, data and procedures (called methods) are bound together.

■ footnote

Name

footnote – A footnote

Synopsis

Content Model

footnote ::= ((calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist| simplelist|variablelist|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)+)

Attributes Common attributes

Name Type Default

label CDATA *None*

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

This element is a wrapper around the contents of a footnote.

Additional references to the same footnote may be generated with FootnoteRef.

Processing expectations

The Footnote element usually generates a mark (a superscript symbol or number) at the place in the flow of the document in which it occurs. The body of the footnote is then presented elsewhere, typically at the bottom of the page.

Alternative presentations are also possible. In print environments that do not support footnotes at the bottom of the page, they may be presented as end notes occurring at the end of the component that contains the Footnote. Online systems may choose to present them inline or as “pop-ups,” or links, or any combination thereof.

Parents

These elements contain footnote: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in footnote: `address`, `blockquote`, `calloutlist`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `fieldsynopsis`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `informalequation`, `informalexample`, `informalfigur` `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `orderedlist`, `para`, `programlisting`, `programlistingco`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `synopsis`, `variablelist`.

Attributes

label Label identifies the desired footnote mark.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
An annual percentage rate (<abbrev>APR</abbrev>) of 13.9%<footnote>
<para>
The prime rate, as published in the <citetitle>Wall Street
Journal</citetitle> on the first business day of the month,
plus 7.0%.
</para>
</footnote>
will be charged on all balances carried forward.
</para>
```

An annual percentage rate (APR) of 13.9%[2] will be charged on all balances carried forward. For additional examples, see also `footnoteref`.

■ footnoteref

Name

footnoteref – A cross reference to a footnote (a footnote mark)

Synopsis

Content Model

footnoteref ::= EMPTY

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

label CDATA *None*

Parameter Entities

`%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;`
`%tbl.entry.mdl;` `%title.char.mix;` `%xref.char.class;`

Description

This element forms an IDREF link to a `Footnote`. It generates the same mark or link as the `Footnote` to which it points.

In technical documentation, `FootnoteRef` occurs most frequently in tables.

[2] The prime rate, as published in the *Wall Street Journal* on the first business day of the month, plus 7.0%.

Processing expectations

The FootnoteRef element usually generates the same mark as the Footnote to which it points, although its mark can be influenced by the Label attribute.

Neither SGML nor XML provide a mechnism (at this time) for asserting that some IDREF links must point to specific classes of elements. If a FootnoteRef points to something other than a Footnote, the parser will not report it as an error. It is an error nonetheless. Processing is undefined.

Parents

These elements contain footnoteref: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Attributes

label Label identifies the desired footnote mark.

linkend Linkend points to the Footnote referenced.

Examples

```
<!DOCTYPE informaltable PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<informaltable>
<tgroup cols='2'>
<tbody>
<row>
<entry>foo<footnote id='fnrex1a'><para>A meaningless
word</para></footnote></entry>
<entry>3<footnote id='fnrex1b'><para>A meaningless
number</para></footnote></entry>
</row>
<row>
<entry>bar<footnoteref linkend='fnrex1a'/></entry>
<entry>5<footnoteref linkend='fnrex1b'/></entry>
</row>
</tbody>
</tgroup>
</informaltable>
```

foo[3]	3[4]
bar[3]	5[4]

[3] A meaningless word
[4] A meaningless number

■ foreignphrase

Name

foreignphrase – A word or phrase in a language other than the primary language of the document

Synopsis

Mixed Content Model

foreignphrase ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

The ForeignPhrase element can be used to markup the text of a foreign word or phrase. “Foreign” in this context means that it is a language other than the primary language of the document and is not intended to be pejorative in any way.

Processing expectations

ForeignPhrases are often given special typographic treatment, such as italics.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain foreignphrase: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in foreignphrase: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database,

destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

abbrev, acronym, emphasis, phrase, quote, wordasword.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Like so many others, it became a <foreignphrase>de facto</foreignphrase>
standard.
</para>
```

Like so many others, it became a *de facto* standard.

■ formalpara

Name

formalpara – A paragraph with a title

Synopsis

Content Model

formalpara ::= (title, (indexterm)*, para)

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %footnote.mix;
%glossdef.mix; %highlights.mix; %indexdivcomponent.mix;
%legalnotice.mix; %listpreamble.mix; %para.class;
%qandaset.mix; %refcomponent.mix; %revdescription.mix;
%sidebar.mix; %tabentry.mix; %tbl.entry.mdl;
%textobject.mix;

Description

Formal paragraphs have a title.

Processing expectations

Formatted as a displayed block. The Title of a FormalPara is often rendered as a run-in head.

Parents

These elements contain formalpara: abstract, answer, appendix, article, authorblurb, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, epigraph, example, footnote, glossary, glossdef, glossdiv, highlights, important, index, indexdiv, informalexample, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, partintro, personblurb, preface, printhistory, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in formalpara: indexterm, para, title.

See Also

para, simpara.

Examples

```
<!DOCTYPE formalpara PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<formalpara><title>This Paragraph Has a Title</title>
<para>
This is a test.  This is only a test.  Had this been a real
example, it would have made more sense.  Or less.
</para>
</formalpara>
```

This Paragraph Has a Title This is a test. This is only a test. Had this been a real example, it would have made more sense. Or less.

For additional examples, see also para.

■ funcdef

Name

funcdef – A function (subroutine) name and its return type

Synopsis

Mixed Content Model

funcdef ::= (#PCDATA|type|replaceable|function)*

Attributes Common attributes

Description

A FuncDef contains the name of a programming language function, and its return type.

Within the FuncDef, the function name is identified with Function, and the rest of the content is assumed to be the return type.

In the following definition, max is the name of the function and int is the return type:

```
<funcdef>int <function>max</function></funcdef>
```

Processing expectations

Formatted inline. For a complete description of the processing expecations, see `FuncSynopsis`.

Parents

These elements contain funcdef: `funcprototype`.

Children

The following elements occur in funcdef: `function`, `replaceable`, `type`.

See Also

`funcparams`, `funcprototype`, `funcsynopsisinfo`, `function`, `paramdef`, `parameter`, `returnvalue`, `varargs`, `void`.

Examples

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
<funcdef>int <function>rand</function></funcdef>
  <void/>
</funcprototype>
</funcsynopsis>
```

int **rand** (); For additional examples, see also `funcparams`, `funcsynopsis`, `paramdef`, `refentry`, `varargs`.

■ funcparams

Name

funcparams – Parameters for a function referenced through a function pointer in a synopsis

Synopsis

Mixed Content Model

funcparams ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

In some programming languages (like C), it is possible for a function to have a pointer to another function as one of its parameters. In the syntax summary for such a function, the `FuncParams` element provides a wrapper for the function pointer.

For example, the following prototype describes the function `sort`, which takes two parameters. The first parameter, *`arr`*, is an array of integers. The second parameter is a pointer to a function, `comp` that returns an int. The `comp` function takes two parameters, both of type int *:

```
<funcprototype>
  <funcdef>void <function>sort</function></funcdef>
    <paramdef>int *<parameter>arr</parameter>[]</paramdef>
    <paramdef>int <parameter>(* comp)</parameter>
      <funcparams>int *, int *</funcparams></paramdef>
</funcprototype>
```

Processing expectations

Formatted inline. For a complete description of the processing expecations, see `FuncSynopsis`.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain funcparams: `methodparam`, `paramdef`.

Children

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

See Also

`funcdef`, `funcprototype`, `funcsynopsisinfo`, `function`, `paramdef`, `parameter`, `returnvalue`, `varargs`, `void`.

Examples

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
  <funcdef>void <function>qsort</function></funcdef>
    <paramdef>void *<parameter>dataptr</parameter>[]</paramdef>
    <paramdef>int <parameter>left</parameter></paramdef>
    <paramdef>int <parameter>right</parameter></paramdef>
    <paramdef>int <parameter>(* comp)</parameter>
      <funcparams>void *, void *</funcparams></paramdef>
</funcprototype>
</funcsynopsis>
```

void **qsort** (*dataptr*, *left*, *right*, *(* comp)*);
void **dataptr*[];
int *left*;
int *right*;
int *(* comp)* (void *, void *); For additional examples, see also `funcsynopsis`.

■ funcprototype

Name

funcprototype – The prototype of a function

Synopsis

Content Model

funcprototype ::= (`funcdef`, (`void`|`varargs`|`paramdef`+))

Attributes Common attributes

Description

A wrapper for a function prototype in a `FuncSynopsis`.

Processing expectations

See `FuncSynopsis`.

Parents

These elements contain funcprototype: `funcsynopsis`.

Children

The following elements occur in funcprototype: `funcdef`, `paramdef`, `varargs`, `void`.

See Also

`funcdef`, `funcparams`, `funcsynopsisinfo`, `function`, `paramdef`, `parameter`, `returnvalue`, `varargs`, `void`.

Examples

For examples, see `funcdef`, `funcparams`, `funcsynopsis`, `paramdef`, `refentry`, `varargs`.

■ funcsynopsis

Name

funcsynopsis – The syntax summary for a function definition

Synopsis

Content Model

funcsynopsis ::= ((`funcsynopsisinfo`|`funcprototype`)+)

Attributes Common attributes

Name Type Default

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%para.char.mix;` `%qandaset.mix;`
`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`
`%synop.class;` `%tbl.entry.mdl;`

Description

A FuncSynopsis contains the syntax summary of a function prototype or a set of function prototypes. The content model of this element was designed specifically to capture the semantics of most C-language function prototypes (for use in UNIX reference pages).

This is one of the few places where DocBook attempts to model as well as describe. Using FuncSynopsis for languages that are unrelated to C may prove difficult.

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis. The exception to this rule is that the spacing and other punctuation inside a parameter that is a pointer to a function must be provided in the source markup.

With sufficient author cooperation, it should be possible to markup a function synopsis with enough clarity so that a processing system can generate either K&R-style or ANSI-style renderings.

A Note on the Use of VarArgs The content model of FuncPrototype is such that you cannot use VarArgs in a function prototype in which the first few parameters to the function are given explicitly before the variable arguments (generally rendered as an ellipsis).

In other words, the following synopsis cannot be rendered with VarArgs:

int printf(char *format, ...)

Instead, you can enclose the ellipsis in a final Parameter, like this:

```
<funcsynopsis>
<funcprototype><funcdef>int <function>printf</function></funcdef>
<paramdef>
  <parameter>char *format</parameter>
  <parameter>...</parameter>
</paramdef>
</funcprototype>
</funcsynopsis>
```

Future Changes

In DocBook V4.0, the content model fragment beginning with FuncDef will be removed from the content model of FuncSynopsis. Instead FuncSynopsis will become a mixture of FuncSynopsisInfos and FuncPrototypes.

Future Changes

Future versions of DocBook may provide additional environments for describing the syntax summaries of functions in other programming languages.

Parents

These elements contain funcsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in funcsynopsis: `funcprototype`, `funcsynopsisinfo`.

Attributes

label Label specifies an identifying number or string that may be used in presentation.

See Also

`cmdsynopsis`, `synopsis`.

Examples

The function `max` returns the larger of two integers:

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
  <funcdef>int <function>max</function></funcdef>
  <paramdef>int <parameter>int1</parameter></paramdef>
  <paramdef>int <parameter>int2</parameter></paramdef>
</funcprototype>
</funcsynopsis>
```

int **max** (*int1*, *int2*);
int *int1*;
int *int2*; One can imagine a more flexible `max` function that takes any number of integer arguments and returns the largest integer in the list:

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcsynopsisinfo>
#include &lt;varargs.h&gt;
</funcsynopsisinfo>
<funcprototype>
  <funcdef>int <function>max</function></funcdef>
  <varargs/>
</funcprototype>
</funcsynopsis>
```

```
#include <varargs.h>
```

int **max** (...); The `rand` function takes no arguments and returns a pseudorandom integer between 0 and 2^{31}-1:

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
<funcdef>int <function>rand</function></funcdef>
  <void/>
</funcprototype>
</funcsynopsis>
```

int **rand** (); The `qsort` function takes several arguments, including a pointer to a function (the function that should perform the comparison between two elements in order to sort them).

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
  <funcdef>void <function>qsort</function></funcdef>
    <paramdef>void *<parameter>dataptr</parameter>[]</paramdef>
    <paramdef>int <parameter>left</parameter></paramdef>
    <paramdef>int <parameter>right</parameter></paramdef>
    <paramdef>int <parameter>(* comp)</parameter>
      <funcparams>void *, void *</funcparams></paramdef>
</funcprototype>
</funcsynopsis>
```

void **qsort** (*dataptr*, *left*, *right*, *(* comp)*);
void **dataptr*[];
int *left*;
int *right*;
int *(* comp)* (void *, void *); For additional examples, see also `funcdef`, `funcparams`, `paramdef`, `refentry`, `varargs`.

■ funcsynopsisinfo

Name

funcsynopsisinfo – Information supplementing the `FuncDefs` of a `FuncSynopsis`

Synopsis

Mixed Content Model

funcsynopsisinfo ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`textobject`|`lineannotation`)*

Attributes Common attributes

Name Type Default

format linespecific "linespecific"

linenumbering numbered | unnumbered

None

Description

Supplementary information in a `FuncSynopsis`. See `FuncSynopsis`.

Unlike the other info elements, `FuncSynopsisInfo` is not a container for meta-information. Instead `FuncSynopsisInfo` is a verbatim environment for adding additional information to a function synopsis.

Processing expectations

This element is displayed “verbatim”; whitespace and linebreaks within this element are significant.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain funcsynopsisinfo: `funcsynopsis`.

Children

The following elements occur in funcsynopsisinfo: `action`, `anchor`, `application`, `beginpage`, `classname`, `command`, `computeroutput`, `constant`, `database`, `email`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `function`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `lineannotation`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `parameter`, `prompt`, `property`, `remark`, `replaceable`, `returnvalue`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `textobject`, `token`, `type`, `ulink`, `userinput`, `varname`.

Attributes

format The Format attribute applies the `linespecific` notation to all `FuncSynopsisInfo`s. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of a `FuncSynopsisInfo` are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

See Also

`funcdef`, `funcparams`, `funcprototype`, `function`, `paramdef`, `parameter`, `returnvalue`, `varargs`, `void`.

Examples

For examples, see `funcsynopsis`, `refentry`, `varargs`.

■ function

Name

function – The name of a function or subroutine, as in a programming language

Synopsis

Mixed Content Model

function ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry
"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;
%tech.char.class; %title.char.mix;

Description

This element marks up the name of a function. To markup the parts of a function definition, see FuncSynopsis.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

In some environments, the Function element generates additional punctuation, such as a set of trailing parenthesis.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain function: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcdef, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in function: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Function.

See Also

funcdef, funcparams, funcprototype, funcsynopsisinfo, paramdef, parameter, returnvalue, varargs, void.

Examples

For examples, see funcdef, funcparams, funcsynopsis, paramdef, programlistingco, refentry, returnvalue, structname, type, varargs.

■ glossary

Name

glossary – A glossary

Synopsis

Content Model

glossary ::= (glossaryinfo?, (title,subtitle?,titleabbrev?)?, (calloutlist|glosslist| 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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)*, (glossdiv+|glossentry+), bibliography?)

Attributes Common attributes

Name Type Default

status CDATA *None*

Parameter Entities

%nav.class; %partcontent.mix;

Description

A Glossary contains a collection of terms and brief descriptions or definitions of those terms.

Processing expectations

Formatted as a displayed block. A Glossary in a Book frequently causes a forced page break in print media.

Parents

These elements contain glossary: appendix, article, book, chapter, part, preface, sect1, sect2, sect3, sect4, sect5, section.

Children

The following elements occur in glossary: abstract, address, anchor, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossaryinfo, glossdiv, glossentry, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

status Status identifies the editorial or publication status of the `Glossary`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

Examples

```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<glossary><title>Example Glossary</title>
<para>
This is not a real glossary, it’s just an example.
</para>

<!-- ... -->

<glossdiv><title>E</title>

<glossentry id="xml"><glossterm>Extensible Markup Language</glossterm>
  <acronym>XML</acronym>
<glossdef>
  <para>Some reasonable definition here.</para>
  <glossseealso otherterm="sgml">SGML</glossseealso>
</glossdef>
</glossentry>

</glossdiv>

<!-- ... -->

<glossdiv><title>S</title>

<glossentry><glossterm>SGML</glossterm>
<glosssee otherterm="sgml"/>
</glossentry>

<glossentry id="sgml"><glossterm>Standard Generalized
  Markup Language</glossterm><acronym>SGML</acronym>
  <abbrev>ISO 8879:1986</abbrev>
<glossdef>
  <para>Some reasonable definition here.</para>
  <glossseealso otherterm="xml">XML</glossseealso>
</glossdef>
</glossentry>

</glossdiv>
</glossary>
```

■ glossaryinfo

Name

glossaryinfo – Meta-information for a `Glossary`

Synopsis

Content Model

glossaryinfo ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

The GlossaryInfo element is a wrapper for a large collection of meta-information about a Glossary. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain glossaryinfo: glossary.

Children

The following elements occur in glossaryinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

■ glossdef

Name

glossdef – A definition in a GlossEntry

Synopsis

Content Model

glossdef ::= ((calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist|simplelist|variablelist|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| remark|indexterm|beginpage)+, glossseealso*)

Attributes Common attributes

Name Type Default

subject CDATA *None*

Description

A GlossDef contains the description or definition of a GlossTerm.

Processing expectations

Formatted as a displayed block.

Future Changes

The Subject attribute will be renamed Keywords in DocBook V4.0.

Parents

These elements contain glossdef: glossentry.

Children

The following elements occur in glossdef: address, beginpage, blockquote, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, glossseealso, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayou mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, programlisting, programlisting remark, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, table, variablelist.

Attributes

subject Subject holds a list of keywords for the GlossDef's definition.

See Also

glosssee, glossseealso.

Examples

For examples, see glossary, glosslist.

■ glossdiv

Name

glossdiv – A division in a Glossary

Synopsis

Content Model

glossdiv ::= ((title,subtitle?,titleabbrev?), (calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)*, glossentry+)

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

A GlossDiv is a section of a Glossary. A Glossary might be divided into sections in order to group terms, perhaps alphabetically.

A Glossary may contain any number of GlossEntry or GlossDiv elements, but it cannot contain a mixture of both at the same level.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain glossdiv: glossary.

Children

The following elements occur in glossdiv: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossentry, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

status Status identifies the editorial or publication status of the GlossDiv.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

Examples

For examples, see glossary.

■ glossentry

Name

glossentry – An entry in a `Glossary` or `GlossList`

Synopsis

Content Model

glossentry ::= (`glossterm`,`acronym`?,`abbrev`?, (`indexterm`)*, `revhistory`?, (`glosssee`|`glossdef`+))

Attributes Common attributes

Name Type Default

sortas CDATA *None*

Description

`GlossEntry` is a wrapper around a glossary term and its definition.

Processing expectations

Formatted as a displayed block. Glossary entries are usually formatted to highlight the terms and definitions, frequently in a style similar to `VarListEntry`s.

Parents

These elements contain glossentry: `glossary`, `glossdiv`, `glosslist`.

Children

The following elements occur in glossentry: `abbrev`, `acronym`, `glossdef`, `glosssee`, `glossterm`, `indexterm`, `revhistory`.

Attributes

sortas SortAs specifies the string by which the element's content is to be sorted. If unspecified, the proper content is used.

Examples

For examples, see `glossary`, `glosslist`.

■ glosslist

Name

glosslist – A wrapper for a set of `GlossEntry`s

Synopsis

Content Model

glosslist ::= (`glossentry`+)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

%divcomponent.mix; %example.mix; %footnote.mix;
%glossdef.mix; %highlights.mix; %legalnotice.mix;
%list.class; %para.mix; %qandaset.mix;
%refcomponent.mix; %revdescription.mix; %sidebar.mix;
%tabentry.mix; %tbl.entry.mdl; %textobject.mix;

Description

While Glossarys are usually limited to component or section boundaries, appearing at the end of a Book or Chapter, for instance, GlossLists can appear anywhere that the other list types are allowed.

Using a GlossList in running text, instead of a VariableList, for example, maintains the semantic distinction of a Glossary. This distinction may be necessary if you want to automatically point to the members of the list with GlossTerms in the body of the text.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain glosslist: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, footnote, glossary, glossdef, glossdiv, highlights, important, index, informalexample, legalnotice, listitem, msgexplan, msgtext, note, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, warning.

Children

The following elements occur in glosslist: glossentry.

Examples

```
<!DOCTYPE glosslist PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<glosslist>
<glossentry><glossterm>C</glossterm>
<glossdef>
<para>
A procedural programming language invented by K&amp;R.
</para>
</glossdef>
</glossentry>
<glossentry><glossterm>Pascal</glossterm>
<glossdef>
<para>
A procedural programming language invented by Niklaus Wirth.
</para>
</glossdef>
</glossentry>
</glosslist>
```

C A procedural programming language invented by K&R.

Pascal A procedural programming language invented by Niklaus Wirth.

■ glosssee

Name

glosssee – A cross-reference from one `GlossEntry` to another

Synopsis

Mixed Content Model

glosssee ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

otherterm IDREF *None*

Description

`GlossSee` directs the reader to another `GlossEntry` instead of this one. A “See” cross-reference occurs in place of the definition.

Processing Expectations

Formatted as a displayed block, in the same style as a `GlossDef`.

`GlossSee` elements are expected to generate the necessary cross-reference text, usually “See” in English, as well as any necessary punctuation.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain glosssee: `glossentry`.

Children

The following elements occur in glosssee: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`,

personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

otherterm OtherTerm points to the GlossEntry being cross-referenced. (Go see *that* one.)

See Also

glossdef, glossseealso.

Examples

For examples, see glossary.

■ glossseealso

Name

glossseealso – A cross-reference from one GlossEntry to another

Synopsis

Mixed Content Model

glossseealso ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

otherterm IDREF *None*

Description

GlossSeeAlso directs the reader to another GlossEntry for additional information. It is presented in addition to the GlossDef.

Processing expectations

Formatted as a displayed block, in the same style as the GlossDef.

GlossSeeAlso elements are expected to generate the necessary cross-reference text, usually “See also” in English, as well as any necessary punctuation.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain glossseealso: `glossdef`.

Children

The following elements occur in glossseealso: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

otherterm OtherTerm points to the `GlossEntry` being cross-referenced. (Go see *that* one, too.)

See Also

`glossdef`, `glosssee`.

Examples

For examples, see `glossary`.

■ glossterm

Name

glossterm – A glossary term

Synopsis

Mixed Content Model

glossterm ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|
`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|
`trademark`|`wordasword`|`personname`|`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`|`constant`|
`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|
`property`| `replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|
`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`| `author`|`authorinitials`|

corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

baseform CDATA *None*

linkend IDREF *None*

Parameter Entities

%gen.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

Description

GlossTerm identifies a term that appears in a Glossary or GlossList. This element occurs in two very different places: it is both an inline, and a structure element of a GlossEntry. As an inline, it identifies a term defined in a glossary, and may point to it. Within a GlossEntry, it identifies the term defined by that particular entry.

Processing expectations

As an inline, GlossTerms frequently get special typographic treatment, such as italics. In an online environment, they may also form a link (explicitly or implicitly) to the corresponding definition in a glossary.

GlossTerms must not be nested within other GlossTerms. Because DocBook is harmonizing towards XML, this restriction is difficult to enforce with the DTD. The processing of nested GlossTerms is undefined.

As part of a GlossEntry, GlossTerms are usually set as block and separated from the definition.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Future Changes

RevHistory will be removed.

Parents

These elements contain glossterm: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glossentry, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in glossterm: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter,

personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

baseform BaseForm, if specified, contains the root form of the word identified as a GlossTerm. If unspecified, the content of GlossTerm element should be used. BaseForm allows sorting, collating, and indexing to work properly even when different parts of speech are identifed as GlossTerms. For example, <glossterm>sort</glossterm> and <glossterm baseform="sort">sorting</glossterm>.

linkend Linkend points to a related element, generally the GlossEntry that defines it.

See Also

firstterm.

Examples

For examples, see beginpage, firstterm, glossary, glosslist.

■ graphic

Name

graphic – A displayed graphical object (not an inline)

Synopsis

Content Model

graphic ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

valign bottom | middle | top *None*

width CDATA *None*

contentwidth CDATA *None*

BMP
CGM-BINARY
CGM-CHAR
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
IGES
format JPEG
JPG
linespecific
PCX
PIC
PNG
PS
SGML
SVG
TBL
TEX
TIFF
WMF
WPG
None

entityref ENTITY *None*

fileref CDATA *None*

scalefit CDATA *None*

depth CDATA *None*

scale CDATA *None*

contentdepth CDATA *None*

center
align left
right
None

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%equation.content;` `%example.mix;`
`%figure.mix;` `%footnote.mix;` `%glossdef.mix;`
`%indexdivcomponent.mix;` `%info.class;` `%informal.class;`
`%inlineequation.content;` `%listpreamble.mix;` `%para.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%tabentry.mix;` `%tbl.entry.mdl;`
`%tbl.table-main.mdl;` `%tbl.table.mdl;`

Description

This element contains graphical data, or a pointer to an external entity containing graphical data. One of the deficiencies of the DocBook `Graphic` element is that there is no way to specify an alternate text description of the graphic. This has been rectified by the introduction of `MediaObject`.

Processing expectations

Formatted as a displayed block.

There are several ways to provide content for a `Graphic`. It is best to use only one of these methods. However, if multiple graphic sources are provided, the processing expectations are as follows: element content should be used in favor of either EntityRef or FileRef and EntityRef should be used in favor of FileRef.

Future Changes

In DocBook V4.0, `Graphic` will be declared `EMPTY`. This change will require that any embedded graphic content be stored outside the SGML source and pointed to with an EntityRef or FileRef attribute.

Future Changes

In DocBook V5.0, `Graphic` will be discarded. At that time, graphics will have to be incorporated using `MediaObject` or `InlineMediaObject`.

Parents

These elements contain graphic: `answer`, `appendix`, `appendixinfo`, `article`, `articleinfo`, `bibliodiv`, `bibliography`, `bibliographyinfo`, `blockinfo`, `blockquote`, `bookinfo`, `callout`, `caution`, `chapter`, `chapterinfo`, `constraintdef`, `entry`, `equation`, `example`, `figure`, `footnote`, `glossary`, `glossaryinfo`, `glossdef`, `glossdiv`, `graphicco`, `important`, `index`, `indexdiv`, `indexinfo`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `inlineequation`, `itemizedlist`, `listitem`, `msgexplan`, `msgtext`, `note`, `objectinfo`, `orderedlist`, `para`, `partinfo`, `partintro`, `preface`, `prefaceinfo`, `procedure`, `qandadiv`, `qandaset`, `question`, `refentryinfo`, `referenceinfo`, `refsect1`, `refsect1info`, `refsect2`, `refsect2info`, `refsect3`, `refsect3info`, `refsection`, `refsectioninfo`, `refsynopsisdiv`, `refsynopsisdivinfo`, `revdescription`, `screenshot`, `sect1`, `sect1info`, `sect2`, `sect2info`, `sect3`, `sect3info`, `sect4`, `sect4info`, `sect5`, `sect5info`, `section`, `sectioninfo`, `setindex`, `setindexinfo`, `setinfo`, `sidebar`, `sidebarinfo`, `simplesect`, `step`, `synopsis`, `table`, `tip`, `variablelist`, `warning`.

Attributes

align Align specifies the horizontal alignment of the graphic on the page or within the element that frames it.

depth Depth specifies the desired depth (vertical distance, at least in horizontal writing systems) of the image.

entityref EntityRef identifies the general entity which contains (or points to) the content of the graphic.

fileref FileRef specifies the name of the file which contains the content of the graphic.

format Format identifies the format of the graphic content. The Format must be a defined notation.

scale Scale specifies integer representing a percentage scaling factor (retaining the relative dimensions of the original graphic). If unspecified, the value 100 (100%) is assumed.

scalefit If ScaleFit has the value 1 (true), then the graphic is to be scaled (uniformly) to the specified width or depth. The default value of 0 (false) indicates that the image will not be scaled to fit (although it may still be scaled by the Scale attribute).

srccredit SrcCredit contains details about the source of the `Graphic`.

width Width indicates the width of the graphic.

See Also

`alt`, `audioobject`, `caption`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

```
<!DOCTYPE figure PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<figure><title>Notre Dame Cathedral</title>
<graphic srccredit="Norman Walsh, 1998" fileref="figures/notredame.png"/>
</figure>
```

For additional examples, see also `equation`, `figure`, `graphicco`, `informalequation`, `inlineequation`, `screenshot`.

graphicco

Name

graphicco – A graphic that contains callout areas

Synopsis

Content Model

graphicco ::= (`areaspec`,`graphic`,`calloutlist`*)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%informal.class;` `%listpreamble.mix;` `%para.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;`

Description

`Callouts`, such as numbered bullets, are an annotation mechanism. In an online system, these bullets are frequently "hot," and clicking on them sends you to the corresponding annotation.

A `GraphicCO` is a wrapper around an `AreaSpec` and a `Graphic`. An `AreaSpec` identifies the locations (coordinates) on the `Graphic` in which the callouts occur. The `GraphicCO` may also contain the list of annotations in a `CalloutList`, although the `CalloutList` may also occur outside of the wrapper, elsewhere in the document.

Processing expectations

Formatted as a displayed block.

The mandatory processing expectations of a `GraphicCO` are minimal: a system is expected to render the graphic, if possible, and the callout list, if present.

In online environments, the processing system may be able to instantiate the linking relationships between the callout marks on the graphic and the annotations. For example, an HTML presentation system might use the coordinate information to construct a client-side image map. Some processing systems may even be able to go a step further and generate the callout marks automatically from the coordinate information. But this level of sophistication is not mandatory.

Future Changes

In DocBook V5.0, `GraphicCO` will be discarded. In its place, use `MediaObjectCO`.

Parents

These elements contain graphicco: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caution`, `chapter`, `constraintdef`, `example`, `figure`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `indexdiv`, `informalexample`, `informalfigure`, `itemizedlist`, `listitem`, `msgexplan`, `msgtext`, `note`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `screenshot`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `tip`, `variablelist`, `warning`.

Children

The following elements occur in graphicco: `areaspec`, `calloutlist`, `graphic`.

See Also

`areaspec`, `calloutlist`, `co`, `coref`, `imageobjectco`, `mediaobjectco`, `programlistingco`, `screenco`.

Examples

```
<!DOCTYPE graphicco PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<graphicco>
<areaspec units="calspair">
  <areaset id="oneway" coords="">
    <area id="oneway1" coords="300 400"/>
    <area id="oneway2" coords="325 340"/>
  </areaset>
  <area id="myhouse" coords="425 590"/>
</areaspec>
<!-- This is not a real URL! -->
<graphic fileref="http://maps.example.com/EARTH?USA?MA?AMHERST"/>
</graphicco>
```

■ group

Name

group – A group of elements in a `CmdSynopsis`

Synopsis

Content Model

group ::= ((`arg`|`group`|`option`|`synopfragmentref`|`replaceable`|`sbr`)+)

Attributes Common attributes

Name Type Default

rep norepeat | repeat "norepeat"

choice opt | plain | req "opt"

Description

A `Group` surrounds several related items. Usually, they are grouped because they are mutually exclusive. The user is expected to select one of the items.

Processing expectations

Formatted inline. The additional processing expectations of a `Group` are significant. For a complete discussion, see `CmdSynopsis`.

- Multiple arguments within a group are considered exclusive and are separated by vertical bars.
- Brackets are used to distinguish between optional, required, or plain arguments. Usually square brackets are placed around optional arguments, `[-f | -g]`, and curly brackets are placed around required arguments, `{-f | -g}`. Plain arguments are required, but are not decorated with brackets.
- Repeatable arguments are followed by an ellipsis.

Future Changes

The `OptMult` and `ReqMult` values for the Choice attribute will be removed in DocBook V4.0. Use the Rep attribute instead to indicate that the choice is repeatable.

Parents

These elements contain group: `arg`, `cmdsynopsis`, `group`, `synopfragment`.

Children

The following elements occur in group: `arg`, `group`, `option`, `replaceable`, `sbr`, `synopfragmentref`.

Attributes

choice Choice indicates whether the `Arg` is required (`Req` or `Plain`) or optional (`Opt`). Arguments identified as `Plain` are required, but are shown without additional decoration. The `OptMult` and `ReqMult` choices will be removed in DocBook V4.0; use the Rep attribute instead.

rep A Rep value of `Repeat` indicates that the `Group` is reapeatable. This is frequently rendered with an ellipsis.

See Also

`arg`, `cmdsynopsis`, `refsynopsisdiv`, `sbr`, `synopfragment`, `synopfragmentref`.

Examples

For examples, see `cmdsynopsis`, `synopfragment`.

■ guibutton

Name

guibutton – The text on a button in a GUI

Synopsis

Mixed Content Model

guibutton ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage|accel)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

GUIButton identifies the text that appears on a button in a graphical user interface.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guibutton: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, menuchoice, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in guibutton: accel, beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the GUIButton.

See Also

accel, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, keycap, keycode, keycombo, keysym, menuchoice, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The exact text of the <guilabel>Legend</guilabel> and other labels on the
graph is dependent upon the language of the current locale.  Likewise,
the text of the <guibutton>OK</guibutton> button and other buttons may vary.
The <guiicon><inlinegraphic fileref="figures/legend.png"/></guiicon>
icon and the other icons on the left side of the display may be
configured by the local administrator, but they are not
generally expected to vary from locale to locale.
</para>
```

The exact text of the **Legend** and other labels on the graph is dependent upon the language of the current locale.

Likewise, the text of the **OK** button and other buttons may vary. The  icon and the other icons on the left side of the display may be configured by the local administrator, but they are not generally expected to vary from locale to locale. For additional examples, see also `guiicon`, `guilabel`.

■ guiicon

Name

guiicon – Graphic and/or text appearing as a icon in a GUI

Synopsis

Mixed Content Model

guiicon ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`accel`)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

`GUIIcon` identifies a graphic or text icon that appears in a graphical user interface.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guiicon: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `menuchoice`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`,

primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in guiicon: accel, beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the GUIIcon.

See Also

accel, guibutton, guilabel, guimenu, guimenuitem, guisubmenu, keycap, keycode, keycombo, keysym, menuchoice, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The exact text of the <guilabel>Legend</guilabel> and other labels on the
graph is dependent upon the language of the current locale.  Likewise,
the text of the <guibutton>OK</guibutton> button and other buttons may vary.
The <guiicon><inlinegraphic fileref="figures/legend.png"/></guiicon>
icon and the other icons on the left side of the display may be
configured by the local administrator, but they are not
generally expected to vary from locale to locale.
</para>
```

The exact text of the **Legend** and other labels on the graph is dependent upon the language of the current locale.

Likewise, the text of the **OK** button and other buttons may vary. The  icon and the other icons on the left side of the display may be configured by the local administrator, but they are not generally expected to vary from locale to locale. For additional examples, see also guibutton, guilabel.

■ guilabel

Name

guilabel – The text of a label in a GUI

Synopsis

Mixed Content Model

guilabel ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage|accel)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

GUILabel identifies text that appears as a label in a graphical user interface.

What constitutes a label may vary from application to application. In general, any text that appears in a GUI may be considered a label, for example a message in a dialog box or a window title.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guilabel: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, menuchoice, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in guilabel: accel, beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the GUILabel.

See Also

accel, guibutton, guiicon, guimenu, guimenuitem, guisubmenu, keycap, keycode, keycombo, keysym, menuchoice, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The exact text of the <guilabel>Legend</guilabel> and other labels on the
graph is dependent upon the language of the current locale.  Likewise,
the text of the <guibutton>OK</guibutton> button and other buttons may vary.
The <guiicon><inlinegraphic fileref="figures/legend.png"/></guiicon>
icon and the other icons on the left side of the display may be
configured by the local administrator, but they are not
```

```
generally expected to vary from locale to locale.
</para>
```

The exact text of the **Legend** and other labels on the graph is dependent upon the language of the current locale.

Likewise, the text of the **OK** button and other buttons may vary. The  icon and the other icons on the left side of the display may be configured by the local administrator, but they are not generally expected to vary from locale to locale. For additional examples, see also `guibutton`, `guiicon`.

■ guimenu

Name

guimenu – The name of a menu in a GUI

Synopsis

Mixed Content Model

guimenu ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`accel`)*

Attributes Common attributes

Name Type Default

moreinfo none
refentry

"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

`GUIMenu` identifies a menu name in a graphical user interface. In particular, this is distinct from a menu item (`GUIMenuItem`), which is terminal, and a submenu (`GUISubmenu`), which occurs as a selection from a menu.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guimenu: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `menuchoice`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in guimenu: `accel`, `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `GUIMenu`.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenuitem`, `guisubmenu`, `keycap`, `keycode`, `keycombo`, `keysym`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
You can exit from GNU Emacs with the keyboard shortcut
<keycombo><keysym>C-c</keysym><keysym>C-x</keysym></keycombo>
or by selecting <guimenuitem>Exit Emacs</guimenuitem> from
the <guimenu>Files</guimenu> menu.
</para>
```

You can exit from GNU Emacs with the keyboard shortcut **C-c–C-x** or by selecting **Exit Emacs** from the **Files** menu. For additional examples, see also `accel`, `guimenuitem`, `guisubmenu`, `menuchoice`, `shortcut`.

■ guimenuitem

Name

guimenuitem – The name of a terminal menu item in a GUI

Synopsis

Mixed Content Model

guimenuitem ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`accel`)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

`GUIMenuItem` identifies a terminal selection from a menu in a graphical user interface. In particular, this is distinct from a menu (`GUIMenu`) and a submenu (`GUISubmenu`). The distinction between a `GUIMenuItem` and a `GUISubmenu` is simply whether or not the selection is terminal or leads to an additional submenu.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guimenuitem: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, menuchoice, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in guimenuitem: accel, beginpage, indexterm, inlinegraphic, inlinemediaobje replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the GUIMenuItem.

See Also

accel, guibutton, guiicon, guilabel, guimenu, guisubmenu, keycap, keycode, keycombo, keysym, menuchoice, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
You can exit from GNU Emacs with the keyboard shortcut
<keycombo><keysym>C-c</keysym><keysym>C-x</keysym></keycombo>
or by selecting <guimenuitem>Exit Emacs</guimenuitem> from
the <guimenu>Files</guimenu> menu.
</para>
```

You can exit from GNU Emacs with the keyboard shortcut **C-c**–**C-x** or by selecting **Exit Emacs** from the **Files** menu. For additional examples, see also accel, action, guimenu, guisubmenu, menuchoice, shortcut.

■ guisubmenu

Name

guisubmenu – The name of a submenu in a GUI

Synopsis

Mixed Content Model

guisubmenu ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage|accel)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

The name of a submenu in a graphical user interface is identified by the `GUISubmenu` element. A submenu is a menu invoked from another menu that leads either to terminal items (`GUIMenuItem`s) or additional submenus.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain guisubmenu: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `menuchoice`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in guisubmenu: `accel`, `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `GUISubMenu`.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `keycap`, `keycode`, `keycombo`, `keysym`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In GNU Emacs, the <guimenuitem>Print Buffer</guimenuitem> command is
located off of the <guisubmenu>Print</guisubmenu> submenu of the
<guimenu>Tools</guimenu> menu.
</para>
```

In GNU Emacs, the **Print Buffer** command is located off of the **Print** submenu of the **Tools** menu.

■ hardware

Name

hardware – A physical part of a computer system

Synopsis

Mixed Content Model

hardware ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

Hardware identifies some physical component of a computer system. Even though DocBook provides a broad range of inlines for describing the various software components of a system, it provides relatively few for describing hardware.

If you need to identify a number of different hardware components, you may wish to consider extending DocBook, or at least using the Role attribute to further classify Hardware.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

In DocBook V4.0, the content model of Hardware will be constrained to (#PCDATA | Replaceable | InlineGraphic).

Parents

These elements contain hardware: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in hardware: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel,

menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Hardware.

See Also

application, database, filename, medialabel, productname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <acronym>IRQ</acronym> of the <hardware>SCSI Controller</hardware>
can be set to 7, 11, or 15.  The factory default setting is 7.
</para>
```

The IRQ of the SCSI Controller can be set to 7, 11, or 15. The factory default setting is 7.

■ highlights

Name

highlights – A summary of the main points of the discussed component

Synopsis

Content Model

highlights ::= ((calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist| simplelist|variablelist|caution|important|note|tip|warning| formalpara|para| simpara|indexterm)+)

Attributes Common attributes

Parameter Entities

%bookcomponent.content; %component.mix; %divcomponent.mix;

%genobj.class; %listpreamble.mix; %qandaset.mix;

%refcomponent.mix; %revdescription.mix; %sidebar.mix;

Description

Highlights are generally presented at the beginning of a component and offer some sort of summary of the main points that will be discussed.

Processing expectations

Formatted as a displayed block. Highlights often contain some sort of list.

Parents

These elements contain highlights: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, chapter, glossary, glossdiv, index, itemizedlist, listitem, msgexplan, msgtext, orderedlist, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, variablelist.

Children

The following elements occur in highlights: calloutlist, caution, formalpara, glosslist, important, indexterm, itemizedlist, note, orderedlist, para, segmentedlist, simpara, simplelist, tip, variablelist, warning.

See Also

abstract, blockquote, epigraph, sidebar.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Example Chapter</title>
<highlights>
<para>
This chapter will teach you
<itemizedlist>
<listitem>
<para>
How to disassemble an automobile.
</para>
</listitem>
<listitem>
<para>
How to properly carry the component pieces.
</para>
</listitem>
<listitem>
<para>
How to reassemble an automobile in a standard telephone booth.
</para>
</listitem>
</itemizedlist>
</para>
</highlights>
<para>&hellip</para>
</chapter>
```

■ holder

Name

holder – The name of the individual or organization that holds a copyright

Synopsis

Mixed Content Model

holder ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark| subscript| superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Description

`Holder` in `Copyright` identifies an individual or organization that asserts a copyright on the document.

Processing expectations

The formatting of `Holder` depends on the formatting of its parent `Copyright`. In the case of a `Copyright` with multiple holders, additional punctuation may need to be generated when `Holder` is processed.

Parents

These elements contain holder: `copyright`.

Children

The following elements occur in holder: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

For examples, see `bibliography`, `biblioset`, `bookinfo`, `copyright`.

■ honorific

Name

honorific – The title of a person

Synopsis

Mixed Content Model

honorific ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;` `%person.ident.mix;`

Description

An `Honorific` occurs in the name of an individual. It is the honorific title of the individual, such as “Dr.,” “Mr.,” or “Ms.”

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

On some systems, `Honorific` may generate the trailing period automatically.

Parents

These elements contain honorific: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `personname`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in honorific: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `firstname`, `lineage`, `othername`, `surname`.

Examples

For examples, see `author`, `authorgroup`.

■ html:form

Name

html:form – An HTML form

Synopsis

If the HTML Forms Module is used, `html:form` elements can be used in most of the places where `para` can be used.

Description

XHTML `http://www.w3.org/TR/xhtml-basic/` defines a `html:form` element that can be used to add interactivity to web pages. The HTML Forms Module for DocBook allows `html:form` elements to be used in DocBook documents.

A complete description of XHTML is outside the scope of this reference.

Examples

```
<!DOCTYPE section PUBLIC "-//OASIS//DTD DocBook HTML Forms Module V1.0//EN"
               "dbforms.dtd">
<section><title>HTML Forms Test</title>

<para><html:input/>ClassSynopsis version:</para>

<html:form action="dosomething">
<para>foo<html:button>xxx</html:button>
</para>
</html:form>
</section>
```

■ imagedata

Name

imagedata – Pointer to external image data

Synopsis

Content Model

imagedata ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

valign
- bottom
- middle
- top

None

width CDATA *None*

contentwidth CDATA *None*

format
- BMP
- CGM-BINARY
- CGM-CHAR
- CGM-CLEAR
- DITROFF
- DVI
- EPS
- EQN
- FAX
- GIF
- GIF87a
- GIF89a
- IGES
- JPEG
- JPG
- linespecific
- PCX
- PIC
- PNG
- PS
- SGML
- SVG
- TBL
- TEX
- TIFF
- WMF
- WPG

None

entityref ENTITY *None*

fileref CDATA *None*

scalefit CDATA *None*

depth CDATA *None*

scale CDATA *None*

contentdepth CDATA *None*

align
- center
- left
- right

None

Description

This element points to an external entity containing graphical image data.

Processing expectations

Render the image. May be formatted inline or as a displayed block, depending on context.

There are two ways to provide content for `ImageData`: EntityRef or FileRef. It is best to use only one of these methods, however, if multiple sources are provided, EntityRef will be used in favor of FileRef.

`ImageData` provides a selection of attributes that can be used to control how the image is rendered. These attributes define two rectangles, the viewport area and the content area, and how these rectangles are related to each other. The intrinsic size of the image is a third rectangle that sometimes influences the way an image is rendered.

It is important to understand the distinction between these three areas. When rendering an image, the viewport area defines the space reserved in the flow of content for the image. If a 6in x 4in viewport area is specified, that's how much space will be reserved for the image, independent of the actual size of the rendered image. The content area defines the actual size of the rendered image, independent of the intrinsic size of the image. The intrinisic size of the image is its actual, real size.

DocBook provides three mutually exclusive mechanisms for specifying the content area of an image: it can be specified directly, it can be specified by selecting a scale factor, or it can be specified to be the same size as the viewport area.

Finally, DocBook provides two attributes, align and valign to specify the alignment of the content area within the viewport area.

DocBook provides no mechanism for specifying how an image should be rendered if the content area exceeds the viewport area in either or both dimensions. Implementations are free to perform clipping, allow the image to overflow, and/or generate errors.

Units of Measure The size of the viewport area and the content area are defined in terms of lengths (width and depth).

Lengths must be expressed as a decimal value followed immediately by an optional unit of measure or a percentage. Six and one eight inches, for example, must be expressed as "6.125in". It is an error to put a space or other punctuation between the decimal value and the unit of measure.

The following units of measure may be used:

`pt`	Points (1/72 of an inch)
`cm`	Centimeters
`mm`	Millimeters
`in`	Inches
`pc`	Picas (1/6 of an inch)
`px`	Pixels
`em`	Ems

If no unit of measure is provided, `px` is assumed. Note that pixels have no universally accepted absolute size and ems are relative units of measure. Implementations may define pixel sizes differently and stylesheets may or may not be able to determine the current font size in order to correctly calculate the absolute size of an em. It is best to avoid these units of measure.

Percetages are expressed as a decimal value followed immediately by a `%` sign.

Specifying the Viewport Area The viewport area is specified by the width and depth attributes.

If neither width nor depth is specified, an implementation is free to choose defaults. These defaults may be influenced by context. For example, when rendering an inline graphic, the viewport area often defaults to the size of the content area. For block graphics, the width often defaults to the column width while the depth defaults to the depth of the content area.

If only one of width or depth is specified, an implementation is free to choose a default for the other dimension.

Viewport area dimensions expressed as a percentage are a percentage of the available area. For example, a width of `50%` when an implementation is rendering in a column 6in wide is equivalent to specifying a width of 3in.

Percentages must be used with care. Some media are unbounded in one or more directions (for example, web pages are generally unbounded in depth). Specifying a percentage of an unbounded dimension is undefined. Implementations may choose arbitrary defaults or may generate errors.

Specifing the Content Area The content area is specified by the contentwidth and contentdepth attributes.

If neither content width nor content depth is specified, an implementation is expected to render the image at its intrinsic size (unless scaling or scaling to fit is requested). If only one of content width or content depth is specified, an implementation is expected to choose a default for the other dimension such that the image is scaled proportionally. For example, if an image has an intrinsic size of one square inch and the content width is specified as `2in`, the content depth must default to `2in`.

Content area dimensions expressed as a percentage are a percentage of the intrinsic size of the image.

Percentages must be used with care. Some implementations may be unable to determine the intrinsic size of an image and will therefore be forced to make compromises. Implementations may choose arbitrary values or may generate errors if the intrinsic size cannot be obtained.

Scaling There are two ways that scaling can be specified, with the scale attribute or with the scalefit attribute.

If scale is specified, it must be a positive integer. It is always interpreted to be a percentage value where "`100`" represents 100%.

The legal values of scalefit are `0` (false) or `1` (true). If scaling to fit is requested, the content area is scaled until *either* the content width is the same as the viewport width (and the content depth is less than or equal to the viewport depth) *or* the content depth is the same as the viewport depth (and the content width is less than or equal to the viewport width), whichever comes first. In other words, scaling to fit never causes anamorphic scaling, it simply scales the image as large as possible without overflowing the bounds of the viewport area.

Specification of content area, scaling, and scaling to fit are mutually exclusive. If a content area (contentwidth, contentdepth, or both) is specified, *both* scaling and scaling to fit are ignored. If the content area is not specified and both scaling and scaling to fit are specified, scalefit is ignored.

In order to achieve a level of backwards compatibility with previous versions of DocBook (which did not have attributes for specifying a content area) while maintaining coherent semantics, the default value of scalefit depends on other attributes:

Viewport area	Content area	scalefit default
unspecified	unspecified	irrelevant
specified	unspecified	1
unspecified	specified	0
specified	specified	0

If a viewport area is specified (and neither a content area nor scaling is specified) and scalefit is explicitly "`0`", the viewport area specification must be ignored.

Alignment Two alignment attributes are provided, align and valign.

If specified, align indicates how the content area should be aligned horizontally within the viewport area. If not specified, implementations are free to choose any default value.

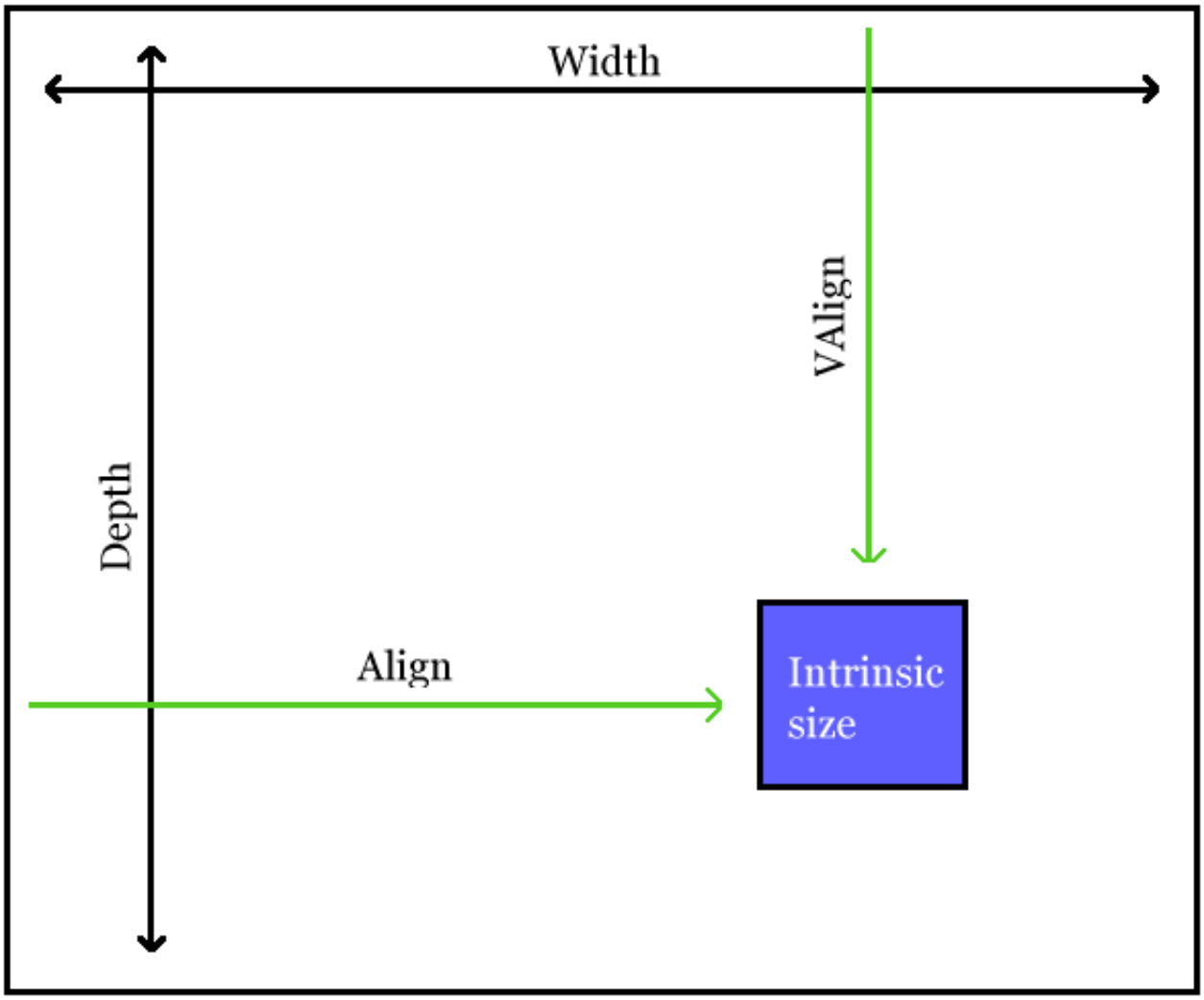
If specified, valign indicates how the content area should be aligned vertically within the viewport area. If not specified, implementations are free to choose any default value.

Examples If nothing is specified about the size of an image, it is rendered in a content area that is the same as its intrinsic size in a viewport area that is implementation defined:

```
<imagedata fileref="image.png"/>
```

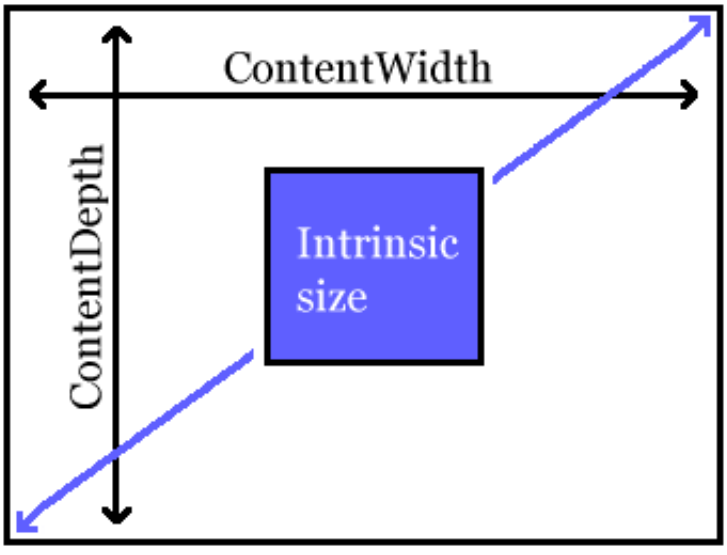
If a viewport area is specified, the image is rendered in a content area that is the same as its intrinsic size in the specified viewport area:

```
<imagedata fileref="image.png" width="6in" depth="5.5in" scalefit="0"/>
```



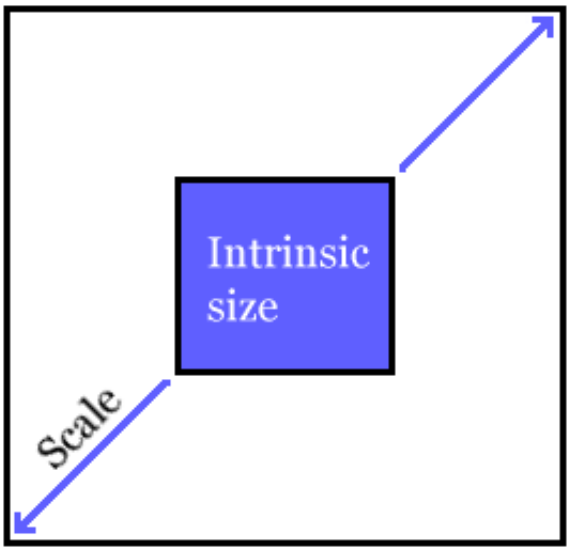
If a content area is specified, the image is scaled (possibly anamorphically) to that size and rendered in a viewport area that is implementation defined:

```
<imagedata fileref="image.png" contentwidth="4in" contentdepth="3in"/>
```



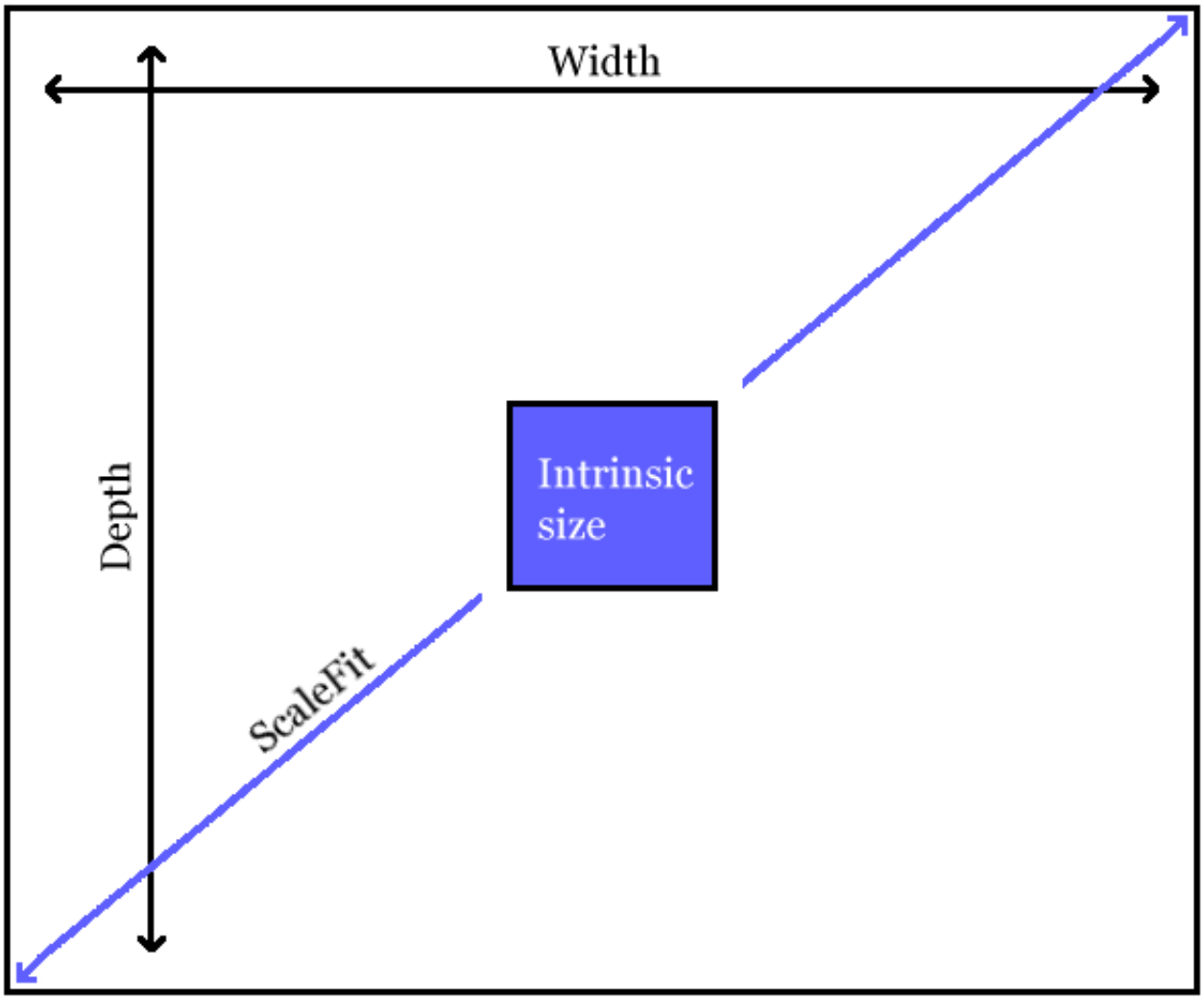
If a scaling factor is specified, the intrinsic size is scaled uniformly by that amount to obtain the content area which is rendered in a viewport area that is implementation defined:

```
<imagedata fileref="image.png" scale="300"/>
```



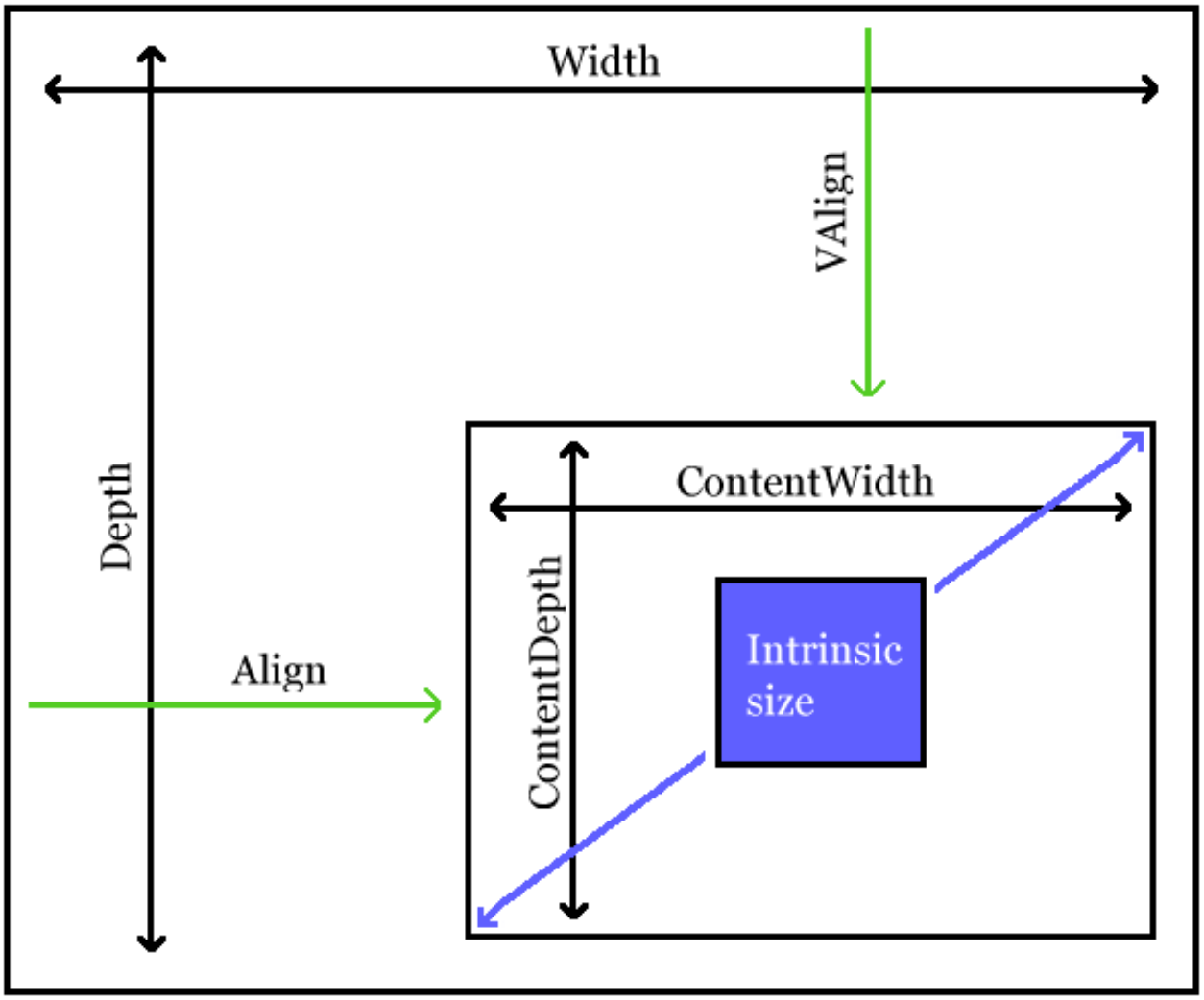
If a viewport area is specified and scaling to fit is requested, the intrinsic size is scaled (uniformly) as large as possible without extending beyond the bounds of the viewport area which is rendered as specified.

```
<imagedata fileref="image.png" width="6in" depth="5.5in"/>
<!-- note that scalefit="1" is the default in this case -->
```



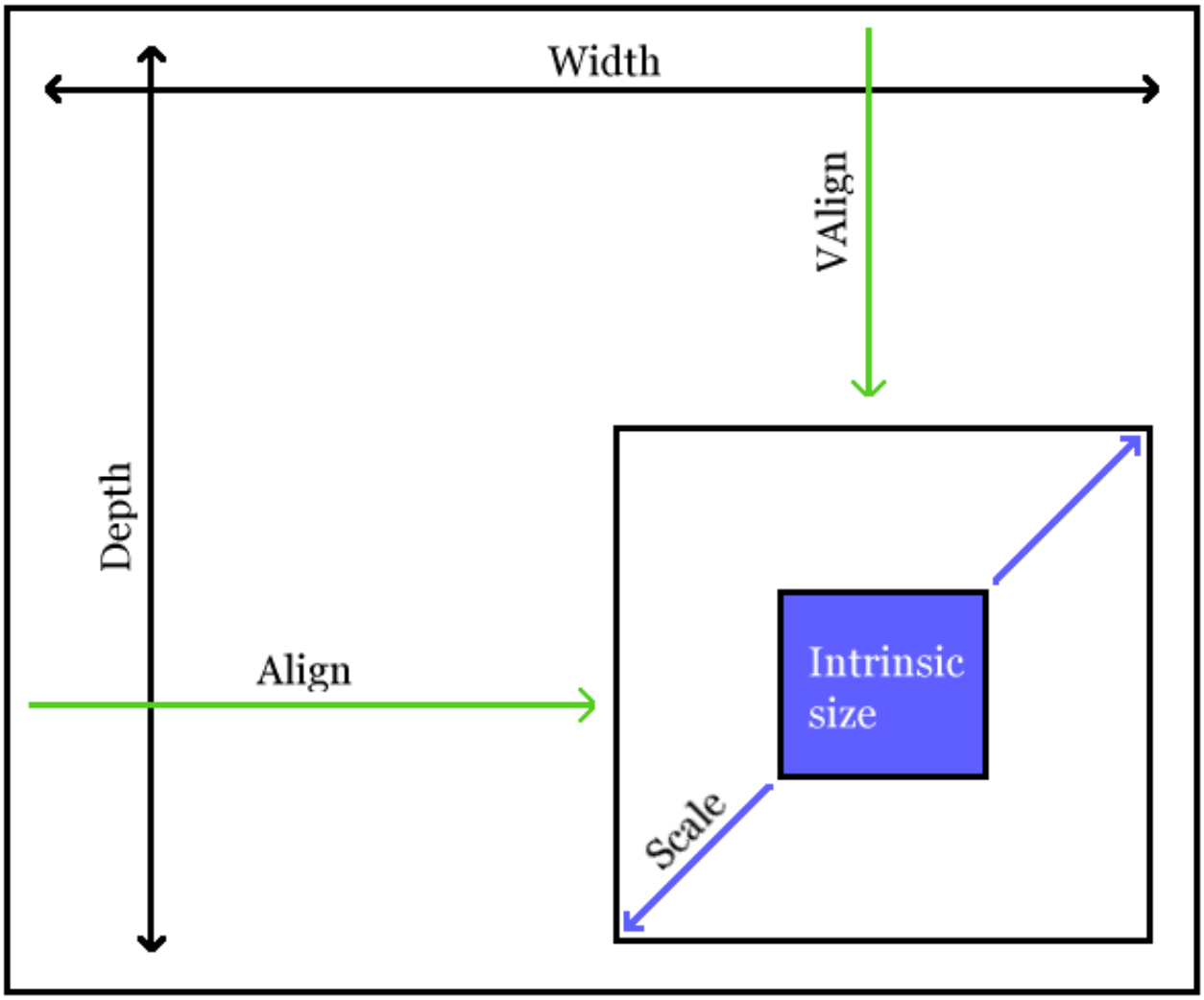
If the viewport area and content area are specified, the image is scaled (possibly anamorphically) to the content area size and rendered in the specified viewport area:

```
<imagedata fileref="image.png" width="6in" depth="5.5in"
           contentwidth="4in" contentdepth="3in"/>
```



If the viewport area and a scaling factor are specified, the intrinsic size is scaled uniformly by the scaling factor amount to obtain the content area which is rendered in the specified viewport area:

```
<imagedata fileref="image.png" width="6in" depth="5.5in" scale="300"/>
```



Parents

These elements contain imagedata: `imageobject`.

Attributes

align Align specifies the horizontal alignment of the content area in the viewport area.

contentdepth ContentDepth specifies the desired depth of the content area.

contentwidth ContentWidth specifies the desired width of the content area.

depth Depth specifies the desired depth of the viewport area.

entityref EntityRef identifies the general entity which points to the content of the image data.

fileref FileRef specifies the name of the file which contains the content of the image data.

format Format identifies the format of the image data. The Format must be a defined notation.

scale Scale is an integer representing a percentage scaling factor (retaining the relative dimensions of the original image). If unspecified, the value 100 (100%) is assumed.

scalefit If ScaleFit has the value 1 (true), then the image data is to be scaled (uniformly) to the specified width or depth. The default value of 0 (false) indicates that the image will not be scaled to fit (although it may still be scaled by the Scale attribute).

srccredit SrcCredit contains details about the source of the image data.

width Width indicates the width of the graphic.

Examples

For examples, see `imageobject`, `informalfigure`, `inlinemediaobject`, `mediaobjectco`, `videoobject`.

■ imageobject

Name

imageobject – A wrapper for image data and its associated meta-information

Synopsis

Content Model

imageobject ::= (`objectinfo`?,`imagedata`)

Attributes Common attributes

Parameter Entities

`%mediaobject.mix;`

Description

An `ImageObject` is a wrapper containing `ImageData` and its associated meta-information.
If the SVG Module is used, `ImageObject` can also contain the `svg:svg` element.

Processing expectations

May be formatted inline or as a displayed block, depending on context. It might not be rendered at all, depending on its placement within a `MediaObject` or `InlineMediaObject` and the constraints on the publishing system. For a more detailed description of the semantics involved, see `MediaObject`.

Parents

These elements contain imageobject: `imageobjectco`, `inlinemediaobject`, `mediaobject`.

Children

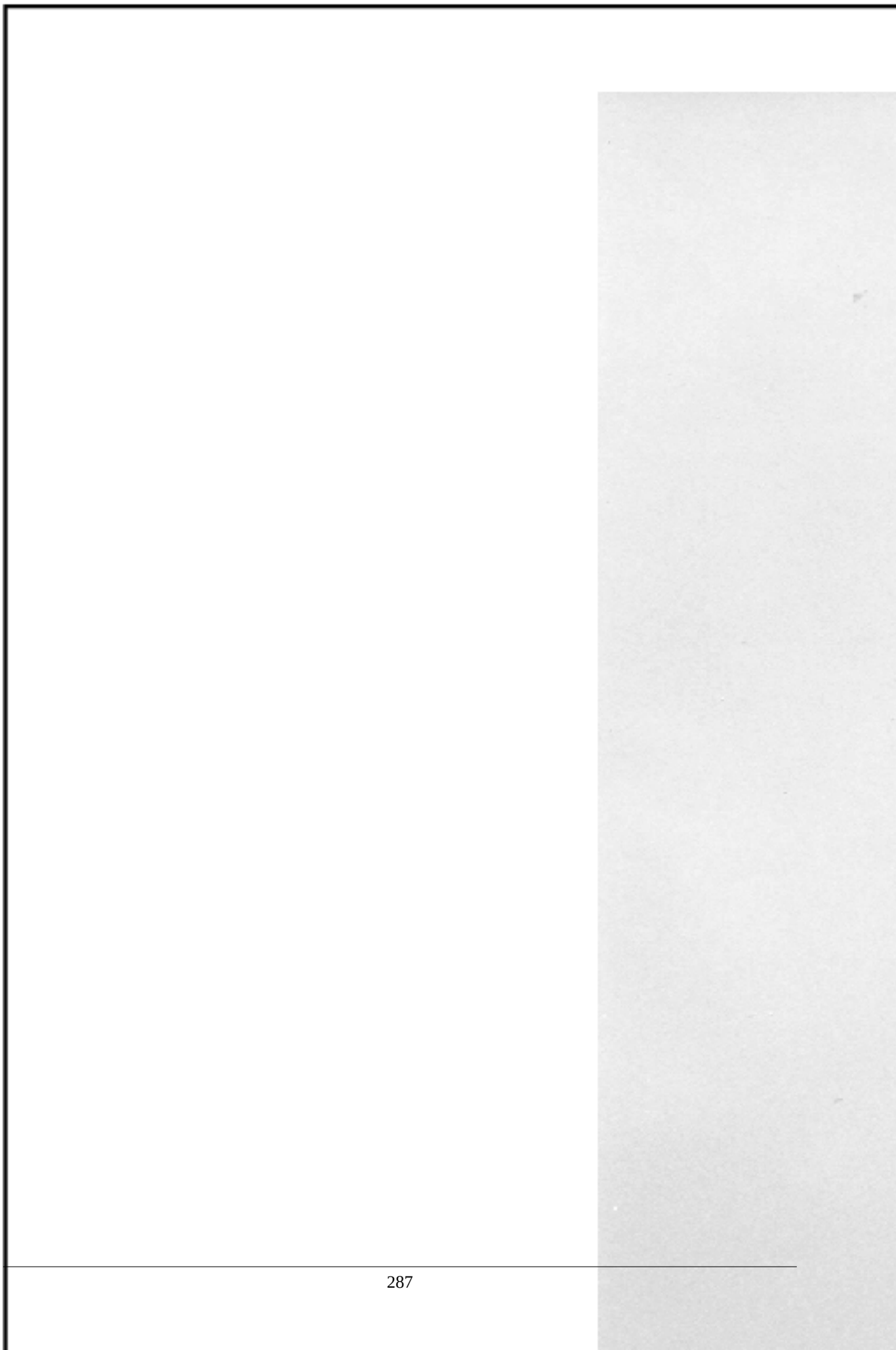
The following elements occur in imageobject: `imagedata`, `objectinfo`.

See Also

`alt`, `audioobject`, `caption`, `graphic`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

```
<!DOCTYPE mediaobject PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<mediaobject>
  <imageobject>
    <imagedata fileref="figures/eiffeltower.eps" format="EPS"/>
  </imageobject>
  <imageobject>
    <imagedata fileref="figures/eiffeltower.png" format="PNG"/>
  </imageobject>
  <textobject>
    <phrase>The Eiffel Tower</phrase>
  </textobject>
  <caption>
   <para>Designed by Gustave Eiffel in 1889, The Eiffel Tower is one of the
most widely recognized buildings in the world.
</para>
  </caption>
</mediaobject>
```



videoobject.

imageobjectco

Name

imageobjectco – A wrapper for an image object with callouts

Synopsis

Content Model

imageobjectco ::= (areaspec,imageobject,calloutlist*)

Attributes Common attributes

Description

Callouts, such as numbered bullets, are an annotation mechanism. In an online system, these bullets are frequently "hot," and clicking on them navigates to the corresponding annotation.

A ImageObjectCO is a wrapper around an AreaSpec and an ImageObject. An AreaSpec identifies the locations (coordinates) on the image where the Callouts occur. The ImageObjectCO may also contain the list of annotations in a CalloutList, although the CalloutList may also occur outside of the wrapper, elsewhere in the document.

Processing expectations

Formatted as a displayed block. It may not be rendered at all, depending on its placement within the MediaObject that contains it and the constraints on the publishing system. For a more detailed description of the semantics involved, see MediaObject.

The mandatory processing expectations of a ImageObjectCO are minimal: a system is expected to render the image, if possible, and the callout list, if present.

In online environments, the processing system may be able to instantiate the linking relationships between the callout marks on the image and the annotations. For example, an HTML presentation system might use the coordinate information to construct a client-side image map. Some processing systems may even be able to go a step further and generate the callout marks automatically from the coordinate information. But this level of sophistication is not mandatory.

Future Changes

In DocBook V5.0, ImageObjectCO will be discarded. In its place, use MediaObjectCO.

Parents

These elements contain imageobjectco: mediaobjectco.

Children

The following elements occur in imageobjectco: areaspec, calloutlist, imageobject.

See Also

areaspec, calloutlist, co, coref, graphicco, mediaobjectco, programlistingco, screenco.

Examples

For examples, see mediaobjectco.

■ important

Name

important – An admonition set off from the text

Synopsis

Content Model

important ::= (`title`?, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`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`|`sidebar`|`anchor`|`bridgehead`|`remark`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Parameter Entities

`%admon.class;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%highlights.mix;` `%legalnotice.mix;`

`%listpreamble.mix;` `%para.mix;` `%qandaset.mix;`

`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`

`%tabentry.mix;` `%tbl.entry.mdl;` `%textobject.mix;`

Description

`Important` is an admonition set off from the main text.

In some types of documentation, the semantics of admonitions are clearly defined (`Caution` might imply the possibility of harm to equipment whereas `Warning` might imply harm to a person), but DocBook makes no such assertions.

Processing expectations

Formatted as a displayed block. It often outputs the generated text “Important” or some other visible indication of the type of admonition, especially if a `Title` is not present. Sometimes outputs a graphical icon or other symbol as well.

Parents

These elements contain important: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `chapter`, `colophon`, `dedication`, `entry`, `glossary`, `glossdiv`, `highlights`, `index`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `variablelist`.

Children

The following elements occur in important: `address`, `anchor`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `orderedlist`, `para`, `procedure`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `synopsis`, `table`, `title`, `variablelist`.

See Also

`caution`, `note`, `tip`, `warning`.

Examples

```
<!DOCTYPE important PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<important>
<para>
No user-servicable parts inside. Breaking this seal voids all warranties.
</para>
</important>
```

IMPORTANT

No user-servicable parts inside. Breaking this seal voids all warranties.

■ index

Name

index – An index

Synopsis

Content Model

index ::= (`indexinfo`?, (`title`,`subtitle`?,`titleabbrev`?)?, (`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)*, (`indexdiv`*|`indexentry`*))

Attributes Common attributes

Parameter Entities

`%index.class;` `%nav.class;` `%partcontent.mix;`

Description

An `Index` contains the formatted index of a document. An index may begin with introductory material, followed by any number of `IndexEntry`s or `IndexDiv`s.

Processing expectations

Formatted as a displayed block. An `Index` in a `Book` frequently causes a forced page break in print media.

In many processing systems, indexes are generated automatically or semiautomatically and never appear instantiated as DocBook markup.

Future Changes

Formal objects and other elements inappropriate for an index will be removed from the content that can appear before the first `IndexDiv` or `IndexEntry`.

Parents

These elements contain index: `appendix`, `article`, `book`, `chapter`, `part`, `preface`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`.

Children

The following elements occur in index: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynop` `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexdiv`, `indexentry`, `indexinfo`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Examples

```
<!DOCTYPE index PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<index><title>Index</title>
<indexdiv><title>D</title>
<indexentry>
  <primaryie>database (bibliographic), 253, 255</primaryie>
  <secondaryie>structure, 255</secondaryie>
  <secondaryie>tools, 259</secondaryie>
</indexentry>
<indexentry>
  <primaryie>dates (language specific), 179</primaryie>
</indexentry>
<indexentry>
  <primaryie>DC fonts, <emphasis>172</emphasis>, 177</primaryie>
  <secondaryie>Math fonts, 177</secondaryie>
</indexentry>
</indexdiv>
</index>


<!DOCTYPE index PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<index>
<indexentry>
  <primaryie>Example</primaryie>
  <secondaryie>Chapter</secondaryie>
  <seeie>Example Chapter</seeie>
</indexentry>

<indexentry>
  <primaryie>Example Chapter, 35-48</primaryie>
  <seealsoie>Examples</seealsoie>
</indexentry>

<indexentry>
  <primaryie>Examples, 18, 36, 72-133</primaryie>
```

```
</indexentry>

</index>
```

■ indexdiv

Name

indexdiv – A division in an index

Synopsis

Content Model

indexdiv ::= ((`title`,`subtitle`?,`titleabbrev`?)?, ((`itemizedlist`|`orderedlist`|`variablelist`|`simplelist`|`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`|`anchor`|`remark`|`link`|`olink`|`ulink`|`beginpage`)*, (`indexentry`+|`segmentedlist`)))

Attributes Common attributes

Description

An `IndexDiv` identifies a section of an `Index`. An index might be divided into sections in order to group entries, usually alphabetically.

An index may contain any number of `IndexEntry` or `IndexDiv` elements, but it cannot contain a mixture of both at the same level.

Processing expectations

Formatted as a displayed block.

Future Changes

Formal objects and other elements inappropriate for an index will be removed the content that can appear before the first `IndexEntry`.

Parents

These elements contain indexdiv: `index`, `setindex`.

Children

The following elements occur in indexdiv: `address`, `anchor`, `beginpage`, `blockquote`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `fieldsynopsis`, `formalpara`, `funcsynopsis`, `graphic`, `graphicco`, `indexentry`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `link`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `olink`, `orderedlist`, `para`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `title`, `titleabbrev`, `ulink`, `variablelist`.

Examples

For examples, see `index`.

■ indexentry

Name

indexentry – An entry in an index

Synopsis

Content Model

indexentry ::= (`primaryie`, (`seeie`|`seealsoie`)*, (`secondaryie`, (`seeie`|`seealsoie`|`tertiaryie`)*)*)

Attributes Common attributes

Description

An `IndexEntry` wraps all of the index terms associated with a particular primary index term. This includes an arbitrary list of secondary and tertiary elements as well as See and SeeAlso elements.

Processing expectations

Formatted as a displayed block. A rendered index usually places secondary items under primary items and tertiary items under secondary.

Parents

These elements contain indexentry: `index`, `indexdiv`, `setindex`.

Children

The following elements occur in indexentry: `primaryie`, `secondaryie`, `seealsoie`, `seeie`, `tertiaryie`.

See Also

`indexterm`, `primary`, `primaryie`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `tertiary`, `tertiaryie`.

Examples

For examples, see `index`.

■ indexinfo

Name

indexinfo – Meta-information for an `Index`

Synopsis

Content Model

indexinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `IndexInfo` element is a wrapper for a large collection of meta-information about a `Index`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain indexinfo: `index`.

Children

The following elements occur in indexinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

`appendixinfo`, `articleinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `objectinfo`, `prefaceinfo`, `refsynopsisdivinfo`, `screeninfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setinfo`.

■ indexterm

Name

indexterm – A wrapper for terms to be indexed

Synopsis

Content Model

indexterm ::= (`primary`?, ((`secondary`, ((`tertiary`, (`see`|`seealso`+)?)| `see`|`seealso`+)?)| `see`|`seealso`+)?)

Attributes Common attributes

Name Type Default

significance normal | preferred "normal"

zone IDREFS *None*

pagenum CDATA *None*

class endofrange | singular | startofrange *None*

scope all | global | local *None*

startref IDREF *None*

Parameter Entities

%admon.mix; %bibliocomponent.mix; %bookcomponent.content;
%component.mix; %cptr.char.mix; %divcomponent.mix;
%docinfo.char.mix; %example.mix; %figure.mix;
%glossdef.mix; %highlights.mix; %info.class;
%legalnotice.mix; %listpreamble.mix; %ndxterm.class;
%para.char.mix; %qandaset.mix; %refcomponent.mix;
%refinline.char.mix; %revdescription.mix; %sidebar.mix;
%smallcptr.char.mix; %tbl.entry.mdl; %tbl.table.mdl;
%title.char.mix; %word.char.mix;

Description

IndexTerms identify text that is to be placed in the index. In the simplest case, the placement of the IndexTerm in the document identifies the location of the term in the text. In other words, the IndexTerm is placed in the flow of the document at the point where the IndexEntry in the Index should point. In other cases, attributes on IndexTerm are used to identify the location of the term in the text.

IndexTerms mark either a single point in the document or a range. A single point is marked with an IndexTerm placed in the text at the point of reference. There are two ways to identify a range of text:

- Place an IndexTerm at the beginning of the range with Class set to StartOfRange and give this term an ID. Place another IndexTerm at the end of the range with StartRef pointing to the ID of the starting IndexTerm. This second IndexTerm must be empty.

  The advantage of this method is that the range can span unbalanced element boundaries.

- Place the IndexTerm anywhere you like and point to the element that contains the range of text you wish to index with the Zone attribute on the IndexTerm. Note that Zone is defined as IDREFS so a single IndexTerm can point to multiple ranges.

  The advantage of this method is that IndexTerms can be collected together or even stored totally outside the flow of the document (in the meta for example).

Processing expectations

IndexTerms are suppressed in the primary text flow, although they contribute to the population of an index and serve as anchors for cross references. Under no circumstances is the actual content of IndexTerm rendered in the primary flow.

It is possible to construct index terms that are difficult to parse at best and totally illogical at worst. Consider the following:

```
<indexterm class='startofrange' zone="id1 id2">...</indexterm>
```

There is no way that this can fit into the semantics of an IndexTerm. Although it claims to be the start of a range, it does not have an ID for the end-of-range IndexTerm to point back to. In addition, it includes zoned terms, and mixing the two different methods for indicating a range in the same IndexTerm is probably a bad idea.

Parents

These elements contain indexterm: abbrev, accel, ackno, acronym, action, answer, appendix, appendixinfo, application, article, articleinfo, artpagenums, attribution, authorinitials, bibliocoverage, bibliodiv, biblioentry, bibliography, bibliographyinfo, biblioid, bibliomisc, bibliomixed, bibliomset, bibliorelation, biblioset, bibliosource, blockinfo, blockquote, bookinfo, bridgehead, callout, caution, chapter, chapterinfo, citation,

citebiblioid, citetitle, city, classname, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, constant, constraintdef, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, dedication, edition, email, emphasis, entry, envar, errorcode, errorname, errortext, errortype, example, exceptionname, fax, figure, filename, firstname, firstterm, foreignphrase, formalpara, funcparams, funcsynopsisinfo, function, glossary, glossaryinfo, glossdef, glossdiv, glossentry, glosssee, glossseealso, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, highlights, holder, honorific, important, index, indexinfo, informalexample, informalfigure, initializer, interface, interfacename, invpartnumber, isbn, issn, issuenum, itemizedlist, itermset, jobtitle, keycap, keycode, keysym, label, legalnotice, lineage, lineannotation, link, listitem, literal, literallayout, lotentry, manvolnum, markup, medialabel, member, methodname, modespec, modifier, mousebutton, msgaud, msgexplan, msglevel, msgorig, msgtext, note, objectinfo, olink, option, optional, orderedlist, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, partinfo, partintro, phone, phrase, pob, postcode, preface, prefaceinfo, procedure, productname, productnumber, programlisting, prompt, property, pubdate, publishername, pubsnumber, qandadiv, qandaset, question, quote, refentry, refentryinfo, refentrytitle, referenceinfo, refmeta, refmiscinfo, refpurpose, refsect1, refsect1info, refsect2, refsect2info, refsect3, refsect3info, refsection, refsectioninfo, refsynopsisdiv, refsynopsisdivinfo, releaseinfo, remark, returnvalue, revdescription, revnumber, revremark, screen, screeninfo, sect1, sect1info, sect2, sect2info, sect3, sect3info, sect4, sect4info, sect5, sect5info, section, sectioninfo, seg, segtitle, seriesvolnums, setindex, setindexinfo, setinfo, sgmltag, shortaffil, sidebar, sidebarinfo, simpara, simplesect, state, step, street, structfield, structname, subtitle, surname, symbol, synopsis, systemitem, table, term, tip, title, titleabbrev, tocback, tocentry, tocfront, token, type, ulink, userinput, variablelist, varname, volumenum, warning, wordasword, year.

Children

The following elements occur in indexterm: primary, secondary, see, seealso, tertiary.

Attributes

class Class identifies the type of IndexTerm. If StartRef is supplied, the default for Class is EndOfRange, otherwise it is Singular.

pagenum PageNum indicates the page on which this index term occurs in some version of the printed document.

scope Scope identifies in which indexes the IndexTerm should appear. Global means the index for the whole collection of documents, Local means the index for this document only, and All means both indexes.

significance Significance specifies whether or not this IndexTerm is considered the most important location for information about the terms being indexed. Generally, Preferred IndexTerms get special typographic treatment in the Index.

startref The use of StartRef implies a spanning index entry. StartRef is used on the term that defines the end of the span and points to the term which defines the beginning.

zone The use of Zone implies a spanning index entry. Zone holds the IDs of the elements to which it applies. The IndexTerm applies to the contents of the entire element(s) to which it points. If Zone is used, the physical placement of the IndexTerm in the flow of the document *is irrelevant*.

See Also

indexentry, primary, primaryie, secondary, secondaryie, see, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The Tiger<indexterm>
<primary>Big Cats</primary>
<secondary>Tigers</secondary></indexterm>
is a very large cat indeed.
</para>
```

The Tiger is a very large cat indeed.

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Example Chapter</title>

<!-- index term for "Example Chapter" is a span -->
<indexterm id="idxexchap" class='startofrange'>
  <primary>Example Chapter</primary></indexterm>

<!-- index term for "Example Chapter" also cross references the
     "Examples" entry in the index -->
<indexterm><primary>Example Chapter</primary>
  <seealso>Examples</seealso></indexterm>

<!-- index term for "Chapter, Example" refers the reader to the entry
     under which the index term is actually listed, "Example Chapter" -->
<indexterm><primary>Chapter</primary><secondary>Example</secondary>
  <see>Example Chapter</see></indexterm>

<!-- other content -->

<!-- index term, end of "Example Chapter" span -->
<indexterm startref="idxexchap" class="endofrange"/>

<para>some content</para>
</chapter>

<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Example Chapter</title>
<indexterm zone="a1"><primary>Network Configuration</primary></indexterm>
<!-- other content here -->
<sect1 id="a1"><title>Configuring Your Network</title>
<para>&hellip;</para>
</sect1>
</chapter>
```

For additional examples, see also chapter.

■ informalequation

Name

informalequation – A displayed mathematical equation without a title

Synopsis

Content Model

informalequation ::= (blockinfo?, (alt?, (graphic+|mediaobject+)))

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %figure.mix;
%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;
%informal.class; %listpreamble.mix; %para.mix;
%qandaset.mix; %refcomponent.mix; %revdescription.mix;
%sidebar.mix;

Description

An InformalEquation is usually a mathematical equation or a group of related mathematical equations.

Processing expectations

Formatted as a displayed block.

> NOTE
>
> 
>
> It is an error to supply a title in the blockinfo for an informal example.

Parents

These elements contain informalequation: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, equation, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in informalequation: alt, blockinfo, graphic, mediaobject.

See Also

equation, example, figure, informalexample, informalfigure, informaltable, inlineequation, subscript, superscript, table.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The equation
<informalequation>
  <alt>e^(pi*i) + 1 = 0</alt>
  <graphic fileref="figures/epi10.png"/>
```

```
</informalequation>
is delightful because it joins together five of the most
important mathematical constants.
</para>
```

The equation

$$e^{(}pi*i)+1=0$$

is delightful because it joins together five of the most important mathematical constants. For additional examples, see also `mml-math`.

■ informalexample

Name

informalexample – A displayed example without a title

Synopsis

Content Model

informalexample ::= (`blockinfo`?, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`| `segmentedlist`| `simplelist`|`variablelist`|`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`|`indexterm`| `beginpage`)+)

Attributes Common attributes

Name Type Default

width CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;` `%divcomponent.mix;` `%example.mix;` `%figure.mix;` `%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;` `%informal.class;` `%listpreamble.mix;` `%para.mix;` `%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`

Description

`InformalExample` is a wrapper for an example without a title. Examples often contain `ProgramListing`s or other large block elements.

Processing expectations

Formatted as a displayed block.

> NOTE
>
> 
>
> It is an error to supply a `title` in the `blockinfo` for an informal example.

Parents

These elements contain informalexample: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in informalexample: address, beginpage, blockinfo, blockquote, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, fieldsynopsis, formalpara, funcsynopsis, glosslist, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, programlisting, programlistingco, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, variablelist.

Attributes

width Width specifies the width (in characters) of the longest line in this InformalExample (formatters may use this value to determine scaling or rotation).

See Also

equation, example, figure, informalequation, informalfigure, informaltable, table.

Examples

```
<!DOCTYPE informalexample PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<informalexample>
<programlisting>
sub print_content_model {
    my($self) = shift;
    local($_) = shift;
    local(*FILE) = shift;

    my(@cm) = $self->format_content_model2($_);
    foreach $_ (@cm) {
        print FILE $self->make_links($_, 1, 1), "\n";
    }
}
</programlisting>
</informalexample>
```

```
sub print_content_model {
    my($self) = shift;
    local($_) = shift;
    local(*FILE) = shift;

    my(@cm) = $self->format_content_model2($_);
    foreach $_ (@cm) {
        print FILE $self->make_links($_, 1, 1), "\n";
    }
```

```
}
```

For additional examples, see also `screenco`.

■ informalfigure

Name

informalfigure – A untitled figure

Synopsis

Content Model

informalfigure ::= (`blockinfo`?, (`literallayout`|`programlisting`|`programlistingco`|`screen`|`screenco`|`screenshot`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`address`|`blockquote`|`graphic`|`graphicco`|`mediaobject`|`mediaobjectco`|`informalequation`|`informalexample`|`informalfigure`|`informaltable`|`indexterm`|`beginpage`|`link`|`olink`|`ulink`)+)

Attributes Common attributes

Name Type Default

float CDATA "0"

pgwide CDATA *None*

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%example.mix;` `%figure.mix;`

`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`

`%informal.class;` `%listpreamble.mix;` `%para.mix;`

`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`

`%sidebar.mix;`

Description

An `InformalFigure` is a figure without a title. Figures often contain `Graphic`s, or other large display elements.

Processing expectations

Formatted as a displayed block.

> NOTE
>
> 
>
> It is an error to supply a `title` in the `blockinfo` for an informal figure.

Parents

These elements contain informalfigure: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in informalfigure: address, beginpage, blockinfo, blockquote, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, fieldsynopsis, funcsynopsis, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, link, literallayout, mediaobject, mediaobjectco, methodsynopsis, olink, programlisting, programlistingco, screen, screenco, screenshot, synopsis, ulink.

Attributes

float If Float has the value 1 (true), then the processing system is free to move the figure to a convenient location. (Where convenient location may be described in the style sheet or may be application dependent.) A value of 0 (false) indicates that the figure should be placed precisely where it occurs in the flow.

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

pgwide If Pgwide has the value 0 (false), then the InformalFigure is rendered in the current text flow (with flow column width). A value of 1 (true) specifies that the figure should be rendered across the full text page.

See Also

equation, example, figure, informalequation, informalexample, informaltable, table.

Examples

```
<!DOCTYPE informalfigure PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<informalfigure>
<mediaobject>
<imageobject>
<imagedata fileref="watarun.eps" srccredit="Norman Walsh, 1998"/>
</imageobject>
<textobject><phrase>Wat Arun</phrase></textobject>
<caption><para>Wat Arun, Temple of the Dawn, on the Chao Phraya River
in Bangkok,
Thailand. In April, 1998, Wat Arun was in the midst of renovation.</para>
</caption>
</mediaobject>
</informalfigure>
```


Wat Arun, Temple of the Dawn, on the Chao Phraya River in Bangkok, Thailand. In April, 1998, Wat Arun was in the midst of renovation.

■ informaltable

Name

informaltable – A table without a title

Synopsis

Content Model

informaltable ::= (`blockinfo`?,`textobject`*, (`graphic`+|`mediaobject`+|`tgroup`+))

Attributes Common attributes

Name Type Default

tocentry CDATA *None*

shortentry CDATA *None*

tabstyle CDATA *None*

orient land | port *None*

label CDATA *None*

colsep CDATA *None*

frame all | bottom | none | sides | top | topbot *None*

pgwide CDATA *None*

rowsep CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;` `%divcomponent.mix;` `%example.mix;` `%figure.mix;` `%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;` `%informal.class;` `%listpreamble.mix;` `%para.mix;` `%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`

Description

An `InformalTable` element identifies an informal table (one without a `Title`). DocBook uses the CALS table model, which describes tables geometrically using rows, columns, and cells.

Tables may include column headers and footers, but there is no provision for row headers.

Processing expectations

Formatted as a displayed block.

NOTE

It is an error to supply a `title` in the `blockinfo` for an informal table.

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* http://www.oasis-open.org/html/a502.htm.

Future Changes

The OASIS Exchange Table Model http://www.oasis-open.org/specs/tm9901.html will replace the full OASIS Table Model.

Parents

These elements contain informaltable: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in informaltable: blockinfo, graphic, mediaobject, textobject, tgroup.

Attributes

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of all columns in this table. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute, not the ColSep.

frame Frame specifies how the table is to be framed:

Value	Meaning
all	Frame all four sides of the table. In some environments with limited control over table border formatting, such as HTML,
bottom	Frame only the bottom of the table.
none	Place no border on the table. In some environments with limited control over table border formatting, such as HTML, this
sides	Frame the left and right sides of the table.
top	Frame the top of the table.
topbot	Frame the top and bottom of the table.

There is no way to obtain a border on only the starting edge (left, in left-to-right writing systems) of the table.

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

orient Orient specifies the orientation of the InformalTable. An orientation of Port is the "upright", the same orientation as the rest of the text flow. An orientation of Land is 90 degrees counterclockwise from the upright orientation.

pgwide If Pgwide has the value 0 (false), then the InformalTable is rendered in the current text flow (with flow column width). A value of 1 (true) specifies that the table should be rendered across the full text page.

rowsep If RowSep has the value 1 (true), then a rule will be drawn below all the rows in the `InformalTable` (unless other, interior elements, suppress some or all of the rules). A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute and the RowSep of the last row is ignored.

shortentry This attribute is meaningless on `InformalTable`..

tabstyle TabStyle holds the name of a table style defined in a stylesheet (e.g., a FOSI) that will be used to process this document.

tocentry This attribute is meaningless on `InformalTable`.

See Also

`colspec`, `entry`, `entrytbl`, `equation`, `example`, `figure`, `informalequation`, `informalexample`, `informalfigure`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

```
<!DOCTYPE informaltable PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<informaltable frame='none'>
<tgroup cols='2'>
<colspec colwidth='0.5in'/>
<colspec colwidth='0.5in'/>
<tbody>
<row><entry>1</entry><entry>1</entry></row>
<row><entry>2</entry><entry>4</entry></row>
<row><entry>3</entry><entry>9</entry></row>
</tbody>
</tgroup>
</informaltable>
```

1	1
2	4
3	9

For additional examples, see also `entrytbl`, `footnoteref`.

■ initializer

Name

initializer – The initializer for a `FieldSynopsis`

Synopsis

Mixed Content Model

initializer ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

An `Initializer` identifies the initial or default value for a field (`FieldSynopsis`) or method parameter (`MethodParam`).

Processing expectations

Formatted inline. May be suppressed in some contexts.

Parents

These elements contain initializer: `fieldsynopsis`, `methodparam`.

Children

The following elements occur in initializer: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

■ inlineequation

Name

inlineequation – A mathematical equation or expression occurring inline

Synopsis

Content Model

inlineequation ::= ((`alt`?, (`graphic`+|`inlinemediaobject`+)))

Attributes Common attributes

Parameter Entities

`%inlineobj.char.class;` `%para.char.mix;` `%tbl.entry.mdl;`
`%title.char.mix;`

Description

`InlineEquation`s are expressions (usually mathematical) that occur in the text flow.

Processing expectations

Formatted inline.
`InlineEquation` should not contain `Graphic`. Instead, it should contain `InlineGraphic`. Within an `InlineEquation`, `Graphic` should be rendered inline.

Future Changes

In DocBook V5.0, `InlineGraphic` and `Graphic` will be discarded.

Parents

These elements contain inlineequation: `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `para`, `phrase`, `productname`, `programlisting`, `quote`, `refentrytitle`, `remark`, `screen`, `screeninfo`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `term`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in inlineequation: `alt`, `graphic`, `inlinemediaobject`.

See Also

`equation`, `informalequation`, `subscript`, `superscript`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Einstein's theory of relativity includes one of the most
widely recognized formulas in the world:
<inlineequation>
  <alt>e=mc^2</alt>
  <graphic fileref="figures/emc2.png"/>
</inlineequation>
</para>
```

Einstein's theory of relativity includes one of the most widely recognized formulas in the world: $e = mc^2$ For additional examples, see also `inlinemediaobject`.

■ inlinegraphic

Name

inlinegraphic – An object containing or pointing to graphical data that will be rendered inline

Synopsis

Content Model

inlinegraphic ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

valign bottom / middle / top *None*

width CDATA *None*

contentwidth CDATA *None*

BMP
CGM-BINARY
CGM-CHAR
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
IGES
format JPEG
JPG
linespecific
PCX
PIC
PNG
PS
SGML
SVG
TBL
TEX
TIFF
WMF
WPG
None

entityref ENTITY *None*

fileref CDATA *None*

scalefit CDATA *None*

depth CDATA *None*

scale CDATA *None*

contentdepth CDATA *None*

center
align left
right
None

Parameter Entities

`%cptr.char.mix;` `%docinfo.char.mix;` `%inlineobj.char.class;`

`%ndxterm.char.mix;` `%para.char.mix;` `%smallcptr.char.mix;`

`%tbl.entry.mdl;` `%title.char.mix;` `%word.char.mix;`

Description

This element contains graphical data, or a pointer to an external entity containing graphical data. One of the deficiencies of the DocBook `Graphic` element is that there is no way to specify an alternate text description of the graphic. This has been rectified by the introduction of `InlineMediaObject`.

Processing expectations

Formatted inline.

There are several ways to provide content for a `Graphic`. It is best to use only one of these methods. However, if multiple graphic sources are provided, the processing expectations are as follows: element content should be used in favor of either EntityRef or FileRef and EntityRef should be used in favor of FileRef.

Future Changes

In DocBook V4.0, `InlineGraphic` will be declared `EMPTY`. This change will require that any embedded graphic content be stored outside the SGML source and pointed to with an EntityRef or FileRef attribute.

Future Changes

In DocBook V5.0, `InlineGraphic` will be discarded. At that time, graphics will have to be incorporated using `MediaObject` or `InlineMediaObject`.

Parents

These elements contain inlinegraphic: `abbrev`, `accel`, `ackno`, `acronym`, `action`, `application`, `artpagenums`, `attribution`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `bibliosource`, `bridgehead`, `citation`, `citebiblioid`, `citetitle`, `city`, `classname`, `classsynopsisinf` `collabname`, `command`, `computeroutput`, `confdates`, `confnum`, `confsponsor`, `conftitle`, `constant`, `contractnum`, `contractsponsor`, `contrib`, `corpauthor`, `corpname`, `country`, `database`, `date`, `edition`, `email`, `emphasis`, `entry`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fax`, `filename`, `firstname`, `firstterm`, `foreignphrase`, `funcparams`, `funcsynopsisinf` `function`, `glosssee`, `glossseealso`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `holder`, `honorific`, `initializer`, `interface`, `interfacename`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `jobtitle`, `keycap`, `keycode`, `keysym`, `label`, `lineage`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `manvolnum`, `markup`, `medialabel`, `member`, `methodname`, `modespec`, `modifier`, `mousebutton`, `msgaud`, `msglevel`, `msgorig`, `olink`, `option`, `optional`, `orgdiv`, `orgname`, `otheraddr`, `othername`, `pagenums`, `para`, `parameter`, `phone`, `phrase`, `pob`, `postcode`, `primary`, `primaryie`, `productname`, `productnumber`, `programlisting`, `prompt`, `property`, `pubdate`, `publishername`, `pubsnumber`, `quote`, `refentrytitle`, `refmiscinfo`, `releaseinfo`, `remark`, `replaceable`, `returnvalue`, `revnumber`, `revremark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `seriesvolnums`, `sgmltag`, `shortaffil`, `simpara`, `state`, `street`, `structfield`, `structname`, `subscript`, `subtitle`, `superscript`, `surname`, `symbol`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `volumenum`, `wordasword`, `year`.

Attributes

align Align specifies the horizontal alignment of the graphic within the element that frames it.

depth Depth specifies the desired depth (vertical distance, at least in horizontal writing systems) of the image.

entityref EntityRef identifies the general entity which contains (or points to) the content of the graphic.

fileref FileRef specifies the name of the file which contains the content of the graphic.

format Format identifies the format of the graphic content. The Format must be a defined notation.

scale Scale specifies integer representing a percentage scaling factor (retaining the relative dimensions of the original graphic). If unspecified, the value 100 (100%) is assumed.

scalefit If ScaleFit has the value 1 (true), then the graphic is to be scaled (uniformly) to the specified width or depth. The default value of 0 (false) indicates that the image will not be scaled to fit (although it may still be scaled by the Scale attribute).

srccredit SrcCredit contains details about the source of the `InlineGraphic`.

width Width indicates the width of the graphic.

See Also

`alt`, `audioobject`, `caption`, `graphic`, `imageobject`, `inlinemediaobject`, `mediaobject`, `textobject`, `videoobject`.

Examples

For examples, see `guibutton`, `guiicon`, `guilabel`.

■ inlinemediaobject

Name

inlinemediaobject – An inline media object (video, audio, image, and so on)

Synopsis

Content Model

inlinemediaobject ::= (`objectinfo`?, (`videoobject`|`audioobject`|`imageobject`|`textobject`)+)

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%docinfo.char.mix;` `%inlineequation.content;`
`%inlineobj.char.class;` `%ndxterm.char.mix;` `%para.char.mix;`
`%smallcptr.char.mix;` `%tbl.entry.mdl;` `%title.char.mix;`
`%word.char.mix;`

Description

`InlineMediaObject` contains a set of alternative "graphical objects." In DocBook V3.1, three types of external graphical objects are defined: `VideoObject`s, `AudioObject`s, and `ImageObject`s. Additional textual descriptions may be provided with `TextObject`s.

Processing expectations

Formatted inline.

The primary purpose of the `InlineMediaObject` is to provide a wrapper around a set of alternative presentations of the same information.

If possible, the processing system should use the content of the first object within the `InlineMediaObject`. If the first object cannot be used, the remaining objects should be considered in the order that they occur. A processor should use the first object that it can, although it is free to choose any of the remaining objects if the primary one cannot be used.

Under no circumstances should more than one object in an `InlineMediaObject` be used or presented at the same time.

For example, an `InlineMediaObject` might contain a high resolution image, a low resolution image, and a text description. For print publishing, the high resolution image is used; for online systems, either the high or low resolution image is used, possibly including the text description as an online alternative. In a text-only environment, the text description is used.

Future Changes

In DocBook V5.0, `InlineMediaObject` will replace `InlineGraphic`.

Parents

These elements contain inlinemediaobject: abbrev, accel, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelatio bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classname, classsynopsisinf collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, constant, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, envar, errorcode, errorname, errortext, errortype, exceptionname, fax, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinf function, glosssee, glossseealso, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, holder, honorific, initializer, inlineequation, interface, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, keycode, keysym, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, markup, medialabel, member, methodname, modespec, modifier, mousebutton, msgaud, msglevel, msgorig, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, prompt, property, pubdate, publishername, pubsnumber, quote, refentrytitle, refmiscinfo, releaseinfo, remark, replaceable, returnvalue, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, sgmltag, shortaffil, simpara, state, street, structfield, structname, subscript, subtitle, superscript, surname, symbol, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, token, trademark, type, ulink, userinput, varname, volumenum, wordasword, year.

Children

The following elements occur in inlinemediaobject: audioobject, imageobject, objectinfo, textobject, videoobject.

See Also

alt, audioobject, caption, graphic, imageobject, inlinegraphic, mediaobject, textobject, videoobject.

Examples

In the following example, the InlineEquation uses InlineMediaObject to provide to alternate renderings for the equation. One is a graphic, the other is a text representation.

For display purposes, both in print and in the online presentation, the graphic is selected. If you were generating content for some other medium, a text only browser or a cell phone, for example, only the text representation might be selected.

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Einstein's most famous equation,
<inlineequation>
<inlinemediaobject>
<imageobject>
<imagedata fileref="figures/emc2.png"/>
</imageobject>
<textobject>
<phrase>E=mc<superscript>2</superscript></phrase>
</textobject>
</inlinemediaobject>
</inlineequation>, expresses the relationship between matter
and energy.
</para>
```

Einstein's most famous equation, $e = mc^2$, expresses the relationship between matter and energy.

interface

Name

interface – An element of a GUI

Synopsis

Mixed Content Model

interface ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`accel`)*

Attributes Common attributes

Name Type Default

moreinfo none
refentry
"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

An `Interface` identifies some part of a graphical user interface. This element became obsolete in DocBook V3.0 with the introduction of `GUIButton`, `GUIIcon`, `GUILabel`, `GUIMenu`, `GUIMenuItem`, and `GUISubMenu`.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

In DocBook V4.0, the Class attribute will be dropped from `Interface`. Use one of the GUI* tags instead or subclass locally with the Role attribute.

The content model of `Interface` will also be constrained to `(#PCDATA | Replaceable | InlineGraphic)`.

Parents

These elements contain interface: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `menuchoice`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in interface: accel, beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Interface.

See Also

classname, property, structfield, structname, symbol, token, type.

■ interfacename

Name

interfacename – The name of an interface

Synopsis

Mixed Content Model

interfacename ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The InterfaceName element is used to identify the name of an interface. This is likely to occur only in documentation about object-oriented programming systems, languages, and architectures.

Processing expectations

Formatted inline.

Parents

These elements contain interfacename: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, oointerface, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso,

seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in interfacename: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

■ invpartnumber

Name

invpartnumber – An inventory part number

Synopsis

Mixed Content Model

invpartnumber ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

An InvPartNumber identifies a number (an “inventory part number”) in some organization-specific numbering scheme.

Processing expectations

Formatted inline. Sometimes suppressed.

DocBook does not control, or specify, the numbering scheme used by an InvPartNumber. It is likely that this number uniquely identifies the document within the organization that assigns the numbers.

Parents

These elements contain invpartnumber: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in invpartnumber: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

biblioid, isbn, issn, issuenum, productnumber, pubsnumber, seriesvolnums, volumenum.

■ isbn

Name

isbn – The International Standard Book Number of a document

Synopsis

Mixed Content Model

isbn ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

An ISBN is the International Standard Book Number of a document.

Processing expectations

Formatted inline. Sometimes suppressed.

Deprecated as of DocBook V4.2, see biblioid instead.

Parents

These elements contain isbn: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in isbn: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

biblioid, invpartnumber, issn, issuenum, productnumber, pubsnumber, seriesvolnums, volumenum.

Examples

For examples, see bibliography.

■ issn

Name

issn – The International Standard Serial Number of a periodical

Synopsis

Mixed Content Model

issn ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark| subscript| superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

An ISSN is the International Standard Serial Number of a periodical.

Processing expectations

Formatted inline. Sometimes suppressed.
Deprecated as of DocBook V4.2, see biblioid instead.

Parents

These elements contain issn: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in issn: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

biblioid, invpartnumber, isbn, issuenum, productnumber, pubsnumber, seriesvolnums, volumenum.

Examples

For examples, see bibliography, biblioset.

■ issuenum

Name

issuenum – The number of an issue of a journal

Synopsis

Mixed Content Model

issuenum ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark| subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The IssueNum contains the issue number of a periodical.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain issuenum: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in issuenum: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`biblioid`, `invpartnumber`, `isbn`, `issn`, `productnumber`, `pubsnumber`, `seriesvolnums`, `volumenum`.

Examples

For examples, see `article`, `bibliography`, `bibliomset`.

■ itemizedlist

Name

itemizedlist – A list in which each entry is marked with a bullet or other dingbat

Synopsis

Content Model

itemizedlist ::= (`blockinfo`?, (`title`,`titleabbrev`?)?, (`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`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)*, `listitem`+)

Attributes Common attributes

Name Type Default

mark CDATA *None*

spacing compact | normal *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;` `%divcomponent.mix;` `%example.mix;` `%footnote.mix;` `%glossdef.mix;` `%highlights.mix;` `%indexdivcomponent.mix;` `%legalnotice.mix;` `%list.class;` `%para.mix;` `%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;` `%tabentry.mix;` `%tbl.entry.mdl;` `%textobject.mix;`

Description

In an `ItemizedList`, each member of the list is marked with a bullet, dash, or other symbol.

Processing expectations

Formatted as a displayed block.

DocBook specifies neither the initial mark nor the sequence of marks to be used in nested lists. If explicit control is desired, the Mark attribute should be used. The values of the Mark attribute are expected to be keywords, not representations (numerical character references, entities, and so on.) of the actual mark.

In order to enforce a standard set of marks at your organization, it may be useful to construct a customization layer that limits the values of the Mark attribute to an enumerated list. See Chapter 5, *Customizing DocBook*.

Future Changes

Introductory material may appear before the first list item.

Parents

These elements contain itemizedlist: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `example`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `highlights`, `important`, `index`, `indexdiv`, `informalexample`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `warning`.

Children

The following elements occur in itemizedlist: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockinfo`, `blockquote`, `bridgehead`, `caution`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `fieldsynopsis`, `formalpara`, `funcsynopsis`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `listitem`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `note`, `para`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `simpara`, `synopsis`, `tip`, `title`, `titleabbrev`, `warning`.

Attributes

mark Mark contains a keyword indicating the type of mark to be used on items in this `ItemizedList`. DocBook does not provide a fixed list of appropriate keywords.

spacing Spacing indicates whether or not the vertical space in the list should be minimized.

See Also

`calloutlist`, `listitem`, `orderedlist`, `segmentedlist`, `simplelist`, `variablelist`.

Examples

```
<!DOCTYPE itemizedlist PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<itemizedlist mark='opencircle'>
<listitem>
<para>
TeX and LaTeX
</para>
</listitem>
```

```
<listitem override='bullet'>
<para>
Troff
</para>
</listitem>
<listitem>
<para>
Lout
</para>
</listitem>
</itemizedlist>
```

- TeX and LaTeX
- Troff
- Lout

For additional examples, see also `highlights`, `para`, `xref`.

■ itermset

Name

itermset – A set of index terms in the meta-information of a document

Synopsis

Content Model

itermset ::= (`indexterm`+)

Attributes Common attributes

Parameter Entities

`%info.class;`

Description

When `IndexTerm`s use the Zone attribute to point to index ranges, it may be handy to hoist them out of the flow and put them in the document meta-information.

The `ITermSet` element, which occurs in the DocBook containers for meta-information, is one place to put them. `ITermSet` is simply a wrapper around a group of `IndexTerm`s.

Processing expectations

Suppressed.

Although more than one `ITermSet` may appear in the meta-information for a document, neither a relationship nor a specific facility for constructing a relationship is defined.

Parents

These elements contain itermset: `appendixinfo`, `articleinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in itermset: `indexterm`.

Examples

For examples, see `chapter`.

■ jobtitle

Name

jobtitle – The title of an individual in an organization

Synopsis

Mixed Content Model

jobtitle ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

A `JobTitle` describes the position of an individual within an organization. This tag is generally reserved for the name of the title for which an individual is paid.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain jobtitle: `affiliation`.

Children

The following elements occur in jobtitle: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `corpname`, `orgdiv`, `orgname`, `shortaffil`.

Examples

For examples, see `author`, `authorgroup`.

■ keycap

Name

keycap – The text printed on a key on a keyboard

Synopsis

Mixed Content Model

keycap ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|

sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none
refentry
"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;
%tech.char.class; %title.char.mix;

Description

The KeyCap identifies the text printed on a physical key on a computer keyboard. This is distinct from any scan code that it may generate (KeyCode), or any symbolic name (KeySym) that might exist for the key.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

In DocBook V4.0, the content model of KeyCap will be constrained to (#PCDATA | Replaceable | InlineGraphic).

Parents

These elements contain keycap: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, keycombo, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, shortcut, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in keycap: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Keycap.

See Also

accel, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, keycode, keycombo, keysym, menuchoice, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <keycap>F1</keycap> key on an IBM PC keyboard generates the
scan code <keycode>0x3B</keycode> when pressed.  This value
is defined as <keysym>KEY_F1</keysym> in
<filename class="headerfile">keyboard.h</filename>.
</para>
```

The **F1** key on an IBM PC keyboard generates the scan code 0x3B when pressed. This value is defined as **KEY_F1** in keyboard.h. For additional examples, see also keycode, keycombo, keysym.

■ keycode

Name

keycode – The internal, frequently numeric, identifier for a key on a keyboard

Synopsis

Mixed Content Model

keycode ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The KeyCode identifies the numeric value associated with a key on a computer keyboard. This is distinct from any scan code that it may generate (KeyCode), or any symbolic name (KeySym) that might exist for the key.

Processing expectations

Formatted inline.

Parents

These elements contain keycode: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in keycode: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `keycap`, `keycombo`, `keysym`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <keycap>F1</keycap> key on an IBM PC keyboard generates the
scan code <keycode>0x3B</keycode> when pressed.  This value
is defined as <keysym>KEY_F1</keysym> in
<filename class="headerfile">keyboard.h</filename>.
</para>
```

The **F1** key on an IBM PC keyboard generates the scan code 0x3B when pressed. This value is defined as **KEY_F1** in `keyboard.h`. For additional examples, see also `keycap`, `keysym`.

■ keycombo

Name

keycombo – A combination of input actions

Synopsis

Content Model

keycombo ::= ((`keycap`|`keycombo`|`keysym`|`mousebutton`)+)

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

otheraction CDATA *None*

action click | double-click | other | press | seq | simul

None

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

For actions that require multiple keystrokes, mouse actions, or other physical input selections, the `KeyCombo` element provides a wrapper for the entire set of events.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain keycombo: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `keycombo`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `shortcut`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in keycombo: `keycap`, `keycombo`, `keysym`, `mousebutton`.

Attributes

action Action identifies the nature of the action taken. If `KeyCombo` contains more than one action element, `Simul` is the default value for Action, otherwise there is no default.

If `Other` is specified, OtherAction should be used to identify the nature of the action.

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Keycombo`.

otheraction OtherAction should be used when Action is set to `Other`. It identifes the nature of the action in some application-specific way.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `keycap`, `keycode`, `keysym`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
To move a highlighted region, use
<keycombo action='simul'>
  <keycap>Shift</keycap>
  <mousebutton>Button1</mousebutton>
</keycombo>
and drag the text to the new location.
</para>
```

To move a highlighted region, use **Shift**+Button1 and drag the text to the new location. For additional examples, see also `guimenu`, `guimenuitem`, `menuchoice`, `shortcut`.

■ keysym

Name

keysym – The symbolic name of a key on a keyboard

Synopsis

Mixed Content Model

keysym ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

The `KeySym` identifies the symbolic name of a key on a computer keyboard. This is distinct from any scan code that it may generate (`KeyCode`), or any symbolic name (`KeySym`) that might exist for the key.

Processing expectations

Formatted inline.

Parents

These elements contain keysym: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `keycombo`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `shortcut`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in keysym: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `keycap`, `keycode`, `keycombo`, `menuchoice`, `mousebutton`, `shortcut`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <keycap>F1</keycap> key on an IBM PC keyboard generates the
scan code <keycode>0x3B</keycode> when pressed.  This value
is defined as <keysym>KEY_F1</keysym> in
<filename class="headerfile">keyboard.h</filename>.
</para>
```

The **F1** key on an IBM PC keyboard generates the scan code 0x3B when pressed. This value is defined as **KEY_F1** in `keyboard.h`. For additional examples, see also `guimenu`, `guimenuitem`, `keycap`, `keycode`, `menuchoice`, `shortcut`.

■ keyword

Name

keyword – One of a set of keywords describing the content of a document

Synopsis

Mixed Content Model

keyword ::= (#PCDATA)

Attributes Common attributes

Description

A `Keyword` is a term describing the content of a document. The keyword applies to the document component that contains it.

Processing expectations

Keywords are rarely displayed to a reader. Usually, they are reserved for searching and retrieval purposes. If they are displayed, they may be displayed either inline or as a displayed block, depending on context.

Unlike `SubjectTerm`s, which should be drawn from a controlled vocabulary, keywords may be chosen freely.

Parents

These elements contain keyword: `keywordset`.

See Also

`keywordset`, `subject`, `subjectset`, `subjectterm`.

Examples

For examples, see `chapter`.

■ keywordset

Name

keywordset – A set of keywords describing the content of a document

Synopsis

Content Model

keywordset ::= (`keyword`+)

Attributes Common attributes

Parameter Entities

`%info.class;`

Description

A set of keywords, provided by the author, editor, publisher, and so on, can be stored in the document meta-information in a `KeywordSet`.

Keywords can form an important part of an automated indexing or searching strategy for a collection of documents.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Keywords are rarely displayed to a reader. Usually, they are reserved for searching and retrieval purposes.

Unlike `SubjectTerm`s, which should be drawn from a controlled vocabulary, keywords may be chosen freely.

Although more than one `KeywordSet` may appear in the meta-information for a document, neither a relationship nor a specific facility for constructing a relationship is defined by DocBook.

Additionally, no relationship is defined between the `KeywordSet`s of a document component and the `KeywordSet`s of its parents or children.

Parents

These elements contain keywordset: `appendixinfo`, `articleinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in keywordset: `keyword`.

See Also

`keyword`, `subject`, `subjectset`, `subjectterm`.

Examples

For examples, see `chapter`.

■ label

Name

label – A label on a `Question` or `Answer`

Synopsis

Mixed Content Model

label ::= (#PCDATA|`acronym`|`emphasis`|`trademark`|`link`|`olink`|`ulink`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

The `Label` of a `Question` or `Answer` identifies the label that is to be used when formatting the question or answer.

Processing expectations

The Label element is used as the label for the Question or Answer. A processing application might, for example, format the label as a heading preceding the question or answer contents, or it might format it as a run-in heading in the first paragraph of the question or answer.

Parents

These elements contain label: answer, question.

Children

The following elements occur in label: acronym, anchor, beginpage, emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, subscript, superscript, trademark, ulink.

■ legalnotice

Name

legalnotice – A statement of legal obligations or requirements

Synopsis

Content Model

legalnotice ::= (blockinfo?,title?, (calloutlist|glosslist|itemizedlist|orderedlist| segmentedlist| simplelist|variablelist|caution|important|note|tip|warning| literallayout|programlisting|programlistingco|screen|screenco| screenshot| formalpara|para|simpara|blockquote|indexterm| beginpage)+)

Attributes Common attributes

Parameter Entities

```
%info.class;
```

Description

LegalNotice identifies a statement of legal obligation, requirement, or warranty. It occurs in the meta-information for a document in which it frequently explains copyright, trademark, and other legal formalities of a document.

Processing expectations

Formatted as a displayed block.

LegalNotices may be presented in a number of ways. In printed documents, they often occur on the verso of the title page, sometimes in a reduced font size. Online, they may occur on the title page or in a separate document behind a hypertext link.

Parents

These elements contain legalnotice: appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in legalnotice: beginpage, blockinfo, blockquote, calloutlist, caution, formalpara, glosslist, important, indexterm, itemizedlist, literallayout, note, orderedlist, para, programlisting, programlistingco, screen, screenco, screenshot, segmentedlist, simpara, simplelist, tip, title, variablelist, warning.

See Also

`copyright`, `trademark`.

Examples

For examples, see `book`, `bookinfo`, `sect1info`.

■ lhs

Name

lhs – The left-hand side of an EBNF production

Synopsis

Mixed Content Model

lhs ::= (#PCDATA)

Attributes Common attributes

Description

This element is only available if you are using the EBNF Module.

The left-hand side (`LHS`) of a production is a non-terminal defined in terms of the right-hand side (`RHS`) of the `Production`.

Parents

These elements contain lhs: `production`.

Examples

For examples, see `productionset`.

■ lineage

Name

lineage – The portion of a person's name indicating a relationship to ancestors

Synopsis

Mixed Content Model

lineage ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;` `%person.ident.mix;`

Description

`Lineage` is a portion of a person's name, typically "Jr." or "Sr."

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain lineage: address, appendixinfo, articleinfo, author, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, editor, glossaryinfo, indexinfo, objectinfo, othercredit, partinfo, personname, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in lineage: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

affiliation, firstname, honorific, othername, surname.

Examples

For examples, see authorgroup.

■ lineannotation

Name

lineannotation – A comment on a line in a verbatim listing

Synopsis

Mixed Content Model

lineannotation ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Description

A LineAnnotation is an author or editor's comment on a line in one of the verbatim environments. These are annotations added by the documentor, not part of the original listing.

Processing expectations

Formatted inline. In verbatim environments like ProgramListing, which are often presented in a fixed width font, they may get special typographic treatment, such as italics.

If several LineAnnotations occur in the same listing, they may be aligned horizontally.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain lineannotation: classsynopsisinfo, funcsynopsisinfo, literallayout, programlisting, rhs, screen, synopsis.

Children

The following elements occur in lineannotation: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

computeroutput, literallayout, programlisting, screen, screenshot, synopsis, userinput.

Examples

The following example, from the description of Entry, shows how LineAnnotation can be used to annotate a Screen listing:

```
<!DOCTYPE screen PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<screen>
&lt;entry>                <lineannotation>Error, cannot have a line break before a blo
&lt;para>
A paragraph of text.
&lt;/para>&lt;/entry>
</screen>
```

```
<entry>                Error, cannot have a line break before a block element
<para>
A paragraph of text.
</para></entry>
```

For additional examples, see also productionset.

■ link

Name

link – A hypertext link

Synopsis

Mixed Content Model

link ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

endterm IDREF *None*

type CDATA *None*

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %indexdivcomponent.mix;

%link.char.class; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;

%word.char.mix;

Description

Link is a general purpose hypertext element. Usually, Link surrounds the text that should be made "hot," (unlike XRef which must generate the text) but the EndTerm attribute can be used to copy text from another element.

Processing expectations

Formatted inline.

If the Link element has content, then that content is processed for output as the "hot" text. If the Link element has content and an EndTerm attribute, then the content is used and the EndTerm is ignored. If the Link element has an EndTerm attribute and no content, then the content of the element pointed to by EndTerm should be repeated at the location of the Link and used as the "hot" text.

Linking elements must not be nested within other linking elements (including themselves). Because DocBook is harmonizing towards XML, this restriction cannot easily be enforced by the DTD. The processing of nested linking elements is undefined.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain link: abbrev, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry,

fax, figure, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, holder, honorific, indexdiv, informalfigure, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, quote, refentry, refentrytitle, refmiscinfo, refnamediv, refpurpose, releaseinfo, remark, replaceable, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, shortaffil, simpara, state, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, volumenum, wordasword, year.

Children

The following elements occur in link: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

endterm Endterm points to the element whose content is to be used as the text of the link. If Endterm is supplied on a Link which has content, the value of Endterm should be ignored.

linkend Linkend points to the target of the link.

type Type is available for application-specific customization of the linking behavior.

See Also

anchor, olink, ulink, xref.

Examples

```
<!DOCTYPE sect1 PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<sect1><title>Examples of <sgmltag>Link</sgmltag></title>

<para>
In this sentence <link linkend='nextsect'>this</link> word is
hot and points to the following section.
</para>

<para>
```

```
There is also a link to the section called
<quote><link linkend='nextsect' endterm="nextsect.title"/></quote>
in this sentence.
</para>

<sect2 id='nextsect'><title id='nextsect.title'>A Subsection</title>

<para>
This section only exists to be the target of a couple of links.
</para>

</sect2>
</sect1>
```

■ listitem

Name

listitem – A wrapper for the elements of a list item

Synopsis

Content Model

listitem ::= ((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+)

Attributes Common attributes

Name Type Default

override CDATA *None*

Description

The `ListItem` element is a wrapper around an item in a list. In an `ItemizedList` or an `OrderedList`, the `ListItem` surrounds the entire list item. In a `VariableList`, `ListItem` surrounds the "definition" part of the list item.

Processing expectations

Formatted as a displayed block. List items usually generate the appropriate mark (a number or bullet) and appear indented, next to the mark. In a `VariableList`, the presentation may be influenced by the length of the `Term` or `Term`s that precede the list item and by attributes on the list itself.

Parents

These elements contain listitem: `itemizedlist`, `orderedlist`, `varlistentry`.

Children

The following elements occur in listitem: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`,

indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, tip, variablelist, warning.

Attributes

override Override specifies the keyword for the type of mark to be used on *this* ListItem instead of the mark currently in use for the list.

See Also

calloutlist, itemizedlist, orderedlist, segmentedlist, simplelist, variablelist.

Examples

For examples, see highlights, itemizedlist, orderedlist, para, variablelist, xref.

■ literal

Name

literal – Inline text that is some literal value

Synopsis

Mixed Content Model

literal ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

A Literal is some specific piece of data, taken literally, from a computer system. It is similar in some ways to UserInput and ComputerOutput, but is somewhat more of a general classification. The sorts of things that constitute literals varies by domain.

Processing expectations

Formatted inline. A literal is frequently distinguished typographically and `Literal` is often used wherever that typographic presentation is desired.

The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of `Literal` will be constrained to `(#PCDATA | Replaceable | InlineGraphic)` in DocBook V4.0.

Parents

These elements contain literal: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Literal`.

See Also

`command`, `computeroutput`, `constant`, `markup`, `option`, `optional`, `parameter`, `prompt`, `replaceable`, `sgmltag`, `userinput`, `varname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>There are several undocumented settings for <varname>debug</varname>,
among them <literal>3.27</literal> to enable a complete trace and
<literal>3.8</literal> to debug the spell checker. For a complete
list of the possible settings,
see <filename class="headerfile">edit/debug.h</filename>.</para>
```

There are several undocumented settings for `debug`, among them `3.27` to enable a complete trace and `3.8` to debug the spell checker. For a complete list of the possible settings, see `edit/debug.h`. For additional examples, see also `programlistingco`.

■ literallayout

Name

literallayout – A block of text in which line breaks and white space are to be reproduced faithfully

Synopsis

Mixed Content Model

literallayout ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage|co|coref|textobject|lineannotation)*

Attributes Common attributes

Name Type Default

width CDATA *None*

format linespecific "linespecific"

linenumbering numbered | unnumbered *None*

class monospaced | normal "normal"

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix; %divcomponent.mix; %example.mix; %figure.mix; %footnote.mix; %glossdef.mix; %indexdivcomponent.mix; %legalnotice.mix; %linespecific.class; %listpreamble.mix; %para.mix; %qandaset.mix; %refcomponent.mix; %revdescription.mix; %sidebar.mix; %tabentry.mix; %tbl.entry.mdl; %textobject.mix;

Description

LiteralLayout is a verbatim environment. Unlike the other verbatim environments, it does not have strong semantic overtones and may not imply a font change.

Processing expectations

This element is displayed “verbatim”; whitespace and linebreaks within this element are significant.

Unlike ProgramListing and Screen, which usually imply a font change, LiteralLayout does not. How spaces are to be represented faithfully in a proportional font is not addressed by DocBook.

In DocBook V3.1, the Class attribute was added to give users control over the font used in LiteralLayouts. If the Class attribute is specified and its value is Monospaced, then the LiteralLayout will be presented in

a monospaced font, probably the same one used for other verbatim environments. The default value for Class is `Normal`, meaning that no font change will occur.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Future Changes

The `xml:space` attribute is automatically provided in the XML DTD.

Parents

These elements contain literallayout: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `epigraph`, `example`, `figure`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `indexdiv`, `informalexample`, `informalfigure`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `variablelist`, `warning`.

Children

The following elements occur in literallayout: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `co`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `coref`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `lineannotation`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `textobject`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

class Class distinguishes between literal layout environments that are presented in a monospaced font and literal layout environments that have no implicit font change.

format The Format attribute applies the `linespecific` notation to all `LiteralLayout`s. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of a `LiteralLayout` are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

width Width specifies the width (in characters) of the longest line in this `LiteralLayout` (formatters may use this value to determine scaling or rotation).

See Also

computeroutput, lineannotation, programlisting, screen, screenshot, synopsis, userinput.

Examples

```
<!DOCTYPE blockquote PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<blockquote>
<attribution>William Shakespeare, <citetitle>Henry V</citetitle></attribution>
<literallayout>
  O, for a muse of fire, that would ascend
The brightest heaven of invention!
A kingdom for a stage, princes to act,
And monarchs to behold the swelling scene!
</literallayout>
</blockquote>
```

> O, for a muse of fire, that would ascend
> The brightest heaven of invention!
> A kingdom for a stage, princes to act,
> And monarchs to behold the swelling scene!
>
> —William Shakespeare, *Henry V*

For additional examples, see also attribution, part.

■ lot

Name

lot – A list of the titles of formal objects (as tables or figures) in a document

Synopsis

Content Model

lot ::= (beginpage?, (title,subtitle?,titleabbrev?)?, lotentry*)

Attributes Common attributes

Name Type Default

label CDATA *None*

Parameter Entities

%nav.class; %partcontent.mix;

Description

A LoT is a *list of titles*. It can be used to generate lists of Figures, Tables, Examples, or Equations.

Processing expectations

Formatted as a displayed block. A list of titles in a Book usually introduces a forced page break.

Most often, lists of titles are generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain lot: appendix, article, book, chapter, part, preface, sect1, sect2, sect3, sect4, sect5, section.

Children

The following elements occur in lot: `beginpage`, `lotentry`, `subtitle`, `title`, `titleabbrev`.

Attributes

label Label specifies an identifying number or string that may be used in presentation.

Examples

```
<!DOCTYPE lot PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<lot><title>List of Figures</title>
<lotentry pagenum='5'>The Letters &ldquo;g&rdquo; and &ldquo;h&rdquo;
   inside their boxes</lotentry>
<lotentry pagenum='15'>Example figure produced by both TeX and
   troff</lotentry>
<!-- ... -->
</lot>
```

■ lotentry

Name

lotentry – An entry in a list of titles

Synopsis

Mixed Content Model

lotentry ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

linkend IDREF *None*

pagenum CDATA *None*

Description

A `LoTentry` identifies an individual title in a `LoT`. For example, in a *list of figures*, each individual figure title would be repeated in a `LoTentry` in the `LoT` for the list of figures.

Processing expectations

Formatted as a displayed block.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain lotentry: `lot`.

Children

The following elements occur in lotentry: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

linkend Linkend points to the element which is represented in this `LoTEntry`.

pagenum PageNum indicates the page on which this entry occurs in some version of the printed document.

srccredit SrcCredit contains details about the source of the element referenced by this entry.

Examples

For examples, see `lot`.

■ manvolnum

Name

manvolnum – A reference volume number

Synopsis

Mixed Content Model

manvolnum ::= (#PCDATA|`acronym`|`emphasis`|`trademark`|`link`|`olink`|`ulink`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

In a DocBook reference page, the ManVolNum holds the number of the volume in which the RefEntry belongs.

The notion of a volume number is historical. UNIX manual pages ("man pages"), for which RefEntry was devised, were typically stored in three ring binders. Each bound manual was a volume in a set and contained information about a particular class of things. For example, volume 1 was for user commands, and volume 8 was for administrator commands.

Volume numbers need not be strictly numerical; volume "l" frequently held manual pages for local additions to the system, and the X Window System manual pages had an "x" in the volume number: for example, 1x.

Processing expectations

The content of ManVolNum is usually printed in parentheses after the element title or citation.

Parents

These elements contain manvolnum: citerefentry, refmeta.

Children

The following elements occur in manvolnum: acronym, anchor, beginpage, emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, subscript, superscript, trademark, ulink.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
For a further description of print formats, consult the
<citerefentry><refentrytitle>printf</refentrytitle>
<manvolnum>3S</manvolnum></citerefentry> manual page.
</para>
```

For a further description of print formats, consult the printf(3S) manual page. For additional examples, see also citerefentry, refentry.

■ markup

Name

markup – A string of formatting markup in text that is to be represented literally

Synopsis

Mixed Content Model

markup ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

Markup contains a string of formatting markup that is to be represented literally in the text. The utility of this element is almost wholly constrained to books about document formatting tools.

Processing expectations

Formatted inline.

Parents

These elements contain markup: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in markup: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

See Also

`computeroutput`, `constant`, `literal`, `option`, `optional`, `parameter`, `prompt`, `replaceable`, `sgmltag`, `userinput`, `varname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
A presentation system using TeX as a back end might allow you
to insert inline markup, such as <markup role="tex">$x^2$</markup>,
using TeX syntax directly.
</para>
```

A presentation system using TeX as a back end might allow you to insert inline markup, such as x^2, using TeX syntax directly.

■ medialabel

Name

medialabel – A name that identifies the physical medium on which some information resides

Synopsis

Mixed Content Model

medialabel ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

class	cartridge cdrom disk tape	*None*

Parameter Entities

```
%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;
%tech.char.class; %title.char.mix;
```

Description

The MediaLabel element identifies the name of a specific piece of physical media, such as a tape or disk label. Usually, a media label is something external, written by hand on the media itself, for example, but it may also refer to digital labels.

Processing expectations

Formatted inline.

Parents

These elements contain medialabel: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in medialabel: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

class Class indicates the type of media labeled.

See Also

application, database, filename, hardware, productname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
To install <application>The Great Foo</application>, insert the disk
labelled <medialabel>TGF Setup 1</medialabel> and run
<command>setup</command>.
</para>
```

To install The Great Foo, insert the disk labelled *TGF Setup 1* and run **setup**.

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The weekly incremental backup tape is labelled
<medialabel>Backup <replaceable>nn</replaceable></medialabel>, where
<replaceable>nn</replaceable> is the week number.
</para>
```

The weekly incremental backup tape is labelled *Backup nn*, where *nn* is the week number.

■ mediaobject

Name

mediaobject – A displayed media object (video, audio, image, etc.)

Synopsis

Content Model

mediaobject ::= (`objectinfo`?, (`videoobject`|`audioobject`|`imageobject`|`textobject`)+, `caption`?)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%equation.content;` `%example.mix;`
`%figure.mix;` `%footnote.mix;` `%glossdef.mix;`
`%indexdivcomponent.mix;` `%info.class;` `%informal.class;`
`%listpreamble.mix;` `%para.mix;` `%qandaset.mix;`
`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`
`%tabentry.mix;` `%tbl.entry.mdl;` `%tbl.table.mdl;`

Description

This element contains a set of alternative "media objects." In DocBook V3.1, three types of external objects are defined: `VideoObject`s, `AudioObject`s, and `ImageObject`s. Additional textual descriptions may be provided with `TextObject`s.

Processing expectations

Formatted as a displayed block.

The primary purpose of the `MediaObject` is to provide a wrapper around a set of alternative presentations of the same information.

If possible, the processing system should use the content of the first object within the `MediaObject`. If the first object cannot be used, the remaining objects should be considered in the order that they occur. A processor should use the first object that it can, although it is free to choose any of the remaining objects if the primary one cannot be used.

Under no circumstances should more than one object in a `MediaObject` be used or presented at the same time.

For example, a `MediaObject` might contain a video, a high resolution image, a low resolution image, a long text description, and a short text description. In a "high end" online system, the video is used. For print publishing, the high resolution image is used. For other online systems, either the high or low resolution image is used, possibly including the short text description as the online alternative. In a text-only environment, either the long or short text descriptions are used.

Future Changes

In DocBook V5.0, `MediaObject` will replace `Graphic`.

Parents

These elements contain mediaobject: `answer`, `appendix`, `appendixinfo`, `article`, `articleinfo`, `bibliodiv`, `bibliography`, `bibliographyinfo`, `blockinfo`, `blockquote`, `bookinfo`, `callout`, `caution`, `chapter`, `chapterinfo`, `constraintdef`, `entry`, `equation`, `example`, `figure`, `footnote`, `glossary`, `glossaryinfo`, `glossdef`, `glossdiv`, `important`, `index`, `indexdiv`, `indexinfo`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `listitem`, `msgexplan`, `msgtext`, `note`, `objectinfo`, `orderedlist`, `para`, `partinfo`, `partintro`, `preface`, `prefaceinfo`, `procedure`, `qandadiv`, `qandaset`, `question`, `refentryinfo`, `referenceinfo`,

refsect1, refsect1info, refsect2, refsect2info, refsect3, refsect3info, refsection, refsectioninfo, refsynopsisdiv, refsynopsisdivinfo, revdescription, screenshot, sect1, sect1info, sect2, sect2info, sect3, sect3info, sect4, sect4info, sect5, sect5info, section, sectioninfo, setindex, setindexinfo, setinfo, sidebar, sidebarinfo, simplesect, step, synopsis, table, tip, variablelist, warning.

Children

The following elements occur in mediaobject: audioobject, caption, imageobject, objectinfo, textobject, videoobject.

See Also

alt, audioobject, caption, graphic, imageobject, inlinegraphic, inlinemediaobject, textobject, videoobject.

Examples

For examples, see audioobject, imageobject, informalfigure, svg-svg, videoobject.

■ mediaobjectco

Name

mediaobjectco – A media object that contains callouts

Synopsis

Content Model

mediaobjectco ::= (objectinfo?,imageobjectco, (imageobjectco|textobject)*)

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %figure.mix;
%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;
%informal.class; %listpreamble.mix; %para.mix;
%qandaset.mix; %refcomponent.mix; %revdescription.mix;
%sidebar.mix;

Description

A MediaObjectCO is a wrapper around a set of alternative, annotated media objects.

Processing expectations

Formatted as a displayed block.
See also MediaObject.

Parents

These elements contain mediaobjectco: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, screenshot, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, variablelist, warning.

Children

The following elements occur in mediaobjectco: imageobjectco, objectinfo, textobject.

See Also

areaspec, calloutlist, co, coref, graphicco, imageobjectco, programlistingco, screenco.

Examples

```
<!DOCTYPE mediaobjectco PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<mediaobjectco>
<imageobjectco>
<areaspec units="calspair">
  <areaset id="oneway" coords="">
    <area id="oneway1" coords="300 400"/>
    <area id="oneway2" coords="325 340"/>
  </areaset>
  <area id="myhouse" coords="425 590"/>
</areaspec>
<imageobject>
<imagedata fileref="http://somemap.site.com/EARTH?USA?MA?01007"/>
</imageobject>
</imageobjectco>
</mediaobjectco>
```

[mediaobjectco] [imageobjectco] [/imageobjectco] [/mediaobjectco]

■ member

Name

member – An element of a simple list

Synopsis

Mixed Content Model

member ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis| cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Description

A Member is an element of a SimpleList. Unlike the other lists, items in a SimpleList are constrained to character data and inline elements.

Processing expectations

Formatted inline. How the inline `Member`s are formatted with respect to each other is controlled by the containing `SimpleList`.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain member: `simplelist`.

Children

The following elements occur in member: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Examples

For examples, see `simplelist`.

■ menuchoice

Name

menuchoice – A selection or series of selections from a menu

Synopsis

Content Model

menuchoice ::= (`shortcut`?, (`guibutton`|`guiicon`|`guilabel`|`guimenu`|`guimenuitem`|`guisubmenu`|`interface`)+)

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

In applications that present graphical user interfaces, it is often necessary to select an item, or a series of items, from a menu in order to accomplish some action. The MenuChoice element provides a wrapper to contain the complete combination of selections.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

MenuChoice may generate arrows or other punctuation between multiple GUI elements. The ShortCut may be suppressed, or sometimes it is presented in parentheses after the rest of the items.

Parents

These elements contain menuchoice: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in menuchoice: guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, interface, shortcut.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the MenuChoice.

See Also

accel, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, keycap, keycode, keycombo, keysym, mousebutton, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
You can exit from GNU Emacs with
<menuchoice>
  <shortcut>
    <keycombo><keysym>C-x</keysym><keysym>C-c</keysym></keycombo>
  </shortcut>
  <guimenu>Files</guimenu>
  <guimenuitem>Exit Emacs</guimenuitem>
</menuchoice>.
</para>
```

You can exit from GNU Emacs with **Files** → **Exit Emacs** (**C-x**–**C-c**). Compare this example with the similar example in GUIMenu. Here the KeyCombo and MenuChoice elements are required to process thier content in some intelligent way in order to produce useful output.

For additional examples, see also shortcut.

■ methodname

Name

methodname – The name of a method

Synopsis

Mixed Content Model

methodname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

The `MethodName` element is used to identify the name of a method. This is likely to occur only in documentation about object-oriented programming systems, languages, and architectures.

Processing expectations

Formatted inline.

Parents

These elements contain methodname: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `constructorsynopsis`, `database`, `destructorsynopsis`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `methodsynopsis`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in methodname: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

■ methodparam

Name

methodparam – Parameters to a method

Synopsis

Content Model

methodparam ::= (`modifier`*,`type`?, ((`parameter`,`initializer`?)| `funcparams`), `modifier`*)

Attributes Common attributes

Name Type Default

rep norepeat repeat "norepeat"

choice opt plain req "req"

Description

In the syntax summary of a `ConstructorSynopsis`, `DestructorSynopsis`, or `MethodSynopsis`, `MethodParam` provides the description of a parameter to the method. Typically, this includes the data type of the parameter and its name, but may also include an initial value and other modifiers.

Processing expectations

Formatted inline. For a complete description of the processing expectations, see `ClassSynopsis`.

Parents

These elements contain methodparam: `constructorsynopsis`, `destructorsynopsis`, `methodsynopsis`.

Children

The following elements occur in methodparam: `funcparams`, `initializer`, `modifier`, `parameter`, `type`.

Attributes

choice Choice indicates whether the `MethodParam` is required (`Req` or `Plain`) or optional (`Opt`). Arguments identified as `Plain` are required, but are shown without additional decoration.

rep A Rep value of `Repeat` indicates that the `MethodParam` is reapeatable.

■ methodsynopsis

Name

methodsynopsis – A syntax summary for a method

Synopsis

Content Model

methodsynopsis ::= (`modifier`*, (`type`|`void`)?, `methodname`, (`methodparam`+|`void`), `exceptionname`*,`modi`

Attributes Common attributes

Name Type Default

language CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%method.synop.class;` `%para.char.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%synop.class;` `%tbl.entry.mdl;`

Description

A MethodSynopsis contains the syntax summary of a method (generally speaking, methods in the object-oriented programming language sense).

This is one of the few places where DocBook attempts to model as well as describe. Unlike FuncSynopsis which was designed with C language function prototypes in mind, the content model of MethodSynopsis was designed to capture a wide range of semantics.

Processing expectations

For the most part, the processing application is expected to generate all of the parenthesis, semicolons, commas, and so on. required in the rendered synopsis.

Parents

These elements contain methodsynopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, classsynopsis, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in methodsynopsis: exceptionname, methodname, methodparam, modifier, type, void.

■ mml:math

Name

mml:math – A MathML equation

Synopsis

If the MathML Module is used, Equations and InlineEquations can include mml:math equations.

Description

The Mathematical Markup Language http://www.w3.org/TR/MathML2/ Recommendation from the W3C defines the Mathematical Markup Language, or MathML. MathML is an XML application for describing mathematical notation and capturing both its structure and content.

A complete description of MathML is outside the scope of this reference.

Examples

```
<!DOCTYPE informalequation
          PUBLIC "-//OASIS//DTD DocBook MathML Module V1.0//EN"
          "http://www.oasis-open.org/docbook/xml/mathml/1.0/dbmathml.dtd">
<informalequation>
<mml:math><mml:apply><mml:divide/></mml:apply></mml:math>
</informalequation>
```

■ modespec

Name

modespec – Application-specific information necessary for the completion of an OLink

Synopsis

Mixed Content Model

modespec ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Name Type Default

application BMP, CGM-BINARY, CGM-CHAR, CGM-CLEAR, DITROFF, DVI, EPS, EQN, FAX, GIF, GIF87a, GIF89a, IGES, JPEG, JPG, linespecific, PCX, PIC, PNG, PS, SGML, SVG, TBL, TEX, TIFF, WMF, WPG *None*

Parameter Entities

%docinfo.char.class; %info.class; %ndxterm.char.mix;

%para.char.mix; %refinline.char.mix; %tbl.entry.mdl;

%title.char.mix;

Description

ModeSpec contains application-specific instructions required to process an OLink. See OLink.

Processing expectations

Suppressed. This element provides data for processing but is not expected to be rendered directly.

Parents

These elements contain modespec: appendixinfo, application, articleinfo, attribution, bibliographyin bibliomisc, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotati link, literallayout, lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in modespec: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

Attributes

application Application indicates the nature of the action required to complete the OLink. Application must be a notation declared in the DTD.

Examples

For examples, see olink.

■ modifier

Name

modifier – Modifiers in a synopsis

Synopsis

Mixed Content Model

modifier ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

A `Modifier` identifies additional information about some identifier. For example, the public or private nature of a `OOClass` name, or information about a static or synchronized nature of a `MethodSynopsis`.

Processing expectations

Formatted inline.

Parents

These elements contain modifier: `constructorsynopsis`, `destructorsynopsis`, `fieldsynopsis`, `methodparam`, `methodsynopsis`, `ooclass`, `ooexception`, `oointerface`.

Children

The following elements occur in modifier: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

■ mousebutton

Name

mousebutton – The conventional name of a mouse button

Synopsis

Mixed Content Model

mousebutton ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry

"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

The MouseButton element identifies the conventional name of a mouse button. Because mouse buttons are not physically labelled, the name is just that, a convention. Adding explicit markup for the naming of mouse buttons allow easier translation from one convention to another and might allow an online system to adapt to right- or left-handed usage.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain mousebutton: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, keycombo, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, shortcut, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in mousebutton: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the MouseButton.

See Also

accel, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, keycap, keycode, keycombo, keysym, menuchoice, shortcut.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Select a region of text by dragging the mouse pointer with the
<mousebutton>left</mousebutton> mouse button depressed.  Copy the
selected text to a new location by placing the mouse pointer at the
desired position and pressing the <mousebutton>middle</mousebutton>
button.
</para>
```

Select a region of text by dragging the mouse pointer with the left mouse button depressed. Copy the selected text to a new location by placing the mouse pointer at the desired position and pressing the middle button. For additional examples, see also keycombo.

msg

Name

msg – A message in a message set

Synopsis

Content Model

msg ::= (`title`?,`msgmain`, (`msgsub`|`msgrel`)*)

Attributes Common attributes

Description

In a `MsgSet`, each `MsgEntry` contains at least one `Msg`. A `Msg` consists of a main message (`MsgMain`), and optionally one or more submessages (`MsgSub`) or related messages (`MsgRel`).

Additional information or explanation for the message is contained in the siblings of `Msg` within the `MsgEntry`. See `MsgSet`.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msg: `msgentry`.

Children

The following elements occur in msg: `msgmain`, `msgrel`, `msgsub`, `title`.

Examples

For examples, see `msgset`.

msgaud

Name

msgaud – The audience to which a message in a message set is relevant

Synopsis

Mixed Content Model

msgaud ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`| `citetitle`| `emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|`trademark`| `wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`| `replaceable`|`returnvalue`| `sgmltag`|`structfield`|`structname`|`symbol`| `systemitem`|`token`|`type`|`userinput`|`varname`| `nonterminal`|`anchor`| `author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|`subscript`| `superscript`|`inlinegraphic`| `inlinemediaobject`|`inlineequation`| `synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`| `fieldsynopsis`| `constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`| `beginpage`)*

Attributes Common attributes

Description

`MsgAud` is part of the additional information associated with a message in a `MsgSet`. It identifies the audience to which a particular `Msg` is relevant.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

DocBook doesn't specify anything about how a particular audience might be identified, or how different audiences are distinguished.

On the whole, the semantics of `MsgSet` are not clearly defined.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain msgaud: `msginfo`.

Children

The following elements occur in msgaud: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Examples

For examples, see `msgset`.

■ msgentry

Name

msgentry – A wrapper for an entry in a message set

Synopsis

Content Model

msgentry ::= (`msg`+,`msginfo`?,`msgexplan`*)

Attributes Common attributes

Description

In a MsgSet, each MsgEntry contains some number of messages (Msgs) and additional informative and explanatory material about them.

Processing expectations

Formatted as a displayed block.

On the whole, the semantics of MsgSet are not clearly defined.

Parents

These elements contain msgentry: msgset.

Children

The following elements occur in msgentry: msg, msgexplan, msginfo.

Examples

For examples, see msgset.

■ msgexplan

Name

msgexplan – Explanatory material relating to a message in a message set

Synopsis

Content Model

msgexplan ::= (title?, (calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+)

Attributes Common attributes

Description

A MsgExplan contains some sort of explanatory information about a Msg or a set of Msgs in a MsgEntry.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.

If a MsgEntry contains multiple Msgs and multiple MsgExplans, DocBook makes no assertions about how they are related.

On the whole, the semantics of MsgSet are not clearly defined.

Parents

These elements contain msgexplan: msgentry, simplemsgentry.

Children

The following elements occur in msgexplan: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `synopsis`, `table`, `tip`, `title`, `variablelist`, `warning`.

Examples

For examples, see `msgset`.

■ msginfo

Name

msginfo – Information about a message in a message set

Synopsis

Content Model

msginfo ::= ((`msglevel`|`msgorig`|`msgaud`)*)

Attributes Common attributes

Description

`MsgInfo` provides additional information about a `Msg` in a `MsgEntry`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msginfo: `msgentry`.

Children

The following elements occur in msginfo: `msgaud`, `msglevel`, `msgorig`.

Examples

For examples, see `msgset`.

■ msglevel

Name

msglevel – The level of importance or severity of a message in a message set

Synopsis

Mixed Content Model

msglevel ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

`MsgLevel` is part of the additional information associated with a message in a `MsgSet`. It identifies the relative importance or severity of a message.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msglevel: `msginfo`.

Children

The following elements occur in msglevel: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Examples

For examples, see `msgset`.

■ msgmain

Name

msgmain – The primary component of a message in a message set

Synopsis

Content Model

msgmain ::= (`title`?,`msgtext`)

Attributes Common attributes

Description

Every `Msg` must have one primary message. This is stored in the `MsgMain`. The primary message is distinguished from any number of submessages (`MsgSub`) or related messages (`MsgRel`) that a `Msg` might have.

Processing expectations

Formatted as a displayed block.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msgmain: `msg`.

Children

The following elements occur in msgmain: `msgtext`, `title`.

Examples

For examples, see `msgset`.

■ msgorig

Name

msgorig – The origin of a message in a message set

Synopsis

Mixed Content Model

msgorig ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Description

`MsgOrig` is part of the additional information associated with a message in a `MsgSet`. It identifies the origin or source of a particular `Msg`, for example, a piece of hardware, the operating system, or an application.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msgorig: `msginfo`.

Children

The following elements occur in msgorig: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Examples

For examples, see `msgset`.

■ msgrel

Name

msgrel – A related component of a message in a message set

Synopsis

Content Model

msgrel ::= (`title`?,`msgtext`)

Attributes Common attributes

Description

Every `Msg` has one primary message (`MsgMain`). It may also have any number of related messages, stored in `MsgRel` elements within the same `Msg`.

Related messages are usually messages that appear elsewhere in response to the same event (or set of events) that triggered the main message. For example, if a network client produces a failure or warning message, a related message might appear on the server console.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.
On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msgrel: `msg`.

Children

The following elements occur in msgrel: `msgtext`, `title`.

Examples

For examples, see `msgset`.

■ msgset

Name

msgset – A detailed set of messages, usually error messages

Synopsis

Content Model

msgset ::= (`blockinfo`?, (`title`,`titleabbrev`?)?, (`msgentry`+|`simplemsgentry`+))

Attributes Common attributes

Parameter Entities

`%bookcomponent.content;` `%component.mix;` `%compound.class;`
`%divcomponent.mix;` `%refcomponent.mix;`

Description

`MsgSet` is a complex structure designed to hold a detailed set of messages, usually error messages. In addition to the actual text of each message, it can contain additional information about each message and the messages related to it.

Processing expectations

Formatted as a displayed block.
On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msgset: `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `chapter`, `glossary`, `glossdiv`, `index`, `listitem`, `msgexplan`, `msgtext`, `partintro`, `preface`, `procedure`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `simplesect`, `step`.

Children

The following elements occur in msgset: `blockinfo`, `msgentry`, `simplemsgentry`, `title`, `titleabbrev`.

See Also

`errorcode`, `errorname`, `errortext`, `errortype`.

Examples

```
<!DOCTYPE msgset PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<msgset>
  <msgentry>
    <msg>
       <msgmain>
          <msgtext><para>Record failed CRC</para></msgtext>
       </msgmain>
       <msgsub>
          <msgtext><para>Record <replaceable>n</replaceable>
                     in <replaceable>database</replaceable></para></msgtext>
       </msgsub>
       <msgrel>
          <msgtext><para>File read error on
                    <replaceable>database</replaceable></para></msgtext>
       </msgrel>
       <msgrel>
          <msgtext><para>Panic! Corrupt record!</para></msgtext>
       </msgrel>
     </msg>
     <msginfo>
        <msglevel>severe</msglevel>
        <msgorig>server</msgorig>
        <msgaud>all</msgaud>
     </msginfo>
     <msgexplan>
        <para>
        Indicates that some sort of error occurred while attempting to load
        a record from the database.  Retry.  If failure persists,
        contact the database administrator.
        </para>
     </msgexplan>
 </msgentry>
 <!-- more entries -->
</msgset>
```

Record failed CRC

Record *n* in *database*

File read error on *database*

Panic! Corrupt record! Level: severeOrigin: serverAudience: all Indicates that some sort of error occurred while attempting to load a record from the database. Retry. If failure persists, contact the database administrator.

■ msgsub

Name

msgsub – A subcomponent of a message in a message set

Synopsis

Content Model

msgsub ::= (`title`?,`msgtext`)

Attributes Common attributes

Description

A `MsgSub` represents some subpart of a message. Different `MsgSub`s might arise in different contexts.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

On the whole, the semantics of `MsgSet` are not clearly defined.

Parents

These elements contain msgsub: `msg`.

Children

The following elements occur in msgsub: `msgtext`, `title`.

Examples

For examples, see `msgset`.

■ msgtext

Name

msgtext – The actual text of a message component in a message set

Synopsis

Content Model

msgtext ::= ((`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|
`productionset`| `constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|
`authorblurb`|`epigraph`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Description

The `MsgText` is the actual content of the message in a `MsgMain`, `MsgSub`, or `MsgRel`.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

On the whole, the semantics of `MsgSet` are not clearly defined.

Future Changes

`MsgText` is currently one of the general, technical inlines by accident. In DocBook V4.0, it will be limited to use within a `MsgSet`.

Parents

These elements contain msgtext: `msgmain`, `msgrel`, `msgsub`, `simplemsgentry`.

Children

The following elements occur in msgtext: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, tip, variablelist, warning.

Examples

For examples, see msgset.

■ nonterminal

Name

nonterminal – A non-terminal in an EBNF production

Synopsis

Mixed Content Model

nonterminal ::= (#PCDATA)

Attributes Common attributes

Name Type Default

def CDATA *Required*

Parameter Entities

%cptr.char.mix; %ebnf.inline.hook; %ndxterm.char.mix;

%para.char.mix; %refinline.char.mix; %refname.char.mix;

%tbl.entry.mdl; %tech.char.class; %title.char.mix;

Description

This element is only available if you are using the EBNF Module.

A NonTerminal is a symbol in an EBNF Production that represents a portion of the grammar.

Parents

These elements contain nonterminal: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, rhs, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Attributes

def The def attribute contains a URI reference that points to a `production` where the `nonterminal` is defined.

Note that this *is not* an IDREF, it is a URI reference (as, for example, are href attributes on HTML `a` elements). This is necessary because references to non-terminals might need to cross document boundaries.

■ note

Name

note – A message set off from the text

Synopsis

Content Model

note ::= (`title`?, (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`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`|`sidebar`|`anchor`|`bridgehead`|`remark`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Parameter Entities

`%admon.class;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%highlights.mix;` `%legalnotice.mix;`

`%listpreamble.mix;` `%para.mix;` `%qandaset.mix;`

`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`

`%tabentry.mix;` `%tbl.entry.mdl;` `%textobject.mix;`

Description

A `Note` is an admonition set off from the main text.

In some types of documentation, the semantics of admonitions are clearly defined (`Caution` might imply the possibility of harm to equipement whereas `Warning` might imply harm to a person), but DocBook makes no such assertions.

Processing expectations

Formatted as a displayed block. Often outputs the generated text “Note” or some other visible indication of the type of admonition, especially if a `Title` is not present. Sometimes outputs a graphical icon or another symbol as well.

Parents

These elements contain note: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `chapter`, `colophon`, `dedication`, `entry`, `glossary`, `glossdiv`, `highlights`, `index`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `variablelist`.

Children

The following elements occur in note: address, anchor, beginpage, blockquote, bridgehead, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayou mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, procedure, programlisting, programlistingco, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, title, variablelist.

See Also

caution, important, tip, warning.

Examples

```
<!DOCTYPE note PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<note><title>Upcoming Changes</title>
<para>
Future versions of this feature may not be backward-compatible.
Consider implementing the revised interface now.
</para>
</note>
```

UPCOMING CHANGES

Future versions of this feature may not be backward-compatible. Consider implementing the revised interface now.

■ objectinfo

Name

objectinfo – Meta-information for an object

Synopsis

Content Model

objectinfo ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset| itermset|abbrev|abstract|address|artpagenums|author|authorgroup| authorinitials| bibliomisc|biblioset|collab|confgroup| contractnum|contractsponsor|copyright| corpauthor|corpname|date| edition|editor|invpartnumber|isbn|issn|issuenum| orgname| biblioid|citebiblioid|bibliosource|biblierelation| bibliocoverage| othercredit|pagenums|printhistory|productname| productnumber|pubdate|publisher| publishername|pubsnumber| releaseinfo|revhistory|seriesvolnums|subtitle| title|titleabbrev| volumenum|citetitle|personname|honorific|firstname|surname| lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

The ObjectInfo element is a wrapper for the meta-information about a video, audio, image, or text object.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `DivisionInfo` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain objectinfo: `audioobject`, `imageobject`, `inlinemediaobject`, `mediaobject`, `mediaobjectco`, `textobject`, `videoobject`.

Children

The following elements occur in objectinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

`appendixinfo`, `articleinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `prefaceinfo`, `refsynopsisdivinfo`, `screeninfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setinfo`.

Examples

For examples, see `audioobject`.

■ olink

Name

olink – A link that addresses its target indirectly, through an entity

Synopsis

Mixed Content Model

olink ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`| `citetitle`| `emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|`trademark`| `wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`| `replaceable`|`returnvalue`| `sgmltag`|`structfield`|`structname`|`symbol`| `systemitem`|`token`|`type`|`userinput`|`varname`| `nonterminal`|`anchor`| `author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|`subscript`| `superscript`|`inlinegraphic`| `inlinemediaobject`|`inlineequation`| `synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|

fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

localinfo CDATA *None*

type CDATA *None*

linkmode IDREF *None*

targetptr CDATA *None*

targetdoc CDATA *None*

targetdocent ENTITY *None*

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %indexdivcomponent.mix;
%link.char.class; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;
%word.char.mix;

Description

Unlike Link and ULink, the semantics of OLink are application-specific. OLink provides a mechanism for establishing links across documents, where ID/IDREF linking is not possible and ULink is inappropriate.

In general terms, the strategy employed by OLink is to point to the target document via an external general entity, and point into that document in some application-specific way.

Processing expectations

Formatted inline.

OLink points to its target primarily with the TargetDocEnt attribute. TargetDocEnt must be the name of an entity (previously declared in the DTD or in the document subset).

Because TargetDocEnt is an entity attribute, the entity used as its value must be declared with a notation. Because the target is usually another SGML or XML document, the notation SGML is most often used:

```
<!ENTITY myotherdoc SYSTEM "myotherdoc.sgm" NDATA SGML>
```

The semantics of the link are controlled by three other attributes: LinkMode, LocalInfo, and Type. The LinkMode attribute points to a ModeSpec. The content of ModeSpec describes the semantic of the link in an entirely application-specific way.

The values of LocalInfo and Type may also influence the application. For example, if the ModeSpec describes some sort of query, LocalInfo might hold the query text (allowing multiple OLinks to use the same ModeSpec to achieve different queries with the same query engine).

Linking elements must not be nested within other linking elements (including themselves). Because DocBook is harmonizing towards XML, this restriction cannot easily be enforced by the DTD. The processing of nested linking elements is undefined.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain olink: abbrev, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, contractnum, contractspon

contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, fax, figure, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, holder, honorific, indexdiv, informalfigure, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, quote, refentry, refentrytitle, refmiscinfo, refnamediv, refpurpose, releaseinfo, remark, replaceable, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, shortaffil, simpara, state, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, volumenum, wordasword, year.

Children

The following elements occur in olink: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkmode LinkMode points to the ModeSpec which provides additional application-specific information for resolving this OLink.

localinfo LocalInfo hold additional information that may be used with the ModeSpec (pointed to by LinkMode) by the application when resolving this OLink.

targetdocent TargetDocEnt specifies the name of an entity that is to be used as part of the OLink. Exactly how the link is resolved is application dependent and may be influenced by the MoreInfo and LocalInfo attributes.

type Type is available for application-specific customization of the linking behavior.

See Also

anchor, link, ulink, xref.

Examples

In this example, we see how an OLink might be used for searching. Here the ModeSpec describes the search query (in a fictitious and entirely concocted syntax): “look in the titles of sections and return links using the title as the text of the link”. When the user selects the link, the application is expected to perform the query and then might display the list of titles as a pop-up window in the user interface.

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd" [
<!ENTITY refbook SYSTEM "refbook.sgm" CDATA SGML>
]>
<chapter>
<chapterinfo>
  <modespec id="s1query">query in sect*, title return link text title</modespec>
</chapterinfo>
<title>Printing</title>

<para>
Blah, blah, blah.
</para>

<sect1><title>See Also</title>
<para>
For more information <olink targetdocent="refbook" linkmode="s1query"
localinfo="print or printing">about printing</olink>, consult
the <ulink url="refbook.sgm">reference manual</ulink>.
</para>
</sect1>

</chapter>
```

■ ooclass

Name

ooclass – A class in an object-oriented programming language

Synopsis

Content Model

ooclass ::= (`modifier*`,`classname`)

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

The `OOClass` element identifies programming language classes, generally from object-oriented programming languages. The `OOClass` is a wrapper for the `ClassName` plus some `Modifier`s.

Processing expectations

Formatted inline.

Parents

These elements contain ooclass: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsis`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`

refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in ooclass: classname, modifier.

■ ooexception

Name

ooexception – An exception in an object-oriented programming language

Synopsis

Content Model

ooexception ::= (modifier*,exceptionname)

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The OOException element identifies programming language exceptions, generally from object-oriented programming languages. The OOException is a wrapper for the ExceptionName plus some Modifiers.

Processing expectations

Formatted inline.

Parents

These elements contain ooexception: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsis, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in ooexception: exceptionname, modifier.

■ oointerface

Name

oointerface – An interface in an object-oriented programming language

Synopsis

Content Model

oointerface ::= (`modifier*`,`interfacename`)

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

The `OOInterface` element identifies programming language interfaces, generally from object-oriented programming languages. The `OOInterface` is a wrapper for the `InterfaceName` plus some `Modifier`s.

Processing expectations

Formatted inline.

Parents

These elements contain oointerface: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsis`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in oointerface: `interfacename`, `modifier`.

■ option

Name

option – An option for a software command

Synopsis

Mixed Content Model

option ::= (#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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

```
%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;
%tech.char.class; %title.char.mix;
```

Description

`Option` identifies an optional argument to a software command.

Processing expectations

Formatted inline.

DocBook does not specify whether or not a symbol (such as `-` or `/`) is generated before the content of `Option`, or what that symbol might be. Generating the text may or may not be desirable, but in either case, it is an interchange issue. See Appendix F.

Future Changes

The content model of `Option` will be constrained to `(#PCDATA | Replaceable | InlineGraphic)` in DocBook V4.0.

Parents

These elements contain option: `action`, `application`, `arg`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `group`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

See Also

`computeroutput`, `constant`, `literal`, `markup`, `optional`, `parameter`, `prompt`, `replaceable`, `sgmltag`, `userinput`, `varname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <option>-a</option> option on the <acronym>UNIX</acronym>
<command>ls</command> command or the <option>/r</option> option on the
<acronym>DOS</acronym> <command>attrib</command>
command, for example.
</para>
```

The -a option on the UNIX **ls** command or the /r option on the DOS **attrib** command, for example. For additional examples, see also optional, refentry.

■ optional

Name

optional – Optional information

Synopsis

Mixed Content Model

optional ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The Optional element indicates that a specified argument, option, or other text is optional. The precise meaning of “optional” varies according to the application or process begin documented.

Processing expectations

Formatted inline.

Optional arguments in a Synopsis are usually given special typographic treatment, often they are surrounded by square brackets. The Optional tag is expected to *generate* the brackets.

Outside a Synopsis, the typographic treatment of Optional is application-specific.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain optional: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, replaceable, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in optional: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

See Also

computeroutput, constant, literal, markup, option, parameter, prompt, replaceable, sgmltag, userinput, varname.

Examples

The UNIX **ls** command could be documented as follows:

```
<!DOCTYPE synopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<synopsis>
ls <optional><option>-abcCdfFgilLmnopqrRstux1</option></optional>
   <optional>names</optional>
</synopsis>
```

ls [-abcCdfFgilLmnopqrRstux1] [names] which might generate the following output:

```
ls [ -abcCdfFgilLmnopqrRstux1 ]
   [names]
```

■ orderedlist

Name

orderedlist – A list in which each entry is marked with a sequentially incremented label

Synopsis

Content Model

orderedlist ::= (blockinfo?, (title,titleabbrev?)?, (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|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)*, listitem+)

Attributes Common attributes

Name Type Default

inheritnum ignore | inherit "ignore"

spacing compact | normal *None*

continuation	continues restarts
"restarts"	
numeration	arabic loweralpha lowerroman upperalpha upperroman
None	

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%example.mix;` `%footnote.mix;`

`%glossdef.mix;` `%highlights.mix;` `%indexdivcomponent.mix;`

`%legalnotice.mix;` `%list.class;` `%para.mix;`

`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`

`%sidebar.mix;` `%tabentry.mix;` `%tbl.entry.mdl;`

`%textobject.mix;`

Description

In an `OrderedList`, each member of the list is marked with a numeral, letter, or other sequential symbol (such as roman numerals).

Processing expectations

Formatted as a displayed block.

If no value is specified for Numeration, Arabic numerals (1, 2, 3, ...) are to be used.

In nested lists, DocBook does not specify the sequence of numerations.

Note that the attributes of `OrderedList` have a significant influence on the processing expectations.

Future Changes

Introductory material may appear before the first list item.

Parents

These elements contain orderedlist: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `example`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `highlights`, `important`, `index`, `indexdiv`, `informalexample`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `warning`.

Children

The following elements occur in orderedlist: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockinfo`, `blockquote`, `bridgehead`, `caution`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `fieldsynopsis`, `formalpara`, `funcsynopsis`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `listitem`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `note`, `para`, `programlisting`, `programlistingco`, `remark`, `screen`, `screenco`, `screenshot`, `simpara`, `synopsis`, `tip`, `title`, `titleabbrev`, `warning`.

Attributes

continuation If Continuation is specified, it indicates how list numbering should begin relative to the immediately preceding list. `Restarts`, the default, indicates that numbering should begin again at 1. `Continues` indicates that numbering should begin where the preceding list left off.

inheritnum In a nested list, InheritNum indicates whether or not the enumeration of interior lists should include the numbers of containing list items. If InheritNum is `Inherit` then the third item of a list inside the second item of a list inside the fourth item of a list might be enumerated as "4.2.3". If it is `Ignore`, the default, then it would be simply "3". (The Numeration attribute controls the actual format of the item numbers, of course.)

numeration Numeration specifies the style of numbering to be used for items in the current `OrderedList`.

spacing Spacing indicates whether or not the vertical space in the list should be minimized.

See Also

`calloutlist`, `itemizedlist`, `listitem`, `segmentedlist`, `simplelist`, `variablelist`.

Examples

```
<!DOCTYPE orderedlist PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<orderedlist numeration="lowerroman">
<listitem>
<para>One</para>
</listitem>
<listitem>
<para>Two</para>
</listitem>
<listitem>
<para>Three</para>
</listitem>
<listitem>
<para>Four</para>
</listitem>
</orderedlist>
```

i One

ii Two

iii Three

iv Four

■ orgdiv

Name

orgdiv – A division of an organization

Synopsis

Mixed Content Model

orgdiv ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`OrgDiv` identifies a division in an organization, such as “Chrysler” in “General Motors.”

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain orgdiv: `affiliation`.

Children

The following elements occur in orgdiv: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `corpname`, `jobtitle`, `orgname`, `shortaffil`.

Examples

For examples, see `author`.

■ orgname

Name

orgname – The name of an organization other than a corporation

Synopsis

Mixed Content Model

orgname ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Name Type Default

otherclass CDATA *None*

class
- consortium
- corporation
- informal
- nonprofit
- other

None

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

An OrgName identifies the name of an organization or corporation. Outside of an Affiliation, CorpName is a more appropriate element for the name of a corporation.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain orgname: affiliation, appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in orgname: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

affiliation, corpname, jobtitle, orgdiv, publishername, shortaffil.

Examples

For examples, see author, authorgroup, bookinfo, contractsponsor.

■ otheraddr

Name

otheraddr – Uncategorized information in address

Synopsis

Mixed Content Model

otheraddr ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Description

Within an Address, OtherAddr is a wrapper for parts of an address other than Street, POB, Postcode, City, State, Country, Phone, Fax, and Email, all of which have elements specific to their content.

In early versions of DocBook, Address was not allowed to contain character data (it was a database-like collection of fields). In that context, a wrapper was necessary for any random pieces of information that might be required for an address. With the introduction of character data directly in the Address element, OtherAddr may have lost most of its *raison d'être*.

Processing expectations

Formatted inline. This element may inherit the verbatim qualities of an Address.

Parents

These elements contain otheraddr: address.

Children

The following elements occur in otheraddr: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `city`, `country`, `email`, `fax`, `phone`, `pob`, `postcode`, `state`, `street`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>Please deliver to:
<address>
<othername>The Law Offices of Dewey, Cheatem, and Howe</othername>
<street>100 Main Street</street>
<otheraddr>Third Floor</otheraddr>
<city>Anytown</city>, <state>NY</state> <postcode>12345</postcode>
<country>USA</country>
</address>
</para>
```

Please deliver to:

```
The Law Offices of Dewey, Cheatem, and Howe
100 Main Street
Third Floor
Anytown, NY 12345
USA
```

■ othercredit

Name

othercredit – A person or entity, other than an author or editor, credited in a document

Synopsis

Content Model

othercredit ::= ((`personname`| (`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`)+), (`personblurb`|`email`|`address`)*)

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%docinfo.char.class;` `%info.class;`

`%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;`

`%tbl.entry.mdl;` `%title.char.mix;`

Description

DocBook allows you to directly identify `Author`s and `Editor`s. `OtherCredit` provides a mechanism for identifying other individuals, for example, contributors or production editors, in a similar context.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of OtherCredit in DocBook V4.0. A new wrapper element will be created to hold OtherCredit, AuthorBlurb, and Affiliation.

Parents

These elements contain othercredit: appendixinfo, application, articleinfo, attribution, authorgroup, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayout, lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in othercredit: address, affiliation, authorblurb, contrib, email, firstname, honorific, lineage, othername, personblurb, personname, surname.

See Also

author, authorblurb, authorgroup, collab, collabname, contrib, corpauthor, editor, personblurb, personname.

Examples

```
<!DOCTYPE articleinfo PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<articleinfo>
  <title>Something Snappy</title>
  <author>
    <firstname>Norman</firstname>
    <surname>Walsh</surname>
  </author>
  <othercredit>
    <firstname>John</firstname>
    <surname>Doe</surname>
    <contrib>Extensive review and rough drafts of Section 1.3, 1.4, and 1.5
      </contrib>
  </othercredit>
  <pubsnumber>5</pubsnumber>
</articleinfo>
```

■ othername

Name

othername – A component of a persons name that is not a first name, surname, or lineage

Synopsis

Mixed Content Model

othername ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix; %info.class; %person.ident.mix;`

Description

`OtherName` is a generic wrapper for parts of an individual's name other than `Honorific`, `FirstName`, `Surname` and `Lineage`. One common use is to identify an individual's middle name or initial. Use Role to classify the type of other name.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain othername: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `personname`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in othername: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `firstname`, `honorific`, `lineage`, `surname`.

Examples

For examples, see `author`, `authorgroup`, `otheraddr`.

■ pagenums

Name

pagenums – The numbers of the pages in a book, for use in a bibliographic entry

Synopsis

Mixed Content Model

pagenums ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix; %info.class;`

Description

`PageNums` identifies a page or range of pages. This may be useful in the bibliography of a book, to indicate the number of pages, or in a citation to a journal article.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain pagenums: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in pagenums: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

For examples, see `bibliography`, `biblioset`.

■ para

Name

para – A paragraph

Synopsis

Mixed Content Model

para ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`|`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`|`simplelist`|`variablelist`|`caution`|`important`|`note`|`tip`|`warning`|`literallayout`|`programlisting`|`programlistingco`|`screen`|`screenco`|`screenshot`|`address`|`blockquote`|`graphic`|`graphicco`|`mediaobject`|`mediaobjectco`|`informalequation`|`informalexample`|`informalfigure`|`informaltable`|`equation`|`example`|`figure`|`table`)*

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%footnote.mix;`
`%glossdef.mix;` `%highlights.mix;` `%indexdivcomponent.mix;`
`%legalnotice.mix;` `%listpreamble.mix;` `%para.class;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%tabentry.mix;` `%tbl.entry.mdl;`
`%textobject.mix;`

Description

A Para is a paragraph. Paragraphs in DocBook may contain almost all inlines and most block elements. Sectioning and higher-level structural elements are excluded. DocBook offers two variants of paragraph: SimPara, which cannot contain block elements, and FormalPara, which has a title.

Some processing systems may find the presence of block elements in a paragraph difficult to handle. On the other hand, it is frequently most logical, from a structural point of view, to include block elements, especially informal block elements, in the paragraphs that describe their content. There is no easy answer to this problem.

Processing expectations

Formatted as a displayed block.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain para: abstract, answer, appendix, article, authorblurb, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, epigraph, example, footnote, formalpara, glossary, glossdef, glossdiv, highlights, important, index, indexdiv, informalexample, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, partintro, personblurb, preface, printhistory, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in para: abbrev, acronym, action, address, anchor, application, author, authorinitials, beginpage, blockquote, calloutlist, caution, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, equation, errorcode, errorname, errortext, errortype, example, exceptionname, fieldsynopsis, figure, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glosslist, glossterm, graphic, graphicco, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, important, indexterm, informalequation, informalexample, informalfigure, informaltable, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, itemizedlist, keycap, keycode, keycombo, keysym, link, literal, literallayout, markup, medialabel, mediaobject, mediaobjectco, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, note, olink, ooclass, ooexception, oointerface, option, optional, orderedlist, othercredit, parameter, personname, phrase, productname, productnumber, programlisting, programlistingco, prompt, property, quote, remark, replaceable, returnvalue, revhistory, screen, screenco, screenshot, segmentedlist, sgmltag, simplelist, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, table, tip, token, trademark, type, ulink, userinput, variablelist, varname, warning, wordasword, xref.

See Also

formalpara, simpara.

Examples

Ordinary paragraphs can contain most block elements:

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The component suffered from three failings:
```

```
<itemizedlist>
<listitem><para>It was slow</para></listitem>
<listitem><para>It ran hot</para></listitem>
<listitem><para>It didn't actually work</para></listitem>
</itemizedlist>
Of these three, the last was probably the most important.
</para>
```

The component suffered from three failings:

- It was slow
- It ran hot
- It didn't actually work

Of these three, the last was probably the most important. Formal paragraphs include a title:

```
<!DOCTYPE formalpara PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<formalpara><title>A Test</title>
<para>
This is a test.  This is only a test.  Had this been a real
example, it would have made more sense.
</para>
</formalpara>
```

A Test This is a test. This is only a test. Had this been a real example, it would have made more sense.

Simple paragraphs may not contain block elements:

```
<!DOCTYPE simpara PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<simpara>
Just the text, ma'am.
</simpara>
```

Just the text, ma'am. For additional examples, see also `abbrev`, `abstract`, `accel`, `ackno`, `acronym`, `action`, `address`, `anchor`, `appendix`, `application`, `article`, `authorgroup`, `beginpage`, `blockquote`, `book`, `bookinfo`, `caution`, `chapter`, `citation`, `citerefentry`, `citetitle`, `classname`, `collab`, `colophon`, `command`, `computeroutput`, `constant`, `contractsponsor`, `database`, `emphasis`, `envar`, `errorcode`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `formalpara`, `glossary`, `glosslist`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `highlights`, `html-form`, `imageobject`, `important`, `indexterm`, `informalequation`, `informalfigure`, `inlineequation`, `inlinemediaobject`, `itemizedlist`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `manvolnum`, `markup`, `medialabel`, `menuchoice`, `mousebutton`, `msgset`, `note`, `olink`, `option`, `orderedlist`, `otheraddr`, `parameter`, `part`, `personname`, `phrase`, `printhistory`, `procedure`, `productionset`, `productname`, `productnumber`, `programlistingco`, `prompt`, `property`, `qandaset`, `quote`, `refentry`, `reference`, `remark`, `returnvalue`, `screenco`, `sect1info`, `section`, `segmentedlist`, `seriesvolnums`, `set`, `shortcut`, `sidebar`, `simplelist`, `simplesect`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `table`, `tip`, `title`, `titleabbrev`, `token`, `trademark`, `type`, `ulink`, `userinput`, `variablelist`, `varname`, `videoobject`, `warning`, `wordasword`, `xref`.

■ paramdef

Name

paramdef – Information about a function parameter in a programming language

Synopsis

Mixed Content Model

paramdef ::= (#PCDATA|`type`|`replaceable`|`parameter`|`funcparams`)*

Attributes Common attributes

Description

In the syntax summary for a function in a programming language, `ParamDef` provides the description of a parameter to the function. Typically, this includes the data type of the parameter and its name. For parameters that are pointers to functions, it also includes a summary of the nested parameters.

Within the `ParamDef`, the parameter name is identified with `Parameter`, and the rest of the content is assumed to be the data type.

In the following definition, *`str`* is the name of the parameter and char * is its type:

```
<paramdef>char *<parameter>str</parameter></paramdef>
```

Sometimes a data type requires punctuation on both sides of the parameter. For example, the *`a`* parameter in this definition is an array of char *:

```
<paramdef>char *<parameter>a</parameter>[]</paramdef>
```

Processing expectations

Formatted inline. For a complete description of the processing expectations, see `FuncSynopsis`.

Parents

These elements contain paramdef: `funcprototype`.

Children

The following elements occur in paramdef: `funcparams`, `parameter`, `replaceable`, `type`.

See Also

`funcdef`, `funcparams`, `funcprototype`, `funcsynopsisinfo`, `function`, `parameter`, `returnvalue`, `varargs`, `void`.

Examples

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcprototype>
  <funcdef>int <function>max</function></funcdef>
  <paramdef>int <parameter>int1</parameter></paramdef>
  <paramdef>int <parameter>int2</parameter></paramdef>
</funcprototype>
</funcsynopsis>
```

int **max** (*`int1`*, *`int2`*);
int *`int1`*;
int *`int2`*; For additional examples, see also `funcparams`, `funcsynopsis`, `refentry`.

■ parameter

Name

parameter – A value or a symbolic reference to a value

Synopsis

Mixed Content Model

parameter ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"
class	command function option	*None*

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

A Parameter identifies something passed from one part of a computer system to another. In this regard Parameter is fairly generic, but it may have a more constrained semantic in some contexts (for example in a ParamDef).

In a document that describes more than one kind of parameter, for example, parameters to functions and commands, the Class attribute can be used to distinguish between them, if necessary.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

In DocBook V4.0, the content model of Parameter will be constrained to (#PCDATA | Replaceable | InlineGraphic).

Parents

These elements contain parameter: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, methodparam, msgaud, olink, option, optional, para, paramdef, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in parameter: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

class Class indicates the type of Parameter.

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Parameter.

See Also

computeroutput, constant, funcdef, funcparams, funcprototype, funcsynopsisinfo, function, literal, markup, option, optional, paramdef, prompt, replaceable, returnvalue, sgmltag, userinput, varargs, varname, void.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Using the <parameter class='command'>/w</parameter> parameter on the
DOS <command>dir</command> command prints a wide directory listing.
</para>
```

Using the */w* parameter on the DOS **dir** command prints a wide directory listing. For additional examples, see also funcparams, funcsynopsis, paramdef, refentry.

■ part

Name

part – A division in a book

Synopsis

Content Model

part ::= (beginpage?,partinfo?, (title,subtitle?,titleabbrev?), partintro?, (appendix| chapter|toc|lot|index|glossary|bibliography|article| preface|refentry|reference)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

label CDATA *None*

Description

Parts segment a book into divisions. Each division can contain a number of component-level elements, such as Chapters.

Processing expectations

Formatted as a displayed block. Parts almost always introduce a forced page break. Sometimes starts on the next recto page. Frequently, they also produce a part separator page, on which may be printed the content of the PartIntro.

Future Changes

In DocBook V4.0, the ToC element in the content model will be replaced by TocChap.

Parents

These elements contain part: book.

Children

The following elements occur in part: appendix, article, beginpage, bibliography, chapter, glossary, index, lot, partinfo, partintro, preface, refentry, reference, subtitle, title, titleabbrev, toc.

Attributes

label Label specifies an identifying string for presentation purposes. The first Part in a Book might be labeled "Part I", for example.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the Part.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

appendix, article, book, chapter, colophon, dedication, partintro, preface, set.

Examples

The following example comes from *Java in a Nutshell*:

```
<!DOCTYPE part PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<part label="II">
<title>Programming with the Java API</title>
<partintro>
<para>
The sections in Part II present real-world examples of
programming with Java.  You can study and learn from the
examples, and you can adapt them for use in your own programs.
</para>
```

```
<para>
The example code in these chapters is available for downloading.
See <systemitem role="url">http://www.ora.com/catalog/books/javanut</systemitem>.
</para>

<literallayout>
<xref linkend="jnut-ch-04"/>
<xref linkend="jnut-ch-05"/>
<xref linkend="jnut-ch-06"/>
<xref linkend="jnut-ch-07"/>
<xref linkend="jnut-ch-08"/>
<xref linkend="jnut-ch-09"/>
</literallayout>
</partintro>
<chapter id="jnut-ch-04"><title/><para>...</para></chapter>
<chapter id="jnut-ch-05"><title/><para>...</para></chapter>
<chapter id="jnut-ch-06"><title/><para>...</para></chapter>
<chapter id="jnut-ch-07"><title/><para>...</para></chapter>
<chapter id="jnut-ch-08"><title/><para>...</para></chapter>
<chapter id="jnut-ch-09"><title/><para>...</para></chapter>
</part>
```

■ partinfo

Name

partinfo – Meta-information for a `Part`

Synopsis

Content Model

partinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `PartInfo` element is a wrapper for a large collection of meta-information about a `Part`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain partinfo: `part`.

Children

The following elements occur in partinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

■ partintro

Name

partintro – An introduction to the contents of a part

Synopsis

Content Model

partintro ::= ((title,subtitle?,titleabbrev?)?, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)+, (sect1*| (refentry)*| simplesect*| (section)*))| (sect1+| (refentry)+| simplesect+| (section)+)))

Attributes Common attributes

Name Type Default

label CDATA *None*

Description

PartIntro contains introductory text, often an overview of the content of the Part.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.
PartIntro content is often printed on a part separator page.

Parents

These elements contain partintro: part, reference.

Children

The following elements occur in partintro: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry,

remark, screen, screenco, screenshot, sect1, section, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

See Also

appendix, article, book, chapter, colophon, dedication, part, preface, set.

Examples

For examples, see part.

■ personblurb

Name

personblurb – A short description or note about a person

Synopsis

Content Model

personblurb ::= (title?, (formalpara|para|simpara)+)

Attributes Common attributes

Description

A short description of a person.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.

Parents

These elements contain personblurb: author, editor, othercredit.

Children

The following elements occur in personblurb: formalpara, para, simpara, title.

See Also

author, authorblurb, authorgroup, collab, collabname, contrib, corpauthor, editor, othercredit, personname.

■ personname

Name

personname – The personal name of an individual

Synopsis

Content Model

personname ::= ((honorific|firstname|surname|lineage|othername)+)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %gen.char.class; %info.class;
%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;
%tbl.entry.mdl; %title.char.mix;

Description

The personname identifies the personal name of an individual.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

Parents

These elements contain personname: address, appendixinfo, application, articleinfo, attribution, author, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, editor, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayout, lotentry, member, msgaud, objectinfo, olink, othercredit, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in personname: firstname, honorific, lineage, othername, surname.

See Also

author, authorblurb, authorgroup, collab, collabname, contrib, corpauthor, editor, othercredit, personblurb.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>Starting in DocBook 4.2, personal names, like
<personname><firstname>Albert</firstname><surname>Einstein</surname></personname>
can be marked-up inline.</para>
```

Starting in DocBook 4.2, personal names, like [personname] [firstname] Albert [/firstname] [surname] Einstein [/surname] [/personname] can be marked-up inline.

■ phone

Name

phone – A telephone number

Synopsis

Mixed Content Model

phone ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Description

Phone identifies a telephone number in an Address.

Processing expectations

Formatted inline. Sometimes suppressed. In an Address, this element may inherit the verbatim qualities of an address.

Parents

These elements contain phone: address.

Children

The following elements occur in phone: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

address, city, country, email, fax, otheraddr, pob, postcode, state, street.

Examples

For examples, see address.

■ phrase

Name

phrase – A span of text

Synopsis

Mixed Content Model

phrase ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

```
%gen.char.class; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;
```

Description

The Phrase element in DocBook has no specific semantic. It is provided as a wrapper around a selection of words smaller than a paragraph so that it is possible to provide an ID or other attributes for them.

For example, if you are making note of changes to a document using one of the effectivity attributes, you might use Phrase to mark up specific sentences with revisions.

Processing expectations

Formatted inline.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain phrase: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, textobject, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in phrase: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

abbrev, acronym, emphasis, foreignphrase, quote, wordasword.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Effectivity attributes can be used to keep track of modifications
to a document <phrase revisionflag="deleted">at the word or
sentence level</phrase><phrase revisionflag="added"> as long as the number
and complexity of changes is not too high</phrase>.
</para>
```

Effectivity attributes can be used to keep track of modifications to a document at the word or sentence levelas long as the number and complexity of changes is not too high. For additional examples, see also `audioobject`, `imageobject`, `informalfigure`, `inlinemediaobject`.

■ pob

Name

pob – A post office box in an address

Synopsis

Mixed Content Model

pob ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`POB` is a post office box number in an `Address`.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain pob: `address`.

Children

The following elements occur in pob: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `city`, `country`, `email`, `fax`, `otheraddr`, `phone`, `postcode`, `state`, `street`.

Examples

For examples, see `address`.

■ postcode

Name

postcode – A postal code in an address

Synopsis

Mixed Content Model

postcode ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`PostCode` is a postal code (in the United States, a ZIP code) in an `Address`.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain postcode: `address`.

Children

The following elements occur in postcode: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `city`, `country`, `email`, `fax`, `otheraddr`, `phone`, `pob`, `state`, `street`.

Examples

For examples, see `address`, `otheraddr`.

■ preface

Name

preface – Introductory matter preceding the first chapter of a book

Synopsis

Content Model

preface ::= (`beginpage`?,`prefaceinfo`?, (`title`,`subtitle`?,`titleabbrev`?), (`toc`|`lot`|`index`|`glossary`|`bibliography`)*, `tocchap`?, (((`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)+, (`sect1`*|(`refentry`)*|`simplesect`*|(`section`)*))|(`sect1`+|(`refentry`)+|`simplesect`+|(`section`)+)), (`toc`|`lot`|`index`|`glossary`|`bibliography`)*)

Attributes Common attributes

Name Type Default

status CDATA *None*

Parameter Entities

`%partcontent.mix;`

Description

`Preface` is a preface or forward in a `Book`. The `Preface` element may appear more than once and should be used for all introductory chapter-like material. For example, a `Book` might have both a *Foreward* and an *Introduction*. Both should be tagged as `Preface`s in DocBook.

Processing expectations

Formatted as a displayed block. Usually introduces a forced page break and often starts on the next recto page. It is common for the page numbers in prefaces to be displayed as roman numerals rather than arabic numerals. `Preface`s are usually listed in the Table of Contents.

Parents

These elements contain preface: `book`, `part`.

Children

The following elements occur in preface: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `bibliography`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glossary`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `index`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `lot`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `prefaceinfo`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refentry`, `remark`, `screen`, `screenco`, `screenshot`, `sect1`, `section`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `simplesect`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `toc`, `tocchap`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `Preface`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`appendix`, `article`, `book`, `chapter`, `colophon`, `dedication`, `part`, `partintro`, `set`.

Examples

For examples, see `book`.

■ prefaceinfo

Name

prefaceinfo – Meta-information for a `Preface`

Synopsis

Content Model

prefaceinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The PrefaceInfo element is a wrapper for a large collection of meta-information about a Preface. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain prefaceinfo: preface.

Children

The following elements occur in prefaceinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

■ primary

Name

primary – The primary word or phrase under which an index term should be sorted

Synopsis

Mixed Content Model

primary ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

sortas CDATA *None*

Description

In an IndexTerm, Primary identifies the most significant word or words in the entry. All IndexTerms must have a Primary.

Processing expectations

Suppressed. This element provides data for processing but it is not rendered in the primary flow of text.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain primary: indexterm.

Children

The following elements occur in primary: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

sortas SortAs specifies the string by which the element's content is to be sorted. If unspecified, the proper content is used.

See Also

indexentry, indexterm, primaryie, secondary, secondaryie, see, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see chapter, indexterm.

■ primaryie

Name

primaryie – A primary term in an index entry, not in the text

Synopsis

Mixed Content Model

primaryie ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

linkends IDREFS *None*

Description

PrimaryIE identifies the most significant word or words in an IndexEntry. IndexEntrys occur in an Index, not in the flow of the text. They are part of a formatted index, not markers for indexing.

If a document includes both IndexTerms and IndexEntrys, the IndexEntrys are usually constructed from the IndexTerms by some external process.

Processing expectations

Formatted as a displayed block. The PrimaryIE starts a new entry in the Index.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain primaryie: indexentry.

Children

The following elements occur in primaryie: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkends Linkends, if used, points to the IndexTerms indexed by this entry.

See Also

indexentry, indexterm, primary, secondary, secondaryie, see, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see index.

■ printhistory

Name

printhistory – The printing history of a document

Synopsis

Content Model

printhistory ::= ((formalpara|para|simpara)+)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The PrintHistory of a document identifies when various editions and revisions were printed.

Processing expectations

Formatted as a displayed block. Sometimes suppressed.

Parents

These elements contain printhistory: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in printhistory: formalpara, para, simpara.

See Also

date, edition, pubdate, releaseinfo, revhistory.

Examples

```
<!DOCTYPE printhistory PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<printhistory>
<para>
September, 1996    First Printing
</para>
</printhistory>
```

For a printed example of PrintHistory, consult the reverse of the full title page of this book (if you're holding the print version from O'Reilly).

■ procedure

Name

procedure – A list of operations to be performed in a well-defined sequence

Synopsis

Content Model

procedure ::= (blockinfo?, (title,titleabbrev?)?, (calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)*, step+)

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%compound.class; %divcomponent.mix; %qandaset.mix;

%refcomponent.mix; %revdescription.mix; %sidebar.mix;

Description

A Procedure encapsulates a task composed of Steps (and possibly, SubSteps). Procedures are usually performed sequentially, unless individual Steps direct the reader explicitly.

Often it is important to assure that certain conditions exist before a procedure is performed, and that the outcome of the procedure matches the expected results. DocBook does not provide explicit semantic markup for these pre- and post-conditions. Instead, they must be described as steps (check the pre-conditions in the first step and the results in the last step), or described outside the body of the procedure.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain procedure: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caution, chapter, constraintdef, glossary, glossdiv, important, index, listitem, msgexplan, msgtext, note, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, tip, warning.

Children

The following elements occur in procedure: abstract, address, anchor, authorblurb, beginpage, blockinfo, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, step, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Examples

```
<!DOCTYPE procedure PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<procedure><title>An Example Procedure</title>
<step>
  <para>
    A Step
  </para>
</step>
<step>
  <para>
    Another Step
  </para>
  <substeps>
    <step>
      <para>
        Substeps can be nested indefinitely deep.
      </para>
    </step>
  </substeps>
</step>
<step>
  <para>
    A Final Step
  </para>
</step>
</procedure>
```

AN EXAMPLE PROCEDURE

1. A Step
2. Another Step
   (a) Substeps can be nested indefinitely deep.
3. A Final Step

■ production

Name

production – A production in a set of EBNF productions

Synopsis

Content Model

production ::= (`lhs`,`rhs`,`constraint`*)

Attributes Common attributes

Description

This element is only available if you are using the EBNF Module.

A `Production` is single production in an Extended Backus-Naur Form (EBNF) grammar.

Individual productions have two parts, a left hand side (`lhs`) and a right hand side (`rhs`). Each `NonTerminal` on the left hand side is defined in terms of other non-terminals and literals on the right hand side.

Parents

These elements contain production: `productionset`.

Children

The following elements occur in production: `constraint`, `lhs`, `rhs`.

Examples

For examples, see `productionset`.

■ productionrecap

Name

productionrecap – A cross-reference to an EBNF production

Synopsis

Content Model

productionrecap ::= EMPTY

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

Description

This element is only available if you are using the EBNF Module.

A `ProductionRecap` is a cross reference to a `Production`.

Processing expectations

A `ProductionRecap` is rendered exactly like the production to which it refers. The purpose of the `ProductionRecap` element is to allow a `ProductionSet` to contain copies of `Production`s defined elsewhere. This frequently makes it easier for readers to understand.

Parents

These elements contain productionrecap: `productionset`.

Attributes

linkend FIXME:

■ productionset

Name

productionset – A set of EBNF productions

Synopsis

Content Model

productionset ::= ((`title`,`titleabbrev`?)?, (`production`|`productionrecap`)+)

Attributes Common attributes

Parameter Entities

`%bookcomponent.content;` `%component.mix;` `%compound.class;`
`%divcomponent.mix;` `%ebnf.block.hook;` `%refcomponent.mix;`

Description

This element is only available if you are using the EBNF Module.

A `ProductionSet` is a collection of Extended Backus-Naur Form (EBNF) `Productions`.

EBNF is a notation for describing the grammar of context-free languages. Even if you aren't conversant in the programming language concepts of context-free languages and grammars, it's not really as hard to understand as it sounds.

A set of EBNF productions describes the legal arrangements of tokens in a language. Consider arithmetic expressions as a simple example.

The expression "`3 + 4`" is valid and so is "`3 + 4 - 5`", but "`3 - + - 4`" is not, nor is "`3 + 4 6`". We can use EBNF to describe all the possible legal arrangements: [productionset] [title] Arithemetic Expressions [/title]

[production] [lhs] Expression [/lhs]

[rhs] [nonterminal] ArithExpression [/nonterminal] | [nonterminal] MultExpression [/nonterminal]

Does this grammar actually get precedence right? [/rhs] [/production]

[production] [lhs] ArithExpression [/lhs]

[rhs] [nonterminal] Expression [/nonterminal]

'+' [nonterminal] MultExpression [/nonterminal] | [nonterminal] Expression [/nonterminal]

'-' [nonterminal] MultExpression [/nonterminal] [/rhs] [/production]

[production] [lhs] MultExpression [/lhs]

[rhs] [nonterminal] MultExpression [/nonterminal]

'*' [nonterminal] MultExpression [/nonterminal] | [nonterminal] MultExpression [/nonterminal]

'/' [nonterminal] MultExpression [/nonterminal] | [nonterminal] Number [/nonterminal] [/rhs]

[constraint] [/constraint] [/production]

[production] [lhs] Number [/lhs]

[rhs] [0-9]+ [/rhs] [/production] [/productionset] [constraintdef] [title] Division by Zero [/title]

Division by zero is an error. Constraints, such as this one, are used to express conditions that cannot be expressed in the grammar. [/constraintdef]

Processing expectations

Formatted as a displayed block. The detailed processing expecations with respect to individual productions, left-hand sides, and right-hand sides are quite complex.

`Productions` should be numbered.

Parents

These elements contain productionset: `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `chapter`, `glossary`, `glossdiv`, `index`, `listitem`, `msgexplan`, `msgtext`, `partintro`, `preface`, `procedure`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `simplesect`, `step`.

Children

The following elements occur in productionset: `production`, `productionrecap`, `title`, `titleabbrev`.

Examples

```
<!DOCTYPE simplesect PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<simplesect><title>EBNF Grammar</title>

<productionset><title>Arithemetic Expressions</title>
<production id="ebnf.expression">
  <lhs>Expression</lhs>
  <rhs><nonterminal def="ebnf.arith">ArithExpression</nonterminal> |
       <nonterminal def="ebnf.mult">MultExpression</nonterminal>
  <lineannotation>Does this grammar actually get precedence right?
  </lineannotation>
  </rhs>
```

```
</production>
<production id="ebnf.arith">
  <lhs>ArithExpression</lhs>
  <rhs><nonterminal def="ebnf.expression">Expression</nonterminal>
       '+'
       <nonterminal def="ebnf.mult">MultExpression</nonterminal> |
       <nonterminal def="ebnf.expression">Expression</nonterminal>
       '-'
       <nonterminal def="ebnf.mult">MultExpression</nonterminal>
  </rhs>
</production>
<production id="ebnf.mult">
  <lhs>MultExpression</lhs>
  <rhs><nonterminal def="ebnf.mult">MultExpression</nonterminal>
       '*'
       <nonterminal def="ebnf.mult">MultExpression</nonterminal> |
       <nonterminal def="ebnf.mult">MultExpression</nonterminal>
       '/'
       <nonterminal def="ebnf.mult">MultExpression</nonterminal> |
       <nonterminal def="ebnf.number">Number</nonterminal>
  </rhs>
  <constraint linkend="div0"/>
</production>
<production id="ebnf.number">
  <lhs>Number</lhs>
  <rhs>[0-9]+</rhs>
</production>
</productionset>

<constraintdef id="div0">
<title>Division by Zero</title>
<para>Division by zero is an error.</para>
</constraintdef>
</simplesect>
```

■ productname

Name

productname – The formal name of a product

Synopsis

Mixed Content Model

productname ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|
citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|
trademark|wordasword|personname|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|constant|
markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|
property| replaceable|returnvalue|sgmltag|structfield|structname|symbol|
systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials|
corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|
subscript| superscript|inlinegraphic|inlinemediaobject|inlineequation| synopsis|

cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis| constructorsynopsis| destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

class (copyright | registered | service | trade) "trade"

Parameter Entities

%bibliocomponent.mix; %docinfo.char.class; %info.class;

%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix;

Description

A ProductName is the formal name of any product. Identifying a product this way may be useful if you need to provide explicit disclaimers about product names or information.

For example, the copyright statement on this book includes the following general notice:

> Some of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly & Associates, Inc., was aware of the trademark claim, the designations have been printed in caps or initial caps.

or words to that effect. If every product name in this book had been diligently coded as a ProductName, we could have automatically generated a complete list of all the product names and mentioned them explicitly in the notice.

In running prose, the distinction between an Application and a ProductName may be very subjective.

Processing expectations

Formatted inline.

Two of the values of the Class attribute on ProductName, Trade and Registered, make assertions about trademarks. DocBook also has a TradeMark element; presumably the same markup is intended regardless of which one is used.

The Service and Copyright values should also generate the anticipated marks, if appropriate.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain productname: appendixinfo, application, articleinfo, attribution, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayou lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in productname: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

class Class indicates the type of ProductName.

See Also

application, copyright, database, filename, hardware, medialabel, trademark.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
<productname class='trade'>Frobozz</productname>: it's not
just for breakfast anymore.
</para>
```

Frobozz: it's not just for breakfast anymore. As noted above, the TradeMark element could also be used:

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
<trademark>Frobozz</trademark>: it's not
just for breakfast anymore.
</para>
```

Frobozz™: it's not just for breakfast anymore.

■ productnumber

Name

productnumber – A number assigned to a product

Synopsis

Mixed Content Model

productnumber ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %docinfo.char.class; %info.class;
%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;
%tbl.entry.mdl; %title.char.mix;

Description

An ProductNumber identifies a "product number" in some unspecified numbering scheme. It's possible that product numbers for different products might not even come from the same scheme.

Processing expectations

Formatted inline. Sometimes suppressed.

DocBook does not control, or specify, the numbering scheme used for products.

Parents

These elements contain productnumber: appendixinfo, application, articleinfo, attribution, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayou lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in productnumber: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

biblioid, invpartnumber, isbn, issn, issuenum, pubsnumber, seriesvolnums, volumenum.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
You can order <citetitle>DocBook: The Definitive Guide</citetitle> directly
from O'Reilly &amp;amp; Associates.  Order product number
<productnumber>5807</productnumber> by phone or
<ulink url="http://www.oreilly.com/">over the web</ulink>.
</para>
```

You can order *DocBook: The Definitive Guide* directly from O'Reilly & Associates. Order product number 5807 by phone or over the web http://www.oreilly.com/.

■ programlisting

Name

programlisting – A literal listing of all or part of a program

Synopsis

Mixed Content Model

programlisting ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage|co|coref|lineannotation|textobject)*

Attributes Common attributes

Name Type Default

width CDATA *None*

format linespecific "linespecific"

linenumbering numbered | unnumbered

None

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%divcomponent.mix; %example.mix; %figure.mix;

%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;

%legalnotice.mix; %linespecific.class; %listpreamble.mix;

%para.mix; %qandaset.mix; %refcomponent.mix;

%revdescription.mix; %sidebar.mix; %tabentry.mix;

%tbl.entry.mdl; %textobject.mix;

Description

A ProgramListing is a verbatim environment for program source or source fragment listings. ProgramListings are often placed in Examples or Figures so that they can be cross-referenced from the text.

Processing Expectations

Formatted as a displayed block. This element is displayed “verbatim”; whitespace and linebreaks within this element are significant. ProgramListings are usually displayed in a fixed width font.

Other markup within a ProgramListing is recognized. Contrast this with systems like LaTeX, in which verbatim environments disable markup recognition. If you want to disable markup recognition, you must use a CDATA section:

```
<programlisting>
<![CDATA[
This is a programlisting so white       space and line
breaks are significant.  But it is also a CDATA
section so <emphasis>tags</emphasis> and &entities;
are not recognized.  The only markup that is recognized
```

```
is the end-of-section marker, which is two
"]"'s in a row followed by a >.

]]>

</programlisting>
```

Two markup tags have special significance in ProgramListings: CO and LineAnnotation. A CO identifies the location of a Callout. A LineAnnotation is a comment, added by the *documentor*—not the programmer.

Processing expectations

This element is displayed "verbatim"; whitespace and linebreaks within this element are significant.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Future Changes

The xml:space attribute is automatically provided in the XML DTD.

Parents

These elements contain programlisting: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, programlistingco, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in programlisting: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, co, command, computeroutput, constant, constructorsynopsis, coref, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, lineannotation, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, textobject, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

format The Format attribute applies the linespecific notation to all ProgramListings. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of a `ProgramListing` are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

width Width specifies the width (in characters) of the longest line in this `ProgramListing` (formatters may use this value to determine scaling or rotation).

See Also

`computeroutput`, `lineannotation`, `literallayout`, `screen`, `screenshot`, `synopsis`, `userinput`.

Examples

For examples, see `example`, `informalexample`, `programlistingco`, `refentry`.

■ programlistingco

Name

programlistingco – A program listing with associated areas used in callouts

Synopsis

Content Model

programlistingco ::= (`areaspec`,`programlisting`,`calloutlist`*)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

`%divcomponent.mix;` `%example.mix;` `%figure.mix;`

`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`

`%legalnotice.mix;` `%linespecific.class;` `%listpreamble.mix;`

`%para.mix;` `%qandaset.mix;` `%refcomponent.mix;`

`%revdescription.mix;` `%sidebar.mix;` `%tabentry.mix;`

`%tbl.entry.mdl;` `%textobject.mix;`

Description

Callouts, such as numbered bullets, are an annotation mechanism. In an online system, these bullets are frequently "hot," and clicking on them sends you to the corresponding annotation.

A `ProgramListingCO` is a wrapper around an `AreaSpec` and a `ProgramListing`. An `AreaSpec` identifies the locations (coordinates) in the `ProgramListing` where the callouts occur. The `ProgramListingCO` may also contain the list of annotations in a `CalloutList`, although the `CalloutList` may also occur outside of the wrapper, elsewhere in the document.

It is also possible to embed `CO` elements directly in the verbatim text, in order to avoid having to calculate the correct coordinates. If you decided to go this route, use a `ProgramListing` and a `CalloutList` without the `ProgramListingCO` wrapper. A `ProgramListingCO` must specify at least one coordinate.

For a complete description of callouts, see `Callout`.

Processing expectations

Formatted as a displayed block. This element is displayed “verbatim”; whitespace and linebreaks within this element are significant.

The mandatory processing expectations of a `ProgramListingCO` are minimal: a system is expected to render the program listing and the callout list, if present.

If explicit `CO` elements are embedded in a `ProgramListing`, they must generate appropriate callout marks.

In online environments, the processing system may be able to instantiate the linking relationships between the callout marks in the program listing and the annotations. Some systems may even be able to go a step further and generate the callout marks automatically from the coordinate information, but this level of sophistication is not mandatory.

Parents

These elements contain programlistingco: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `example`, `figure`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `indexdiv`, `informalexample`, `informalfigure`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `variablelist`, `warning`.

Children

The following elements occur in programlistingco: `areaspec`, `calloutlist`, `programlisting`.

See Also

`areaspec`, `calloutlist`, `co`, `coref`, `graphicco`, `imageobjectco`, `mediaobjectco`, `screenco`.

Examples

```
<!DOCTYPE programlistingco PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<programlistingco>
<areaspec>
<areaset id="ex.plco.const" coords="">
  <area id="ex.plco.c1" coords='4'/>
  <area id="ex.plco.c2" coords='8'/>
</areaset>
<area id="ex.plco.ret" coords='12'/>
<area id="ex.plco.dest" coords='12'/>
</areaspec>
<programlisting>
sub do_nothing_useful {
    my($a, $b, $c);

    $a = new A;

    $a->does_nothing_either();

    $b = new B;

    $c = "frog";

    return ($a, $c);
}
</programlisting>
<calloutlist>
```

```
<callout arearefs="ex.plco.const">
<para>
These are calls to the constructor <function>new</function> in the object
classes.
</para>
</callout>
<callout arearefs="ex.plco.ret">
<para>
This function returns a two-element list.
</para>
</callout>
<callout arearefs="ex.plco.dest">
<para>
The <emphasis>destructor</emphasis> (<function>DESTROY</function>) for
the object <literal>$b</literal> will be called automatically for this
object since there can be no other references to it outside this function.
</para>
</callout>
</calloutlist>
</programlistingco>


sub do_nothing_useful {
    my($a, $b, $c);

    $a = new A;

    $a->does_nothing_either();

    $b = new B;

    $c = "frog";

    return ($a, $c);
}
```

callout ??? These are calls to the constructor `new` in the object classes. callout ??? This function returns a two-element list. callout ??? The *destructor* (`DESTROY`) for the object `$b` will be called automatically for this object since there can be no other references to it outside this function.

■ prompt

Name

prompt – A character or string indicating the start of an input field in a computer display

Synopsis

Mixed Content Model

prompt ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`|`co`)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;
%tech.char.class; %title.char.mix;

Description

A Prompt is a character or character string marking the beginning of an input field. Prompts are generally associated with command-line interfaces and not graphical user interfaces (GUIs). In GUIs, GUILabel is usually more appropriate.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

Prompt was added in DocBook V3.0. It duplicates the semantics of <systemitem class="prompt">. The prompt attribute will be removed from SystemItem in the future.

Parents

These elements contain prompt: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in prompt: beginpage, co, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Prompt.

See Also

computeroutput, constant, envar, filename, literal, markup, option, optional, parameter, replaceable, sgmltag, systemitem, userinput, varname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Enter your user name when the system presents the
<prompt>login:</prompt> prompt.
</para>
```

Enter your user name when the system presents the login: prompt.

property

Name

property – A unit of data associated with some part of a computer system

Synopsis

Mixed Content Model

property ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

moreinfo none | refentry "none"

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

The notion of a Property is very domain-dependent in computer documentation. Some object-oriented systems speak of properties; the components from which GUIs are constructed have properties; and one can speak of properties in very general terms; “the properties of a relational database.”

You might use Property for any of these in your documentation.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

The content model of Property will be constrained to (#PCDATA | Replaceable | InlineGraphic) in DocBook V4.0.

Parents

These elements contain property: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in property: action, anchor, application, beginpage, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the Property.

See Also

classname, interface, structfield, structname, symbol, token, type.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
When Emacs is running under X Windows, the <property>borderWidth</property>
resource controls the width of the external border.
</para>
```

When Emacs is running under X Windows, the borderWidth resource controls the width of the external border.

■ pubdate

Name

pubdate – The date of publication of a document

Synopsis

Mixed Content Model

pubdate ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

The PubDate is the date of publication of a document.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain pubdate: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in pubdate: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

date, edition, printhistory, releaseinfo, revhistory.

Examples

For examples, see article, bibliography, bibliomset, bookinfo.

■ publisher

Name

publisher – The publisher of a document

Synopsis

Content Model

publisher ::= (publishername,address*)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class;

Description

Publisher associates a PublisherName and an Address. Many publishers have offices in more than one city. Publisher can be used to list or distinguish between the multiple offices.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain publisher: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in publisher: address, publishername.

Examples

```
<!DOCTYPE publisher PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<publisher>
  <publishername>O'Reilly &amp; Associates, Inc.</publishername>
  <address><street>101 ...</street>
  ...
  </address>
</publisher>
```

For additional examples, see also `article`, `bibliography`, `biblioset`.

■ publishername

Name

publishername – The name of the publisher of a document

Synopsis

Mixed Content Model

publishername ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

A `PublisherName` is the name of a publisher. Historically, this has been used in bibliographic meta-information to identify the publisher of a book or other document. It is also reasonable to identify the publisher of an electronic publication in this way.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain publishername: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `publisher`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in publishername: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`corpname`, `orgname`.

Examples

For examples, see `article`, `bibliography`, `bibliomset`, `biblioset`, `publisher`.

■ pubsnumber

Name

pubsnumber – A number assigned to a publication other than an ISBN or ISSN or inventory part number

Synopsis

Mixed Content Model

pubsnumber ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

A `PubsNumber` identifies a document in some unspecified numbering scheme. This number may exist instead of, or in addition to, an `ISBN` or `ISSN` number.

Processing expectations

Formatted inline. Sometimes suppressed.

DocBook does not control, or specify, the numbering scheme used for documents.

Deprecated as of DocBook V4.2, see `biblioid` instead.

Parents

These elements contain pubsnumber: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in pubsnumber: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`biblioid`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `productnumber`, `seriesvolnums`, `volumenum`.

Examples

For examples, see `contractsponsor`, `othercredit`.

■ qandadiv

Name

qandadiv – A titled division in a `QandASet`

Synopsis

Content Model

qandadiv ::= (blockinfo?, (title,titleabbrev?)?, (calloutlist|glosslist|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)*, (qandadiv+|qandaentry+))

Attributes Common attributes

Description

QandADiv is a section of a QandASet. A question and answer set might be divided into sections in order to group different sets of questions together, perhaps by topic.

A QandASet may contain any number of QandADiv or QandAEntry elements, but it cannot contain a mixture of both at the same level.

Processing expectations

Formatted as a displayed block.

A table of contents for the question and answer set is sometimes generated, especially in online environments.

Parents

These elements contain qandadiv: qandadiv, qandaset.

Children

The following elements occur in qandadiv: address, anchor, blockinfo, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigur informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, note, orderedlist, para, procedure, programlisting, programlistingco, qandadiv, qandaentry, remark, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Examples

For examples, see qandaset.

■ qandaentry

Name

qandaentry – A question/answer set within a QandASet

Synopsis

Content Model

qandaentry ::= (blockinfo?,revhistory?,question,answer*)

Attributes Common attributes

Description

A QandAEntry is an entry in a QandASet. Each QandAEntry defines a Question and (possibly) its Answer or Answers.

Since V4.2, the preferred way to associate a revision history with a QandAEntry is in the BlockInfo. The use of RevHistory directly in QandAEntry is deprecated.

Processing expectations

Formatted as a displayed block. Questions are usually presented before the Answers, and often the Answers are indented to make the questions stand out.

Parents

These elements contain qandaentry: answer, qandadiv, qandaset.

Children

The following elements occur in qandaentry: answer, blockinfo, question, revhistory.

Examples

For examples, see qandaset.

■ qandaset

Name

qandaset – A question-and-answer set

Synopsis

Content Model

qandaset ::= (blockinfo?, (title,titleabbrev?)?, (calloutlist|glosslist|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)*, (qandadiv+|qandaentry+))

Attributes Common attributes

Name Type Default

defaultlabel	none number qanda	*None*

Parameter Entities

%bookcomponent.content; %component.mix; %compound.class;

%divcomponent.mix; %refcomponent.mix;

Description

A QandASet is a list consisting of Questions and Answers. QandASets can be divided into sections.

Every entry in a QandASet must contain a Question, but Answers are optional (some questions have no answers), and may be repeated (some questions have more than one answer).

Common uses for QandASets include reader questionnaires and lists of "Frequently Asked Questions" (FAQs). For the purpose of an FAQ, DocBook V3.1 added the FAQ class to Article.

Processing expectations

Formatted as a displayed block. The DefaultLabel attribute has a significant influence on the presentation of Questions and Answers.

Parents

These elements contain qandaset: appendix, article, bibliodiv, bibliography, blockquote, callout, chapter, glossary, glossdiv, index, listitem, msgexplan, msgtext, partintro, preface, procedure, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, sect1, sect2, sect3, sect4, sect5, section, setindex, simplesect, step.

Children

The following elements occur in qandaset: address, anchor, blockinfo, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigur informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, note, orderedlist, para, procedure, programlisting, programlistingco, qandadiv, qandaentry, remark, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

defaultlabel DefaultLabel identifies the default label that should be used for Questions and Answers:

qanda Questions are labeled "Q:" and Answers are labeled "A:". Other similar labels may be substituted, for example, the words might be spelled out, "Question:" and "Answer:", and the actual characters or words used are dependent on the language.

number The entries are enumerated.

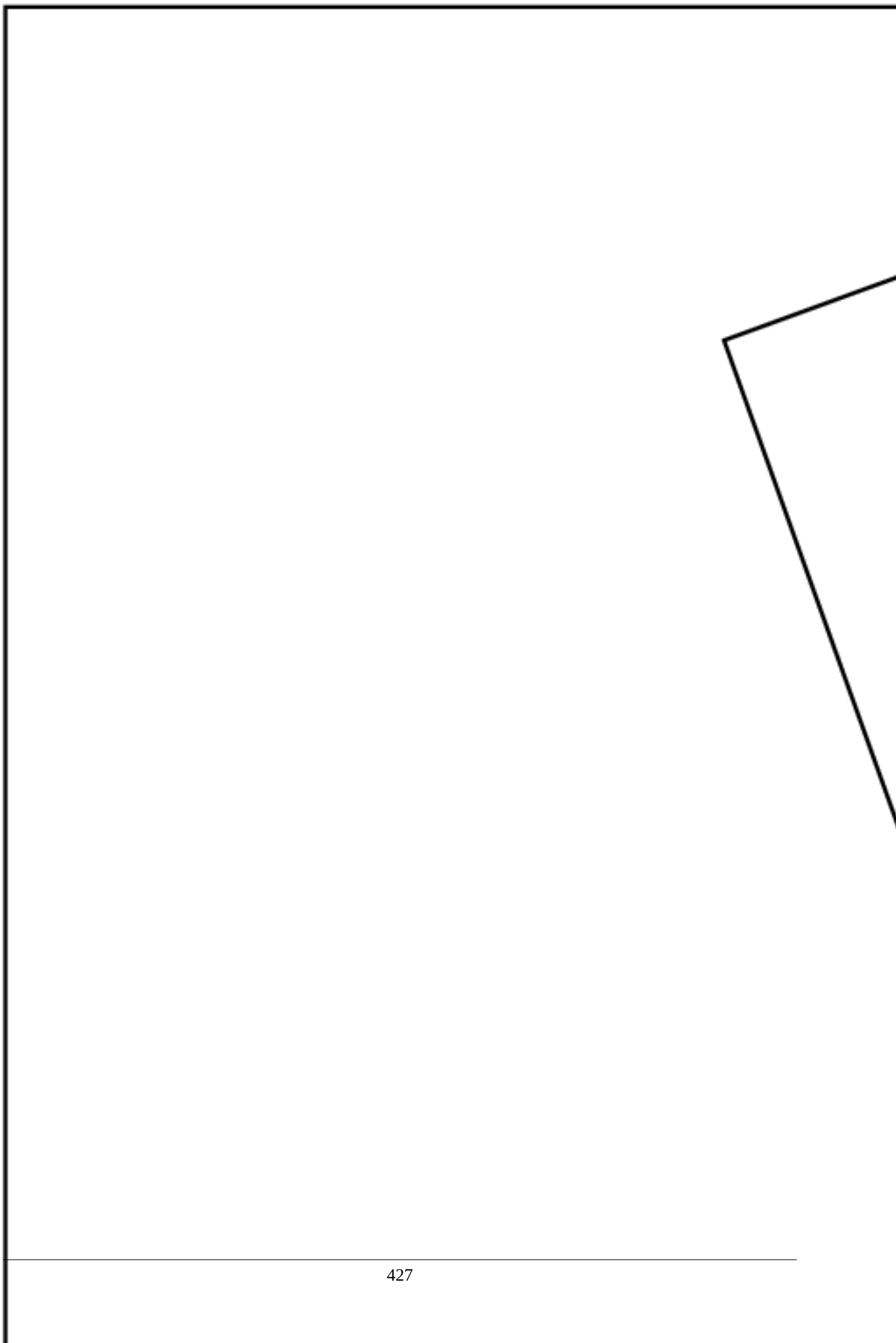
none No distinguishing label precedes Questions or Answers.

If no value is specified, the implied presentation may be any one of these, as defined by the stylesheet. Note that each question and answer can explicitly define a label, regardless of the default label specified.

Examples

```
<!DOCTYPE qandaset PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<qandaset defaultlabel='qanda'>
<qandaentry>
<question>
<para>
To be, or not to be?
</para>
</question>
```

```
<answer>
<para>
That is the question.
</para>
</answer>
</qandaentry>
</qandaset>
```



F.A.Q.

1. **Q:** *To be, or not to be?*
A: That is the question.

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<article class="faq">
<title>Frequently Asked Questions About Fonts</title>

<para>...</para>

<qandaset>
<qandadiv><title>General Information</title>

<para>...</para>

<qandadiv><title>Font Houses</title>

<qandaentry><question><para>Adobe Systems, Inc.</para></question>
<answer><para>...</para></answer>
</qandaentry>

<qandaentry><question><para>Agfa, Inc.</para></question>
<answer><para>...</para></answer>
</qandaentry>

</qandadiv>
</qandadiv>
</qandaset>
</article>
```

■ question

Name

question – A question in a `QandASet`

Synopsis

Content Model

question ::= (`label`?, (`calloutlist`|`glosslist`|`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)+)

Attributes Common attributes

Description

A Question in a QandAEntry poses a question or states a problem that is addressed by the following Answer(s). Answers are optional (some questions have no answers) and may be repeated (some questions have more than one answer).

Processing expectations

Questions are frequently introduced with a label, such as "A:". If a Question has a Label child element, the content of that Label is used as the label for the Question. The DefaultLabel attribute on the nearest ancestor QandASet of a Question can be used to indicate that a processing application should automatically generate a label for the Question.

Parents

These elements contain question: qandaentry.

Children

The following elements occur in question: address, anchor, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, label, literallayout, mediaobject, mediaobjectco, methodsynopsis, note, orderedlist, para, procedure, programlisting, programlistingco, remark, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, table, tip, variablelist, warning.

Examples

For examples, see qandaset.

■ quote

Name

quote – An inline quotation

Synopsis

Mixed Content Model

quote ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

`%gen.char.class;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%tbl.entry.mdl;` `%title.char.mix;`

Description

`Quote` surrounds an inline quotation. Using an element for quotations is frequently more convenient than entering the character entities for the quotation marks by hand, and makes it possible for a presentation system to alter the format of the quotation marks.

Block quotations are properly identified as `BlockQuote`s.

Processing expectations

Formatted inline. The `Quote` element is expected to generate the proper quotation marks. These may be influenced by the Lang attribute on an ancestor element. For example, a quote in French might use «guillments» instead of English “quote marks.”

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain quote: `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `para`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `quote`, `refentrytitle`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in quote: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

See Also

`abbrev`, `acronym`, `emphasis`, `foreignphrase`, `phrase`, `wordasword`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
This software is provided <quote>as is</quote>, without expressed
or implied warranty.
</para>
```

This software is provided “as is”, without expressed or implied warranty. For additional examples, see also `bookinfo`, `link`.

■ refclass

Name

refclass – The scope or other indication of applicability of a reference entry

Synopsis

Mixed Content Model

refclass ::= (#PCDATA|`application`)*

Attributes Common attributes

Description

The `RefClass` element describes the applicability or scope of a `RefEntry`. A `RefClass` might indicate that the entry was only applicable to a particular application, for example, or only to a particular vendor’s operating system.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running heads, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Parents

These elements contain refclass: `refnamediv`.

Children

The following elements occur in refclass: `application`.

■ refdescriptor

Name

refdescriptor – A description of the topic of a reference page

Synopsis

Mixed Content Model

refdescriptor ::= (#PCDATA|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal)*

Attributes Common attributes

Description

Reference pages (RefEntrys) are usually identified by a short, succinct topic name, such as the name of a function or command. The RefName (or one of the RefNames, in the case of a reference page that has several) is generally used as the topic name. When none of the RefNames is appropriate, RefDescriptor is used to specify the topic name.

RefDescriptor is unnecessary when an appropriate RefName can be selected automatically. At least one RefName is required, so RefDescriptor cannot be used in place of a name, only in addition to it.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the RefEntry.

Common presentational features, such as titles and running heads, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

If a RefDescriptor is present, it should be used for the short topic name. This name usually appears in the running header along with the ManVolNum in print media. It may also appear in tables of contents and the index.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain refdescriptor: refnamediv.

Children

The following elements occur in refdescriptor: action, application, classname, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, interface, interfacename, keycap, keycode, keycombo, keysym, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, replaceable, returnvalue, sgmltag, structfield, structname, symbol, systemitem, token, type, userinput, varname.

See Also

refentrytitle, refname.

Examples

For examples, see reference.

■ refentry

Name

refentry – A reference page (originally a UNIX man-style reference page)

Synopsis

Content Model

refentry ::= (`beginpage`?, (`indexterm`)*, `refentryinfo`?,`refmeta`?, (`remark`|`link`|`olink`|`ulink`)*, `refnamediv`,`refsynopsisdiv`?, (`refsect1`+|`refsection`+))

Attributes Common attributes

Name Type Default

status CDATA *None*

Parameter Entities

`%bookcomponent.content;` `%partcontent.mix;` `%refentry.class;`

Description

A `RefEntry` is a reference page. In UNIX parlance this has historically been called a “man page” (short for manual page).

`RefEntry` is an appropriate wrapper for any small unit of reference documentation describing a single topic. Canonical examples are programming language functions and user commands (one `RefEntry` per function or command).[5]

On some projects, the structure of reference pages may be rigorously defined right down to the number, order, and title of individual sections (some or all of which may be required).

Processing expectations

Formatted as a displayed block. It is not uncommon for `RefEntry`s to introduce a forced page break in print media.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running heads, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Parents

These elements contain refentry: `appendix`, `article`, `chapter`, `part`, `partintro`, `preface`, `reference`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`.

Children

The following elements occur in refentry: `beginpage`, `indexterm`, `link`, `olink`, `refentryinfo`, `refmeta`, `refnamediv`, `refsect1`, `refsection`, `refsynopsisdiv`, `remark`, `ulink`.

Attributes

status Status identifies the editorial or publication status of the `RefEntry`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

[5] You’re reading a `RefEntry` right now.

Examples

A typical reference page for a command:

```
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<refentry id="ls">

<refmeta>
<refentrytitle>ls</refentrytitle>
<manvolnum>1</manvolnum>
</refmeta>

<refnamediv>
<refname>ls</refname>
<refpurpose>list contents of a directory</refpurpose>
</refnamediv>

<refsynopsisdiv>
<cmdsynopsis>
<command>/usr/bin/ls</command>
<arg choice="opt">
  <option>aAbcCdfFgilLmnopqrRstux1</option>
</arg>
<arg choice="opt" rep="repeat">file</arg>
</cmdsynopsis>
</refsynopsisdiv>

<refsect1><title>Description</title>
<para>
For each file that is a directory, <command>ls</command> lists the contents of
the directory; for each file that is an ordinary file, <command>ls</command>
repeats its name and any other information requested.
</para>
<para>&hellip;</para>
</refsect1>
</refentry>
```

A typical reference page for a function:

```
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<refentry id="printf">

<refmeta>
<refentrytitle>printf</refentrytitle>
<manvolnum>3S</manvolnum>
</refmeta>

<refnamediv>
<refname>printf</refname>
<refname>fprintf</refname>
<refname>sprintf</refname>
<refpurpose>print formatted output</refpurpose>
</refnamediv>

<refsynopsisdiv>

<funcsynopsis>
<funcsynopsisinfo>
#include &lt;stdio.h&gt;
```

```
</funcsynopsisinfo>
<funcprototype>
  <funcdef>int <function>printf</function></funcdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>

<funcprototype>
  <funcdef>int <function>fprintf</function></funcdef>
  <paramdef>FILE *<parameter>strm</parameter></paramdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>

<funcprototype>
  <funcdef>int <function>sprintf</function></funcdef>
  <paramdef>char *<parameter>s</parameter></paramdef>
  <paramdef>const char *<parameter>format</parameter></paramdef>
  <paramdef>...</paramdef>
</funcprototype>
</funcsynopsis>

</refsynopsisdiv>

<refsect1><title>Description</title>
<para>
<function>printf</function> places output on the standard
output stream stdout.
</para>
<para>&hellip;</para>
</refsect1>
</refentry>
```

A reference page for a data structure:

```
<!DOCTYPE refentry PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<refentry id="iovec">

<refmeta>
<refentrytitle>iovec</refentrytitle>
<manvolnum>9S</manvolnum>
</refmeta>

<refnamediv>
<refname>iovec</refname>
<refpurpose>data storage structure for I/O using uio</refpurpose>
</refnamediv>

<refsynopsisdiv>
<synopsis>
#include &lt;sys/uio.h&gt;
</synopsis>
</refsynopsisdiv>

<refsect1><title>Interface Level</title>
<para>
Architecture independent level 1 (DDI/DKI).
</para>
</refsect1>
```

```
<refsect1><title>Description</title>

<para>
An <structname>iovec</structname> structure describes a data
storage area for transfer in a
<citerefentry><refentrytitle>uio</refentrytitle>
  <manvolnum>9S</manvolnum>
</citerefentry>
structure. Conceptually,
it may be thought of as a base address and length specification.
</para>

</refsect1>
<refsect1><title>Structure Members</title>

<programlisting>
     caddr_t  iov_base;  /* base address of the data storage area */
                         /* represented by the iovec structure */
     int      iov_len;   /* size of the data storage area in bytes */
</programlisting>

<para>&hellip;</para>
</refsect1>
</refentry>
```

For additional examples, see also `reference`.

■ refentryinfo

Name

refentryinfo – Meta-information for a `Refentry`

Synopsis

Content Model

refentryinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblierelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `RefentryInfo` element is a wrapper for a large collection of meta-information about a `Refentry`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain refentryinfo: `refentry`.

Children

The following elements occur in refentryinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

■ refentrytitle

Name

refentrytitle – The title of a reference page

Synopsis

Mixed Content Model

refentrytitle ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|
`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|
`trademark`|`wordasword`|`personname`|`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`|`constant`|
`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|
`property`| `replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|
`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`| `author`|`authorinitials`|
`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|
`subscript`| `superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`| `synopsis`|
`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`| `constructorsynopsis`|
`destructorsynopsis`|`methodsynopsis`|`indexterm`| `beginpage`)*

Attributes Common attributes

Description

A `RefEntryTitle` is the title of a reference page. It is frequently the same as the first `RefName` or the `RefDescriptor`, although it may also be a longer, more general title.

Processing expectations

Formatted as a displayed block.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running heads, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain refentrytitle: citerefentry, refmeta.

Children

The following elements occur in refentrytitle: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

refdescriptor, refname.

Examples

For examples, see citerefentry, manvolnum, refentry.

■ reference

Name

reference – A collection of reference entries

Synopsis

Content Model

reference ::= (beginpage?,referenceinfo?, (title,subtitle?,titleabbrev?), partintro?, (refentry)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

label CDATA *None*

Parameter Entities

%partcontent.mix;

Description

A Reference is a collection of RefEntrys. In a Book, a Reference can occur at either the Part or Chapter level.

Reference pages are usually bound together by topic; in traditional UNIX documentation they are most frequently bound into volumes. See ManVolNum.

Processing expectations

Formatted as a displayed block. Referencess often introduce a forced page break and may start on the next recto page. Frequently, they also produce a separator page, on which may be printed the content of the PartIntro.

Parents

These elements contain reference: book, part.

Children

The following elements occur in reference: beginpage, partintro, refentry, referenceinfo, subtitle, title, titleabbrev.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the Reference.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

Examples

```
<!DOCTYPE reference PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">

<reference><title>Reference Pages</title>

<refentry>
<refnamediv>
<refdescriptor>GNU as</refdescriptor>
<refname>as</refname>
<refpurpose>the portable GNU assembler</refpurpose>
</refnamediv>
<refsynopsisdiv><title>SYNOPSIS</title>
<synopsis>...</synopsis>
</refsynopsisdiv>
<refsect1><title>DESCRIPTION</title>
<para>...</para>
</refsect1>
</refentry>

<refentry>
<refnamediv>
<refname>awk</refname>
<refpurpose>pattern scanning and text processing language</refpurpose>
</refnamediv>
<refsynopsisdiv><title>SYNOPSIS</title>
<synopsis>...</synopsis>
</refsynopsisdiv>
<refsect1><title>DESCRIPTION</title>
<para>...</para>
</refsect1>
</refentry>

<!-- ... -->
```

```
</reference>
```

■ referenceinfo

Name

referenceinfo – Meta-information for a `Reference`

Synopsis

Content Model

referenceinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblioreleation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The `ReferenceInfo` element is a wrapper for a large collection of meta-information about a `Reference`. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain referenceinfo: `reference`.

Children

The following elements occur in referenceinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `biblioreleation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

■ refmeta

Name

refmeta – Meta-information for a reference entry

Synopsis

Content Model

refmeta ::= ((indexterm)*, refentrytitle,manvolnum?,refmiscinfo*, (indexterm)*)

Attributes Common attributes

Description

RefMeta holds the title of the reference page, the number of the volume in which this reference page occurs, and possibly other miscellaneous information (typically used in printing the reference page).

Processing expectations

Suppressed. Most of the elements contained in RefMeta are used in presentation, but they are not generally printed as part of the formatting of the RefMeta wrapper–it merely serves to identify where they occur.

Parents

These elements contain refmeta: refentry.

Children

The following elements occur in refmeta: indexterm, manvolnum, refentrytitle, refmiscinfo.

Examples

For examples, see refentry.

■ refmiscinfo

Name

refmiscinfo – Meta-information for a reference entry other than the title and volume number

Synopsis

Mixed Content Model

refmiscinfo ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Name Type Default

class CDATA *None*

Description

RefMiscInfo is an escape hatch for additional meta-information about a reference page. It may hold copyright information, release or revision information, descriptive text for use in a print header or footer, or any other information not explicitly provided for in RefMeta.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

Parents

These elements contain refmiscinfo: refmeta.

Children

The following elements occur in refmiscinfo: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

Attributes

class Class on RefMiscInfo can be used to indicate the nature of the miscellaneous information being added to RefMeta. Naming this attribute "class" is a violation of DocBook semantics (where Class attributes have delimited value sets), it should really be called Type.

■ refname

Name

refname – The name of (one of) the subject(s) of a reference page

Synopsis

Mixed Content Model

refname ::= (#PCDATA|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal)*

Attributes Common attributes

Description

RefEntrys are small units of reference documentation describing a single topic. The RefName identifies the topic. Often this is the name of the command or function that the reference page describes.

Some reference pages describe a whole family of very closely related commands or functions. In this case, a RefEntry will have multiple RefNames, one for each command or function. When a RefEntry has several RefNames, it is likely to have a RefDescriptor that identifies the whole family of functions.

Processing expectations

May be formatted inline or as a displayed block, depending on context.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the RefEntry.

Common presentational features, such as titles and running headers, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain refname: refnamediv.

Children

The following elements occur in refname: `action`, `application`, `classname`, `command`, `computeroutput`, `constant`, `database`, `email`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `function`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `mousebutton`, `nonterminal`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `parameter`, `prompt`, `property`, `replaceable`, `returnvalue`, `sgmltag`, `structfield`, `structname`, `symbol`, `systemitem`, `token`, `type`, `userinput`, `varname`.

See Also

`refdescriptor`, `refentrytitle`.

Examples

For examples, see `refentry`, `reference`.

■ refnamediv

Name

refnamediv – The name, purpose, and classification of a reference page

Synopsis

Content Model

refnamediv ::= (`refdescriptor`?,`refname`+,`refpurpose`,`refclass`*, (`remark`|`link`|`olink`|`ulink`)*)

Attributes Common attributes

Description

`RefNameDiv` is the first mandatory section in a `RefEntry`. It is a peer to `RefSynopsisDiv` and `RefSect1`.

The elements in `RefNameDiv` identify the topic of the reference page (`RefDescriptor` or `RefName`), provide a concise summary (`RefPurpose`), and classify the page (`RefClass`).

Processing expectations

Formatted as a displayed block. `RefNameDiv` usually generates a section heading, in the same typographic style as a `RefSect1` `Title`, called “Name.”

The content of this section is traditionally the `RefDescriptor` or `RefName`, and the `RefPurpose`, separated by an em dash.

The `RefClass` may be presented, or it may be suppressed and used only to select a group of reference pages to process. You might use the value of `RefClass` to print all the reference pages appropriate to Solaris UNIX, for example.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running headers, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Parents

These elements contain refnamediv: `refentry`.

Children

The following elements occur in refnamediv: link, olink, refclass, refdescriptor, refname, refpurpose, remark, ulink.

See Also

refsect1, refsect2, refsect3, refsection, refsynopsisdiv.

Examples

For examples, see refentry, reference.

■ refpurpose

Name

refpurpose – A short (one sentence) synopsis of the topic of a reference page

Synopsis

Mixed Content Model

refpurpose ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|indexterm|beginpage)*

Attributes Common attributes

Description

The RefPurpose is a concise summary of the topic of the reference page. A RefPurpose is usually limited to a single, short sentence.

Processing expectations

Formatted inline. See RefNameDiv.

In a large Reference, RefNames and RefPurposes are sometimes used to construct a permuted index. A permuted index is a keyword-in-context concordance of lines, like the short definitions in this element reference; the keyword cycles alphabetically through the words of the (definition) lines.

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the RefEntry.

Common presentational features, such as titles and running headers, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain refpurpose: refnamediv.

Children

The following elements occur in refpurpose: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Examples

For examples, see refentry, reference.

■ refsect1

Name

refsect1 – A major subsection of a reference entry

Synopsis

Content Model

refsect1 ::= (refsect1info?, (title,subtitle?,titleabbrev?), (((calloutlist|glosslist| 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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)+, refsect2*)| refsect2+))

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

Reference pages have their own hierarchical structure. A RefSect1 is a major division in a RefEntry, analagous to a Sect1 elsewhere in the document.

The value of a separate hierarchical structure is that it allows the content model of sections in reference pages to be customized differently than the content model of sections outside. For example, because of this split, it was easy to add a recursive sectioning element (Section) as a peer to Sect1 in DocBook V3.1 without introducing it to RefEntrys, in which it would not be desirable.

Processing expectations

Formatted as a displayed block.

In some environments, the name, number, and order of major divisions in a reference page is strictly defined by house style. For example, one style requires that the first major section after the synopsis be the "Description," which it must have as its title.

In those cases, it may be useful to replace `RefSect1` in the content model with a set of named sections (following the pattern of `RefNameDiv` and `RefSynopsisDiv`).

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running headers, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Parents

These elements contain refsect1: `refentry`.

Children

The following elements occur in refsect1: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refsect1info`, `refsect2`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `RefSect1`.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

`refnamediv`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`.

Examples

For examples, see `refentry`, `reference`.

■ refsect1info

Name

refsect1info – Meta-information for a `RefSect1`

Synopsis

Content Model

refsect1info ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|

corpauthor|corpname|date| edition|editor|invpartnumber|isbn|issn|issuenum| orgname| biblioid|citebiblioid|bibliosource|bibliorelation| bibliocoverage| othercredit|pagenums|printhistory|productname| productnumber|pubdate|publisher| publishername|pubsnumber| releaseinfo|revhistory|seriesvolnums|subtitle| title|titleabbrev| volumenum|citetitle|personname|honorific|firstname|surname| lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

Like the other "info" elements, RefSect1Info contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Parents

These elements contain refsect1info: refsect1.

Children

The following elements occur in refsect1info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, biblioreleation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

Examples

For examples, see .

■ refsect2

Name

refsect2 – A subsection of a RefSect1

Synopsis

Content Model

refsect2 ::= (refsect2info?, (title,subtitle?,titleabbrev?), (((calloutlist|glosslist| 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|msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)+, refsect3*)| refsect3+))

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

A `RefSect2` is a second level section in a `RefEntry`, analogous to a `Sect2` elsewhere in the document. See `RefSect1`.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain refsect2: `refsect1`, `refsynopsisdiv`.

Children

The following elements occur in refsect2: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refsect2info`, `refsect3`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `RefSect2`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

■ refsect2info

Name

refsect2info – Meta-information for a `RefSect2`

Synopsis

Content Model

refsect2info ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other "info" elements, `RefSect2Info` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `RefSect2Info` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain refsect2info: `refsect2`.

Children

The following elements occur in refsect2info: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

Examples

See `RefSect1Info` in `RefEntry` for an analogous example.

■ refsect3

Name

refsect3 – A subsection of a `RefSect2`

Synopsis

Content Model

refsect3 ::= (`refsect3info`?, (`title`,`subtitle`?,`titleabbrev`?), (`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)+)

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

A RefSect3 is a third level section in a RefEntry, analogous to a Sect3 elsewhere in the document. See RefSect1.

In DocBook, RefSect3 is the lowest-level section allowed in a RefEntry. There is no element analogous to a Sect4.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain refsect3: refsect2.

Children

The following elements occur in refsect3: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refsect3info, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

Attributes

status Status identifies the editorial or publication status of the RefSect3.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

Examples

See RefSect1 and RefSect2 in RefEntry for analogous examples.

■ refsect3info

Name

refsect3info – Meta-information for a RefSect3

Synopsis

Content Model

refsect3info ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

Like the other "info" elements, `RefSect3Info` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `RefSect3Info` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain refsect3info: `refsect3`.

Children

The following elements occur in refsect3info: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

Examples

See `RefSect1Info` in `RefEntry` for an analogous example.

■ refsection

Name

refsection – A recursive section in a `refentry`

Synopsis

Content Model

refsection ::= (`refsectioninfo`?, (`title`,`subtitle`?,`titleabbrev`?), (((`calloutlist`| `glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`| `constraintdef`| `anchor`|`bridgehead`|`remark`|`highlights`|`abstract`| `authorblurb`|`epigraph`|`indexterm`| `beginpage`)+, `refsection`*)| `refsection`+))

Attributes Common attributes

Name Type Default

status CDATA *None*

Description

Reference pages have their own hierarchical structure. A `RefSection` is a recursive division in a `RefEntry`, analagous to a `Section` elsewhere in the document.

The value of a separate hierarchical structure is that it allows the content model of sections in reference pages to be customized differently than the content model of sections outside.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain refsection: `refentry`, `refsection`.

Children

The following elements occur in refsection: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refsection`, `refsectioninfo`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Attributes

status Status identifies the editorial or publication status of the `RefSection`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`refnamediv`, `refsect1`, `refsect2`, `refsect3`, `refsynopsisdiv`.

■ refsectioninfo

Name

refsectioninfo – Meta-information for a `refsection`

Synopsis

Content Model

refsectioninfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other "info" elements, `refsectioninfo` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Parents

These elements contain refsectioninfo: `refsection`.

Children

The following elements occur in refsectioninfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

■ refsynopsisdiv

Name

refsynopsisdiv – A syntactic synopsis of the subject of the reference page

Synopsis

Content Model

refsynopsisdiv ::= (`refsynopsisdivinfo`?, (`title`,`subtitle`?,`titleabbrev`?)?, (((`calloutlist`| `glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`| `constraintdef`| `anchor`|`bridgehead`|`remark`|`highlights`|`abstract`| `authorblurb`|`epigraph`|`indexterm`| `beginpage`)+, `refsect2`*)| (`refsect2`+)))

Attributes Common attributes

Description

`RefSynopsisDiv` contains a syntactic synopsis of the function or command described by the `RefEntry`. When `RefEntry`s are used to describe other sorts of things, `RefSynopsisDiv` should be used for whatever succinct, synopsis information seems appropriate.[6]

[6] In this book, each element of the DTD is described on a reference page, and the `RefSynopsisDiv` is used for the synopsis at the beginning of each entry.

Processing expectations

Formatted as a displayed block. `RefSynopsisDiv` usually generates a section heading, in the same typographic style as a `RefSect1` `Title`, called "Synopsis."

Formatting reference pages may require a fairly sophisticated processing system. Much of the meta-information about a reference page (its name, type, purpose, title, and classification) is stored in wrappers near the beginning of the `RefEntry`.

Common presentational features, such as titles and running headers, may require data from several of these wrappers plus some generated text. Other formatting often requires that these elements be reordered.

Parents

These elements contain refsynopsisdiv: `refentry`.

Children

The following elements occur in refsynopsisdiv: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refsect2`, `refsynopsisdivinfo`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

See Also

`arg`, `cmdsynopsis`, `group`, `refnamediv`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `sbr`, `synopfragment`, `synopfragmentref`.

■ refsynopsisdivinfo

Name

refsynopsisdivinfo – Meta-information for a RefSynopsisDiv

Synopsis

Content Model

refsynopsisdivinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other "info" elements, `RefSynopsisDivInfo` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `RefSynopsisDivInfo` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain refsynopsisdivinfo: `refsynopsisdiv`.

Children

The following elements occur in refsynopsisdivinfo: `abbrev`, `abstract`, `address`, `affiliation`, `artpagenums`, `author`, `authorblurb`, `authorgroup`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `biblioset`, `bibliosource`, `citebiblioid`, `citetitle`, `collab`, `confgroup`, `contractnum`, `contractsponsor`, `contrib`, `copyright`, `corpauthor`, `corpname`, `date`, `edition`, `editor`, `firstname`, `graphic`, `honorific`, `indexterm`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `itermset`, `keywordset`, `legalnotice`, `lineage`, `mediaobject`, `modespec`, `orgname`, `othercredit`, `othername`, `pagenums`, `personname`, `printhistory`, `productname`, `productnumber`, `pubdate`, `publisher`, `publishername`, `pubsnumber`, `releaseinfo`, `revhistory`, `seriesvolnums`, `subjectset`, `subtitle`, `surname`, `title`, `titleabbrev`, `volumenum`.

See Also

`appendixinfo`, `articleinfo`, `bibliographyinfo`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `prefaceinfo`, `screeninfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setinfo`.

■ releaseinfo

Name

releaseinfo – Information about a particular release of a document

Synopsis

Mixed Content Model

releaseinfo ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

`ReleaseInfo` contains a brief description of the release or published version of a document or part of a document.

For example, the release information may state that the document is in beta, or that the software it describes is a beta version. It may also contain more specific information, such as the version number from a revision control system.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain releaseinfo: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in releaseinfo: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

date, edition, printhistory, pubdate, revhistory.

■ remark

Name

remark – A remark (or comment) intended for presentation in a draft manuscript

Synopsis

Mixed Content Model

remark ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%cptr.char.mix; %divcomponent.mix; %docinfo.char.mix;

%genobj.class; %glossdef.mix; %indexdivcomponent.mix;

%listpreamble.mix; %ndxterm.char.mix; %other.char.class;

%para.char.mix; %qandaset.mix; %refcomponent.mix;

%refinline.char.mix; %revdescription.mix; %sidebar.mix;

%tbl.entry.mdl; %title.char.mix; %word.char.mix;

Description

The Remark element is designed to hold remarks, for example, editorial comments, that are useful while the document is in the draft stage, but are not intended for final publication.

Remarks are available almost anywhere and have a particularly broad content model. Your processing system may or may not support either the use of comments everywhere they are allowed or the full generality of the Remark content model.

Prior to version 4.0 of DocBook, this element was named Comment.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Remarks are often printed only in draft versions of a document and suppressed otherwise. This may be controlled by the Status attribute of an ancestor element (for example, Chapter), or by external processes, such as selecting an alternate stylesheet when publishing.

Remarks must not be nested within other Remarks. Because DocBook is harmonizing towards XML, this restriction cannot be enforced by the DTD. The processing of nested comments is undefined.

Parents

These elements contain remark: abbrev, ackno, acronym, action, answer, appendix, application, article, artpagenums, attribution, authorinitials, bibliocoverage, bibliodiv, bibliography, biblioid, bibliomisc, bibliorelation, bibliosource, blockquote, bridgehead, callout, caution, chapter, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, constraintdef, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, fax, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, hardware, holder, honorific, important, index, indexdiv, interfacename, invpartnumber, isbn, issn, issuenum, itemizedlist, jobtitle, keycap, label, lineage, lineannotation, link, listitem, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, msgexplan, msgtext, note, olink, option, optional, orderedlist, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, partintro, phone, phrase, pob, postcode, preface, primary, primaryie, procedure, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, qandadiv, qandaset, question, quote, refentry, refentrytitle, refmiscinfo, refnamediv, refpurpose, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, releaseinfo, remark, replaceable, revdescription, revnumber, revremark, screen, screeninfo, secondary, secondaryie, sect1, sect2, sect3, sect4, sect5, section, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, setindex, shortaffil, sidebar, simpara, simplesect, state, step, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, tip, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, variablelist, volumenum, warning, wordasword, year.

Children

The following elements occur in remark: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Examples

```
<!DOCTYPE example PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<example><title>The Grand Unified Theory</title>
<para>
<remark>Some details are still a bit shaky</remark>
&hellip;
Q.E.D.
</para>
</example>
```

Example 6.288.1: The Grand Unified Theory

Some details are still a bit shaky … Q.E.D.

■ replaceable

Name

replaceable – Content that may or must be replaced by the user

Synopsis

Mixed Content Model

replaceable ::= (#PCDATA|link|olink|ulink|optional|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|co)*

Attributes Common attributes

Name Type Default

class — command | function | option | parameter — *None*

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %ndxterm.char.mix;

%para.char.mix; %refinline.char.mix; %refname.char.mix;

%smallcptr.char.mix; %tbl.entry.mdl; %tech.char.class;

%title.char.mix;

Description

Replaceable is used to mark text that describes *what* a user is supposed to enter, but not the *actual text* that they are supposed to enter.

It is used to identify a class of object in the document, in which the user is expected to replace the text that identifies the class with some specific instance of that class. A canonical example is

```
<replaceable>filename</replaceable>
```

in which the user is expected to provide the name of some specific file to replace the text “filename.”

Processing expectations

Formatted inline. Usually, the text is given special typographic treatment, such as italics, as a clue to the user that this is replaceable text. Often the font used is described in a "conventions" section at the beginning of the document.

Parents

These elements contain replaceable: accel, ackno, action, application, arg, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classname, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, constant, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, envar, errorcode, errorname, errortext, errortype, exceptionname, fax, filename, firstname, foreignphrase, funcdef, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, group, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, holder, honorific, initializer, interface, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, keycode, keysym, lineage, lineannotation, link, literal, literallayout, lotentry, markup, medialabel, member, methodname, modespec, modifier, mousebutton, msgaud, msglevel, msgorig, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, paramdef, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, prompt, property, pubdate, publishername, pubsnumber, quote, refdescriptor, refentrytitle, refmiscinfo, refname, refpurpose, releaseinfo, remark, returnvalue, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, sgmltag, shortaffil, simpara, state, street, structfield, structname, subscript, subtitle, superscript, surname, symbol, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, token, trademark, type, ulink, userinput, varname, volumenum, year.

Children

The following elements occur in replaceable: anchor, co, inlinegraphic, inlinemediaobject, link, olink, optional, remark, subscript, superscript, ulink.

Attributes

class Class identifies the type of the replaceable information.

See Also

command, computeroutput, constant, literal, markup, option, optional, parameter, prompt, sgmltag, userinput, varname.

Examples

For examples, see cmdsynopsis, medialabel, msgset, synopfragment.

■ returnvalue

Name

returnvalue – The value returned by a function

Synopsis

Mixed Content Model

returnvalue ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

ReturnValue identifies the value returned by a function or command.

Processing expectations

Formatted inline.

Parents

These elements contain returnvalue: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in returnvalue: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

funcdef, funcparams, funcprototype, funcsynopsisinfo, function, paramdef, parameter, varargs, void.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <function>open</function> function returns <returnvalue>2</returnvalue>
(<errorname>ENOFILE</errorname>) if the file does not exist.
</para>
```

The open function returns 2 (ENOFILE) if the file does not exist.

■ revdescription

Name

revdescription – A extended description of a revision to a document

Synopsis

Content Model

revdescription ::= ((calloutlist|glosslist|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)+)

Attributes Common attributes

Description

The RevDescription associated with a revision is a summary of the changes made in that revision. RevDescription is intended for long, complete summaries. For a simple text-only summary, see RevRemark.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain revdescription: revision.

Children

The following elements occur in revdescription: address, anchor, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, note, orderedlist, para, procedure, programlisting, programlistingco, remark, screen, screenco, screenshot, segmentedlist, simpara, simplelist, synopsis, table, tip, variablelist, warning.

■ revhistory

Name

revhistory – A history of the revisions to a document

Synopsis

Content Model

revhistory ::= (revision+)

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %docinfo.char.class; %info.class;

%ndxterm.char.mix; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix;

Description

RevHistory is a structure for documenting a history of changes, specifically, a history of changes to the document or section in which it occurs.

DocBook does not mandate an order for revisions: ascending order by date, descending order by date, and orders based on some other criteria are all equally acceptable.

Processing expectations

Formatted as a displayed block. A tabular or list presentation is most common.

The order of revisions within a revhistory (ascending or descending date order, for example) is not mandated by DocBook

Future Changes

Due to a parameterization oversight in the DTD, RevHistory is allowed in some outlandish places. Still, it is not an inline, so it should not be used inside LineAnnotations, Links, or Quotes. (Not to mention the truly outlandish places like Title and SeeAlso!)

In a future version of DocBook, RevHistory will be removed from these inline contexts.

The original intent for RevHistory was simply to document the history of changes to the document that contains it. In keeping with this meaning, you are advised to limit its use to places where bibliographic meta-information is allowed (the "info" elements).

One can argue that RevHistory has broader applicability for documenting changes to other systems as well, and in light of this, it may become available in more contexts, but that has not yet been decided.

Parents

These elements contain revhistory: appendixinfo, application, articleinfo, attribution, biblioentry, bibliographyinfo, bibliomisc, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, bridgehead, chapterinfo, citation, citetitle, emphasis, entry, foreignphrase, glossaryinfo, glossentry, glosssee, glossseealso, glossterm, indexinfo, lineannotation, link, literallayout, lotentry, member, msgaud, objectinfo, olink, para, partinfo, phrase, prefaceinfo, primary, primaryie, productname, programlisting, qandaentry, quote, refentryinfo, refentrytitle, referenceinfo, refpurpose, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, remark, screen, screeninfo, secondary, secondaryie, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, see, seealso, seealsoie, seeie, seg, segtitle, setindexinfo, setinfo, sidebarinfo, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Children

The following elements occur in revhistory: revision.

See Also

date, edition, printhistory, pubdate, releaseinfo.

Examples

```
<!DOCTYPE revhistory PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<revhistory>

<revision>
  <revnumber>0.91</revnumber>
  <date>11 Dec 1996</date>
  <authorinitials>ndw</authorinitials>
  <revremark>Bug fixes</revremark>
</revision>
```

```
<revision>
  <revnumber>0.90</revnumber>
  <date>30 Nov 1996</date>
  <authorinitials>ndw</authorinitials>
  <revremark>First beta release</revremark>
</revision>

</revhistory>
```

For additional examples, see also `article`.

■ revision

Name

revision – An entry describing a single revision in the history of the revisions to a document

Synopsis

Content Model

revision ::= (`revnumber`,`date`,`authorinitials`*, (`revremark`|`revdescription`)?)

Attributes Common attributes

Description

`Revision` contains information about a single revision to a document. Revisions are identified by a number and a date. They may also include the initials of the author, and additional remarks.

Processing expectations

`Revision`s are often presented in a list or table. In a tabular presentation, each revision most likely forms a row in the table.

Parents

These elements contain revision: `revhistory`.

Children

The following elements occur in revision: `authorinitials`, `date`, `revdescription`, `revnumber`, `revremark`.

Examples

For examples, see `article`, `revhistory`.

■ revnumber

Name

revnumber – A document revision number

Synopsis

Mixed Content Model

revnumber ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

A `RevNumber` identifies the revision number of a document. The revision number should uniquely identify a particular revision of a document.

Processing expectations

Formatted inline. DocBook does not require that `RevNumber`s be sequential or make any demands on their format. They can be numeric, alphanumeric, or whatever suits your needs.

Parents

These elements contain revnumber: `revision`.

Children

The following elements occur in revnumber: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

For examples, see `article`, `revhistory`.

■ revremark

Name

revremark – A description of a revision to a document

Synopsis

Mixed Content Model

revremark ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

The `RevRemark` associated with a revision is a short summary of the changes made in that revision. If a longer, more complete summary is desired, see `RevDescription`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain revremark: `revision`.

Children

The following elements occur in revremark: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

For examples, see `article`, `revhistory`.

■ rhs

Name

rhs – The right-hand side of an EBNF production

Synopsis

Mixed Content Model

rhs ::= (#PCDATA|`nonterminal`|`lineannotation`|`sbr`)*

Attributes Common attributes

Description

This element is only available if you are using the EBNF Module.

The right-hand side (`RHS`) of a production provides a definition for the `NonTerminal` on the left-hand side (`LHS`) of the `Production`.

Parents

These elements contain rhs: `production`.

Children

The following elements occur in rhs: `lineannotation`, `nonterminal`, `sbr`.

Examples

For examples, see `productionset`.

■ row

Name

row – A row in a table

Synopsis

Content Model

row ::= ((`entry`|`entrytbl`)+)

Attributes Common attributes

Name Type Default

valign bottom | middle | top *None*

rowsep CDATA *None*

Parameter Entities

```
%tbl.hdft.mdl;
```

Description

A `Row` is a row in a table. It contains all of the cells (`Entry`s or `EntryTbl`s) that appear in that row.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`.

Within a `Row`, cells are arranged horizontally from the start of the row to the end. Cells can, but are not required to, specify the column in which they occur, so it is possible for a row to contain fewer cells than there are columns in the table. This introduces missing cells, which are assumed to be empty. These missing cells can occur anywhere in the row.

Once a cell has been allocated to a column, subsequent cells may not fill preceding columns. In other words, while three cells can specify that they occur in columns 1, 3, and 5, they cannot specify that they occur in columns 1, 5, and 3. Once a column is passed, you can never go back.

If cells do not specify the column in which they occur, they are placed in the next available column. Calculation of the next available column is complicated by horizontal and vertical spanning. Cells from preceding rows can have a vertical span that causes them to extend into the current row, thus occupying space in the current row. These logically occupied cells are skipped when looking for the next available column. Similarly, if a cell has a horizontal span, it logically occupies the columns that follow it. Cells can simultaneously span rows and columns.

Each of the following conditions is an error:

- A cell spans beyond the boundries of the table.
- A row contains more cells than there are columns in the table.
- The arrangement of cells in a row forces one or more cells past the last column of the table.

Parents

These elements contain row: `tbody`, `tfoot`, `thead`.

Children

The following elements occur in row: `entry`, `entrytbl`.

Attributes

rowsep If RowSep has the value 1 (true), then a rule will be drawn below all the cells in this `Row` (unless other, interior elements, suppress some or all of the rules). A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements.

valign VAlign specifies the vertical alignment of text (and other elements) within the cells of this `Row`. If no alignment is specified, it is inherited from enclosing elements.

See Also

`colspec`, `entry`, `entrytbl`, `informaltable`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

For examples, see `entrytbl`, `footnoteref`, `informaltable`, `table`.

■ sbr

Name

sbr – An explicit line break in a command synopsis

Synopsis

Content Model

sbr ::= EMPTY

Attributes Common attributes

Description

For the most part, DocBook attempts to describe document structure rather than presentation. However, in some complex environments, it is possible to demonstrate that there is no reasonable set of processing expectations that can guarantee correct formatting.

`CmdSynopsis` is one of those environments. Within a long synopsis, it may be necessary to specify the location of a line break explicitly.

The `SBR` element indicates the position of such a line break in a `CmdSynopsis`. It is purely presentational.

Processing expectations

`SBR` causes a line break.

Parents

These elements contain sbr: `arg`, `cmdsynopsis`, `group`, `rhs`.

See Also

`arg`, `cmdsynopsis`, `group`, `refsynopsisdiv`, `synopfragment`, `synopfragmentref`.

Examples

For examples, see `cmdsynopsis`, `synopfragment`.

■ screen

Name

screen – Text that a user sees or might see on a computer screen

Synopsis

Mixed Content Model

screen ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`|`co`|`coref`|`textobject`|`lineannotation`)*

Attributes Common attributes

Name Type Default

width CDATA *None*

format linespecific "linespecific"

linenumbering numbered
unnumbered

None

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%legalnotice.mix;` `%linespecific.class;` `%listpreamble.mix;`
`%para.mix;` `%qandaset.mix;` `%refcomponent.mix;`
`%revdescription.mix;` `%sidebar.mix;` `%tabentry.mix;`
`%tbl.entry.mdl;` `%textobject.mix;`

Description

A `Screen` is a verbatim environment for displaying text that the user might see on a computer terminal. It is often used to display the results of a command.

Having less specific semantic overtones, `Screen` is often used wherever a verbatim presentation is desired, but the semantic of `ProgramListing` is inappropriate.

Processing expectations

This element is displayed “verbatim”; whitespace and linebreaks within this element are significant. `Screen`s are usually displayed in a fixed width font.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Future Changes

The `xml:space` attribute is automatically provided in the XML DTD.

Parents

These elements contain screen: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `example`, `figure`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `indexdiv`, `informalexample`, `informalfigure`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `orderedlist`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `screenco`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `variablelist`, `warning`.

Children

The following elements occur in screen: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `co`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `coref`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobjec`
`interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `lineannotation`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`,

parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, textobject, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

format The Format attribute applies the linespecific notation to all Screens. All white space and line breaks must be preserved.

linenumbering Line numbering indicates whether or not the lines of a Screen are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

width Width specifies the width (in characters) of the longest line in this Screen (formatters may use this value to determine scaling or rotation).

See Also

computeroutput, lineannotation, literallayout, programlisting, screenshot, synopsis, userinput.

Examples

```
<!DOCTYPE screen PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<screen>
 Volume in drive C is SYSTEM         Serial number is 2350:717C
 Directory of  C:\

10/17/97   9:04         &lt;DIR&gt;    bin
10/16/97  14:11         &lt;DIR&gt;    DOS
10/16/97  14:40         &lt;DIR&gt;    Program Files
10/16/97  14:46         &lt;DIR&gt;    TEMP
10/17/97   9:04         &lt;DIR&gt;    tmp
10/16/97  14:37         &lt;DIR&gt;    WINNT
10/16/97  14:25             119  AUTOEXEC.BAT
 2/13/94   6:21          54,619  COMMAND.COM
10/16/97  14:25             115  CONFIG.SYS
11/16/97  17:17      61,865,984  pagefile.sys
 2/13/94   6:21           9,349  WINA20.386
</screen>
```

```
 Volume in drive C is SYSTEM         Serial number is 2350:717C
 Directory of  C:\

10/17/97   9:04         <DIR>    bin
10/16/97  14:11         <DIR>    DOS
10/16/97  14:40         <DIR>    Program Files
10/16/97  14:46         <DIR>    TEMP
10/17/97   9:04         <DIR>    tmp
10/16/97  14:37         <DIR>    WINNT
10/16/97  14:25             119  AUTOEXEC.BAT
 2/13/94   6:21          54,619  COMMAND.COM
10/16/97  14:25             115  CONFIG.SYS
11/16/97  17:17      61,865,984  pagefile.sys
 2/13/94   6:21           9,349  WINA20.386
```

For additional examples, see also `lineannotation`, `screenco`.

■ screenco

Name

screenco – A screen with associated areas used in callouts

Synopsis

Content Model

screenco ::= (`areaspec`,`screen`,`calloutlist`*)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%legalnotice.mix;` `%linespecific.class;` `%listpreamble.mix;`
`%para.mix;` `%qandaset.mix;` `%refcomponent.mix;`
`%revdescription.mix;` `%sidebar.mix;` `%tabentry.mix;`
`%tbl.entry.mdl;` `%textobject.mix;`

Description

`Callout`s, such as numbered bullets, are an annotation mechanism. In an online system, these bullets are frequently "hot," and clicking on them navigates to the corresponding annotation.

A `ScreenCO` is a wrapper around an `AreaSpec` and a `Screen`. An `AreaSpec` identifies the locations (coordinates) in the `Screen` where the callouts occur. The `ScreenCO` may also contain the list of annotations in a `CalloutList`, although the `CalloutList` may also occur outside of the wrapper, elsewhere in the document.

It is also possible to embed `CO` elements directly in the verbatim text, in order to avoid the overhead of calculating the correct coordinates. If you decide to follow this route, use a `Screen` and a `CalloutList` without the `ScreenCO` wrapper. A `ScreenCO` must specify at least one coordinate.

For a complete description of callouts, see `Callout`.

Processing expectations

Formatted as a displayed block. This element is displayed "verbatim"; whitespace and linebreaks within this element are significant.

The mandatory processing expectations of a `ScreenCO` are minimal: a system is expected to render the program listing and the callout list, if present.

If explicit CO elements are embedded in a Screen, they must generate appropriate callout marks.

In online environments, the processing system may be able to instantiate the linking relationships between the callout marks in the program listing and the annotations. Some systems may even be able to go a step further and generate the callout marks automatically from the coordinate information, but this level of sophistication is not mandatory.

Parents

These elements contain screenco: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in screenco: areaspec, calloutlist, screen.

See Also

areaspec, calloutlist, co, coref, graphicco, imageobjectco, mediaobjectco, programlistingco.

Examples

```
<!DOCTYPE informalexample PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<informalexample>
<screen>
 Volume in drive C is SYSTEM         Serial number is 2350:717C
 Directory of  C:\

10/17/97   9:04         &lt;DIR>    bin
10/16/97  14:11         &lt;DIR>    DOS            <co id="dos"/>
10/16/97  14:40         &lt;DIR>    Program Files
10/16/97  14:46         &lt;DIR>    TEMP
10/17/97   9:04         &lt;DIR>    tmp
10/16/97  14:37         &lt;DIR>    WINNT
10/16/97  14:25             119  AUTOEXEC.BAT   <co id="autoexec.bat"/>
 2/13/94   6:21          54,619  COMMAND.COM    <co id="command.com"/>
10/16/97  14:25             115  CONFIG.SYS     <co id="config.sys"/>
11/16/97  17:17      61,865,984  pagefile.sys
 2/13/94   6:21           9,349  WINA20.386     <co id="wina20.386"/>
</screen>
<calloutlist>
<callout arearefs="dos">
<para>
This directory holds <trademark>MS-DOS</trademark>, the
operating system that was installed before <trademark>Windows
NT</trademark>.
</para>
</callout>

<callout arearefs="autoexec.bat command.com config.sys">
<para>
System startup code for DOS.
</para>
</callout>
```

```
<callout arearefs="wina20.386">
<para>
Some sort of <trademark>Windows 3.1</trademark> hack for some 386 processors,
as I recall.
</para>
</callout>
</calloutlist>
</informalexample>
```

```
 Volume in drive C is SYSTEM         Serial number is 2350:717C
 Directory of  C:\

10/17/97   9:04         <DIR>    bin
10/16/97  14:11         <DIR>    DOS
10/16/97  14:40         <DIR>    Program Files
10/16/97  14:46         <DIR>    TEMP
10/17/97   9:04         <DIR>    tmp
10/16/97  14:37         <DIR>    WINNT
10/16/97  14:25             119  AUTOEXEC.BAT
 2/13/94   6:21          54,619  COMMAND.COM
10/16/97  14:25             115  CONFIG.SYS
11/16/97  17:17      61,865,984  pagefile.sys
 2/13/94   6:21           9,349  WINA20.386
```

dos This directory holds MS-DOS™, the operating system that was installed before Windows NT™. autoexec.bat command.com config.sys System startup code for DOS. wina20.386 Some sort of Windows 3.1™ hack for some 386 processors, as I recall.

■ screeninfo

Name

screeninfo – Information about how a screen shot was produced

Synopsis

Mixed Content Model

screeninfo ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Description

ScreenInfo contains meta-information about how a ScreenShot was produced. Note that the content model of ScreenShot is radically different from the other “info” elements, to which it bears little or no resemblance.

ScreenInfo is a good place to store information about how and at what resolution a screen shot was produced, when it was produced, and by whom.

Processing expectations

Suppressed.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain screeninfo: screenshot.

Children

The following elements occur in screeninfo: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

Examples

For examples, see screenshot.

■ screenshot

Name

screenshot – A representation of what the user sees or might see on a computer screen

Synopsis

Content Model

screenshot ::= (screeninfo?, (graphic|graphicco|mediaobject|mediaobjectco))

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%divcomponent.mix; %example.mix; %figure.mix;

%footnote.mix; %glossdef.mix; %indexdivcomponent.mix;
%legalnotice.mix; %linespecific.class; %listpreamble.mix;
%para.mix; %qandaset.mix; %refcomponent.mix;
%revdescription.mix; %sidebar.mix; %tabentry.mix;
%tbl.entry.mdl; %textobject.mix;

Description

A ScreenShot is a graphical environment for displaying an image of what the user might see on a computer screen. It is often used to display application screen shots, dialog boxes, and other components of a graphical user interface.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain screenshot: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, figure, footnote, glossary, glossdef, glossdiv, important, index, indexdiv, informalexample, informalfigure, itemizedlist, legalnotice, listitem, msgexplan, msgtext, note, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, variablelist, warning.

Children

The following elements occur in screenshot: graphic, graphicco, mediaobject, mediaobjectco, screeninfo.

See Also

computeroutput, lineannotation, literallayout, programlisting, screen, synopsis, userinput.

Examples

```
<!DOCTYPE screenshot PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">

<screenshot>
<screeninfo>640x480x256</screeninfo>
<graphic fileref="copilot.gif"></graphic>
</screenshot>
```

■ secondary

Name

secondary – A secondary word or phrase in an index term

Synopsis

Mixed Content Model

secondary ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

sortas CDATA *None*

Description

Secondary contains a secondary word or phrase in an IndexTerm. The text of a Secondary term is less significant than the Primary term, but more significant than the Tertiary term for sorting and display purposes.

In IndexTerms, you can only have one primary, secondary, and tertiary term. If you want to index multiple secondary terms for the same primary, you must repeat the primary in another IndexTerm. You cannot place several Secondarys in the same primary.

Processing expectations

Suppressed. This element provides data for processing but is not rendered in the primary flow of text.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain secondary: indexterm.

Children

The following elements occur in secondary: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

sortas SortAs specifies the string by which the element's content is to be sorted. If unspecified, the proper content is used.

See Also

indexentry, indexterm, primary, primaryie, secondaryie, see, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see chapter, indexterm.

■ secondaryie

Name

secondaryie – A secondary term in an index entry, rather than in the text

Synopsis

Mixed Content Model

secondaryie ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

linkends IDREFS *None*

Description

SecondaryIE identifies a secondary word or words in an IndexEntry.

In IndexEntrys, you can specify as many secondary terms that are necessary. Secondary and tertiary terms can be mixed, following the primary.

Processing expectations

Formatted as a displayed block. SecondaryIEs occur below the PrimaryIE, usually aligned with each other and indented from the primary.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain secondaryie: indexentry.

Children

The following elements occur in secondaryie: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec,

mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkends Linkends, if used, points to the IndexTerms indexed by this entry.

See Also

indexentry, indexterm, primary, primaryie, secondary, see, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see index.

■ sect1

Name

sect1 – A top-level section of document

Synopsis

Content Model

sect1 ::= (sect1info?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| sect2*|simplesect*))| (refentry)+| sect2+|simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

renderas sect2 | sect3 | sect4 | sect5 *None*

status CDATA *None*

label CDATA *None*

Parameter Entities

%bookcomponent.content;

Description

Sect1 is one of the top-level sectioning elements in a component. There are three types of sectioning elements in DocBook:

- Explicitly numbered sections, Sect1...Sect5, which must be properly nested and can only be five levels deep.
- Recursive Sections, which are alternative to the numbered sections and have unbounded depth.
- SimpleSects, which are terminal. SimpleSects can occur as the "leaf" sections in either recursive sections or any of the numbered sections, or directly in components.

None of the sectioning elements is allowed to "float" in a component. You can place paragraphs and other block elements before a section, but you cannot place anything after it.

This means that you cannot have content in the Sect1 after the end of a Sect2. This is consistent with the DocBook book model, because in a printed book it is usually impossible for a reader to detect the end of the enclosed second level section and, therefore, all content after a second level section appears in that section.

Processing Expectations

Formatted as a displayed block. Sometimes sections are numbered.

Future Changes

In DocBook V4.0, the ToC element in the content model will be replaced by TocChap.

Parents

These elements contain sect1: appendix, article, chapter, partintro, preface.

Children

The following elements occur in sect1: abstract, address, anchor, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, sect1info, sect2, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

renderas The RenderAs attribute identifies how the section should be rendered. In this way, a section at one level of the structural hierarchy can be made to appear to be at another level.

status Status identifies the editorial or publication status of the Sect1.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

bridgehead, sect2, sect3, sect4, sect5, section, simplesect.

Examples

For examples, see chapter, indexterm, link, olink, sect1info, sidebar, simplesect, title.

■ sect1info

Name

sect1info – Meta-information for a Sect1

Synopsis

Content Model

sect1info ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

Like the other “info” elements, Sect1Info contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of Sect1Info in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sect1info: sect1.

Children

The following elements occur in sect1info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect2info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

Examples

```
<!DOCTYPE sect1 PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<sect1>
  <sect1info>
    <legalnotice><para>In the public domain.</para></legalnotice>
    <title>Something Pithy</title>
    <subtitle>How I Made Up a Silly Example</subtitle>
  </sect1info>
<title>Something Pithy</title>
<para>
Content.
</para>
</sect1>
```

■ sect2

Name

sect2 – A subsection within a Sect1

Synopsis

Content Model

sect2 ::= (sect2info?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| sect3*|simplesect*))| (refentry)+| sect3+|simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

renderas (sect1 | sect3 | sect4 | sect5) *None*

status CDATA *None*

label CDATA *None*

Description

A Sect2 is a second-level section in a document.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Future Changes

In DocBook V4.0, the `ToC` element in the content model will be replaced by `TocChap`.

Parents

These elements contain sect2: `sect1`.

Children

The following elements occur in sect2: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `bibliography`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glossary`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `index`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `lot`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refentry`, `remark`, `screen`, `screenco`, `screenshot`, `sect2info`, `sect3`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `simplesect`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `toc`, `variablelist`, `warning`.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

renderas The RenderAs attribute identifies how the section should be rendered. In this way, a section at one level of the structural hierarchy can be made to appear to be at another level.

status Status identifies the editorial or publication status of the `Sect2`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`bridgehead`, `sect1`, `sect3`, `sect4`, `sect5`, `section`, `simplesect`.

Examples

For examples, see `link`.

■ sect2info

Name

sect2info – Meta-information for a `Sect2`

Synopsis

Content Model

sect2info ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

Like the other "info" elements, Sect2Info contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of Sect2Info in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sect2info: sect2.

Children

The following elements occur in sect2info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect3info, sect4info, sect5info, sectioninfo, setinfo.

Examples

See Sect1Info for an analogous example.

■ sect3

Name

sect3 – A subsection within a Sect2

Synopsis

Content Model

sect3 ::= (sect3info?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| sect4*|simplesect*))| (refentry)+| sect4+|simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

renderas (sect1 | sect2 | sect4 | sect5) *None*

status CDATA *None*

label CDATA *None*

Description

A Sect3 is a third-level section in a document.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Future Changes

In DocBook V4.0, the ToC element in the content model will be replaced by TocChap.

Parents

These elements contain sect3: sect2.

Children

The following elements occur in sect3: abstract, address, anchor, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, sect3info, sect4, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

renderas The RenderAs attribute identifies how the section should be rendered. In this way, a section at one level of the structural hierarchy can be made to appear to be at another level.

status Status identifies the editorial or publication status of the `Sect3`.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

`bridgehead`, `sect1`, `sect2`, `sect4`, `sect5`, `section`, `simplesect`.

■ sect3info

Name

sect3info – Meta-information for a `Sect3`

Synopsis

Content Model

sect3info ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other "info" elements, `Sect3Info` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `Sect3Info` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sect3info: `sect3`.

Children

The following elements occur in sect3info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect4info, sect5info, sectioninfo, setinfo.

Examples

See Sect1Info for an analogous example.

■ sect4

Name

sect4 – A subsection within a Sect3

Synopsis

Content Model

sect4 ::= (sect4info?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| sect5*|simplesect*))| (refentry)+| sect5+|simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

renderas (sect1 | sect2 | sect3 | sect5) *None*

status CDATA *None*

label CDATA *None*

Description

A Sect4 is a fourth-level section in a document.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Future Changes

In DocBook V4.0, the `ToC` element in the content model will be replaced by `TocChap`.

Parents

These elements contain sect4: `sect3`.

Children

The following elements occur in sect4: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `bibliography`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glossary`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `index`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable` `itemizedlist`, `literallayout`, `lot`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `refentry`, `remark`, `screen`, `screenco`, `screenshot`, `sect4info`, `sect5`, `segmentedlist`, `sidebar`, `simpara`, `simplelist`, `simplesect`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `toc`, `variablelist`, `warning`.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

renderas The RenderAs attribute identifies how the section should be rendered. In this way, a section at one level of the structural hierarchy can be made to appear to be at another level.

status Status identifies the editorial or publication status of the `Sect4`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`bridgehead`, `sect1`, `sect2`, `sect3`, `sect5`, `section`, `simplesect`.

■ sect4info

Name

sect4info – Meta-information for a `Sect4`

Synopsis

Content Model

sect4info ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

Like the other "info" elements, Sect4Info contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of Sect4Info in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sect4info: sect4.

Children

The following elements occur in sect4info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect5info, sectioninfo, setinfo.

Examples

See Sect1Info for an analogous example.

■ sect5

Name

sect5 – A subsection within a Sect4

Synopsis

Content Model

sect5 ::= (sect5info?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| simplesect*))| (refentry)+| simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

renderas (sect1 | sect2 | sect3 | sect4) *None*

status CDATA *None*

label CDATA *None*

Description

A Sect5 is a fifth-level section in a document. This is the lowest-level numbered sectioning element. There is no Sect6.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Future Changes

In DocBook V4.0, the ToC element in the content model will be replaced by TocChap.

Parents

These elements contain sect5: sect4.

Children

The following elements occur in sect5: abstract, address, anchor, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, sect5info, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

renderas The RenderAs attribute identifies how the section should be rendered. In this way, a section at one level of the structural hierarchy can be made to appear to be at another level.

status Status identifies the editorial or publication status of the `Sect5`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`bridgehead`, `sect1`, `sect2`, `sect3`, `sect4`, `section`, `simplesect`.

■ sect5info

Name

sect5info – Meta-information for a `Sect5`

Synopsis

Content Model

sect5info ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`biblierelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other “info” elements, `Sect5Info` contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

`AuthorBlurb` and `Affiliation` will be removed from the inline content of `Sect5Info` in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sect5info: sect5.

Children

The following elements occur in sect5info: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sectioninfo, setinfo.

Examples

See Sect1Info for an analogous example.

■ section

Name

section – A recursive section

Synopsis

Content Model

section ::= (sectioninfo?, (title,subtitle?,titleabbrev?), (toc|lot|index|glossary| bibliography)*, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset| productionset| constraintdef|anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm|beginpage)+, ((refentry)*| (section)*| simplesect*))| (refentry)+| (section)+| simplesect+), (toc|lot|index|glossary|bibliography)*)

Attributes Common attributes

Name Type Default

status CDATA *None*

label CDATA *None*

Parameter Entities

%bookcomponent.content; %section.class;

Description

Section is one of the top-level sectioning elements in a component. There are three types of sectioning elements in DocBook:

- Explicitly numbered sections, Sect1...Sect5, which must be properly nested and can only be five levels deep.
- Recursive Sections, which are an alternative to the numbered sections and have unbounded depth.
- SimpleSects, which are terminal. SimpleSects can occur as the "leaf" sections in either recursive sections or any of the numbered sections, or directly in components.

Sections may be more convenient than numbered sections in some authoring environments because they can be moved around in the document hierarchy without renaming.

None of the sectioning elements is allowed to "float" in a component. You can place paragraphs and other block elements before a section, but you cannot place anything after it.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Use of deeply nested Sections may cause problems in some processing systems.

Parents

These elements contain section: appendix, article, chapter, partintro, preface, section.

Children

The following elements occur in section: abstract, address, anchor, authorblurb, beginpage, bibliography, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glossary, glosslist, graphic, graphicco, highlights, important, index, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, lot, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, refentry, remark, screen, screenco, screenshot, section, sectioninfo, segmentedlist, sidebar, simpara, simplelist, simplesect, subtitle, synopsis, table, tip, title, titleabbrev, toc, variablelist, warning.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

status Status identifies the editorial or publication status of the Section.

Publication status might be used to control formatting (for example, printing a "draft" watermark on drafts) or processing (perhaps a document with a status of "final" should not include any components that are not final).

See Also

bridgehead, sect1, sect2, sect3, sect4, sect5, simplesect.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Test Chapter</title>
<para>This chapter uses recursive sections.</para>
<section>
<sectioninfo>
<abstract><para>A trivial example of recursive sections.</para>
</abstract>
</sectioninfo>
<title>Like a Sect1</title>
<subtitle>Or How I Learned to Let Go of Enumeration
and Love to Recurse</subtitle>
<para>This section is like a Sect1.</para>
<section><title>Like a Sect2</title>
<para>This section is like a Sect2.</para>
<section><title>Like a Sect3</title>
<para>This section is like a Sect3.</para>
<section><title>Like a Sect4</title>
<para>This section is like a Sect4.</para>
<section><title>Like a Sect5</title>
<para>This section is like a Sect5.</para>
<section><title>Would be like a Sect6</title>
<para>This section would be like a Sect6, if there was one.</para>
<section><title>Would be like a Sect7</title>
<para>This section would be like a Sect7, if there was one.</para>
</section>
</section>
</section>
</section>
</section>
</section>
</section>
</chapter>
```

For additional examples, see also `html-form`.

■ sectioninfo

Name

sectioninfo – Meta-information for a recursive section

Synopsis

Content Model

sectioninfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

Like the other "info" elements, SectionInfo contains meta-information about the section of the document in which it occurs.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of SectionInfo in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Parents

These elements contain sectioninfo: section.

Children

The following elements occur in sectioninfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, setinfo.

Examples

For examples, see section.

■ see

Name

see – Part of an index term directing the reader instead to another entry in the index

Synopsis

Mixed Content Model

see ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue|

sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Description

The use of See in an IndexTerm indicates that the reader should be directed elsewhere in the index if they attempt to look up this term.

The content of See identifies another term in the index which the reader should consult *instead* of the current term.

Processing expectations

Suppressed. This element provides data for processing but it is not rendered in the primary flow of text.

It is possible for multiple IndexTerms, taken together, to form an illogical index. For example, given the following IndexTerms:

```
<indexterm><primary>Extensible Markup Language</primary>
  <see>XML</see></indexterm>
<indexterm><primary>Extensible Markup Language</primary>
  <secondary>definition of</secondary>
</indexterm>
```

there's no way to construct a logical index because an entry in the index should never have both a see and other content.

DocBook cannot detect these errors. You will have to rely on an external process to find them.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain see: indexterm.

Children

The following elements occur in see: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

indexentry, indexterm, primary, primaryie, secondary, secondaryie, seealso, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see indexterm.

■ seealso

Name

seealso – Part of an index term directing the reader also to another entry in the index

Synopsis

Mixed Content Model

seealso ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject)*

Attributes Common attributes

Description

The use of SeeAlso in an IndexTerm indicates that the reader should be directed elsewhere in the index for additional information.

The content of SeeAlso identifies another term in the index that the reader should consult *in addition to* the current term.

Processing expectations

Suppressed. This element provides data for processing but is not rendered in the primary flow of text.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain seealso: indexterm.

Children

The following elements occur in seealso: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname,

subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

indexentry, indexterm, primary, primaryie, secondary, secondaryie, see, seealsoie, seeie, tertiary, tertiaryie.

Examples

For examples, see indexterm.

■ seealsoie

Name

seealsoie – A “See also” entry in an index, rather than in the text

Synopsis

Mixed Content Model

seealsoie ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

linkends IDREFS *None*

Description

SeeAlsoIE identifies a “See also” cross-reference in an IndexEntry. IndexEntrys occur in an Index, not in the flow of the text. They are part of a formatted index, not markers for indexing.

Processing expectations

Formatted as a displayed block. IndexEntrys that include a SeeAlsoIE should be formatted normally, with the “See also” indented below the term.

SeeAlsoIE is usually expected to generate the text “See Also”.

The Linkends attribute should point other IndexEntrys in the same Index. Online systems may use them to form hypertext links.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain seealsoie: indexentry.

Children

The following elements occur in seealsoie: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkends Linkends points to a the related IndexEntrys.

See Also

indexentry, indexterm, primary, primaryie, secondary, secondaryie, see, seealso, seeie, tertiary, tertiaryie.

Examples

For examples, see index.

■ seeie

Name

seeie – A “See” entry in an index, rather than in the text

Synopsis

Mixed Content Model

seeie ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

linkend IDREF *None*

Description

SeeIE identifies a “See” cross reference in an IndexEntry. IndexEntrys occur in an Index, not in the flow of the text. They are part of a formatted index, not markers for indexing.

Processing expectations

Formatted as a displayed block. `IndexEntry`s that include a `SeeIE` should be formatted normally, with the "See also" indented below the term. There should be no other entries for this term.

`SeeIE` is usually expected to generate the text, "See."

The Linkend attribute should point to the referenced `IndexEntry`s, which should be in the same `Index`. Online systems may use the link information to form a hypertext link.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain seeie: `indexentry`.

Children

The following elements occur in seeie: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `citation`, `citerefentry`, `citetitle`, `classname`, `command`, `computeroutput`, `constant`, `corpauthor`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

linkend Linkend points to the associated `IndexEntry`.

See Also

`indexentry`, `indexterm`, `primary`, `primaryie`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `tertiary`, `tertiaryie`.

Examples

For examples, see `index`.

■ seg

Name

seg – An element of a list item in a segmented list

Synopsis

Mixed Content Model

seg ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|`trademark`|`wordasword`|`personname`|`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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Description

A SegmentedList consists of a set of headings (SegTitles) and a list of parallel sets of elements. Every SegListItem contains a set of elements that have a one-to-one correspondence with the headings. Each of these elements is contained in a Seg.

Processing expectations

Segmented lists can be formatted in a number of ways. Two popular formats are tabular and as a list of repeated headings and elements. In a tabular presentation, each Seg is a cell in the body of the table. In the list presentation, each Seg occurs next to the appropriate heading.

DocBook cannot detect errors caused by too many or too few Segs in a SegListItem. You will have to rely on external processes to find those errors.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain seg: seglistitem.

Children

The following elements occur in seg: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Examples

For examples, see segmentedlist.

■ seglistitem

Name

seglistitem – A list item in a segmented list

Synopsis

Content Model

seglistitem ::= (`seg`+)

Attributes Common attributes

Description

A `SegmentedList` consists of a set of headings (`SegTitle`s) and a list of parallel sets of elements. Each set of elements is stored in a `SegListItem`.

Processing expectations

Segmented lists can be formatted in a number of ways. Two popular formats are tabular and as a list of repeated headings and elements. In a tabular presentation, each `SegListItem` is a row in the table. In the list presentation, each `SegListItem` contains a block of heading/element pairs.

DocBook cannot detect errors caused by too many or too few `Seg`s in a `SegListItem`. You will have to rely on external processes to find those errors.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain seglistitem: `segmentedlist`.

Children

The following elements occur in seglistitem: `seg`.

Examples

For examples, see `segmentedlist`.

■ segmentedlist

Name

segmentedlist – A segmented list, a list of sets of elements

Synopsis

Content Model

segmentedlist ::= ((`title`,`titleabbrev`?)?, `segtitle`+,`seglistitem`+)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%footnote.mix;`
`%glossdef.mix;` `%highlights.mix;` `%legalnotice.mix;`

```
%list.class; %para.mix; %qandaset.mix;
%refcomponent.mix; %revdescription.mix; %sidebar.mix;
%tabentry.mix; %tbl.entry.mdl; %textobject.mix;
```

Description

A SegmentedList consists of a set of headings (SegTitles) and a list of parallel sets of elements. Every SegListItem contains a set of elements that have a one-to-one correspondence with the headings. Each of these elements is contained in a Seg.

Processing expectations

Segmented lists can be formatted in a number of ways. Two popular formats are tabular and as a list of repeated headings and elements. In a tabular presentation, the SegmentedList is the table. In the list presentation, the SegmentedList surrounds the entire list of blocks of heading/element pairs.

Future Changes

In DocBook V4.0, at least two SegTitles will be required.

Parents

These elements contain segmentedlist: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, footnote, glossary, glossdef, glossdiv, highlights, important, index, indexdiv, informalexample, legalnotice, listitem, msgexplan, msgtext, note, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, warning.

Children

The following elements occur in segmentedlist: seglistitem, segtitle, title, titleabbrev.

See Also

calloutlist, itemizedlist, listitem, orderedlist, simplelist, variablelist.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The capitals of the states of the United States of America are:

<segmentedlist><title>State Capitals</title>
<?dbhtml list-presentation="list"?>
<segtitle>State</segtitle>
<segtitle>Capital</segtitle>
<seglistitem><seg>Alabama</seg><seg>Montgomery</seg></seglistitem>
<seglistitem><seg>Alaska</seg><seg>Anchorage</seg></seglistitem>
<seglistitem><seg>Arkansas</seg><seg>Little Rock</seg></seglistitem>
</segmentedlist>

&hellip;
</para>
```

The capitals of the states of the United States of America are: STATE CAPITALS
State: Alabama *Capital:* Montgomery
State: Alaska *Capital:* Anchorage
State: Arkansas *Capital:* Little Rock
… Alternatively:

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
<segmentedlist><title>State Capitals</title>
<?dbhtml list-presentation="table"?>
<segtitle>State</segtitle>
<segtitle>Capital</segtitle>
<seglistitem><seg>Alabama</seg><seg>Montgomery</seg></seglistitem>
<seglistitem><seg>Alaska</seg><seg>Anchorage</seg></seglistitem>
<seglistitem><seg>Arkansas</seg><seg>Little Rock</seg></seglistitem>
</segmentedlist>

&hellip;
</para>
```

STATE CAPITALS
State: Alabama *Capital:* Montgomery
State: Alaska *Capital:* Anchorage
State: Arkansas *Capital:* Little Rock
…

■ segtitle

Name

segtitle – The title of an element of a list item in a segmented list

Synopsis

Mixed Content Model

segtitle ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`| `citetitle`| `emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|`trademark`| `wordasword`|`personname`|`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`|`constant`|`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`| `replaceable`|`returnvalue`| `sgmltag`|`structfield`|`structname`|`symbol`| `systemitem`|`token`|`type`|`userinput`|`varname`| `nonterminal`|`anchor`| `author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|`subscript`| `superscript`|`inlinegraphic`| `inlinemediaobject`|`inlineequation`| `indexterm`)*

Attributes Common attributes

Description

Each heading in a `SegmentedList` is contained in its own `SegTitle`.

The relationship between `SegTitle`s and `Seg`s is implicit in the document; the first `SegTitle` goes with the first `Seg` in each `SegListItem`, the second `SegTitle` goes with the second `Seg`, and so on.

Processing expectations

Segmented lists can be formatted in a number of ways. Two popular formats are tabular and as list of repeated headings and elements. In a tabular presentation, each `SegTitle` is a column heading. In the list presentation, each `SegTitle` is repeated before the corresponding `Seg`.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain segtitle: `segmentedlist`.

Children

The following elements occur in segtitle: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `citation`, `citerefentry`, `citetitle`, `classname`, `command`, `computeroutput`, `constant`, `corpauthor`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Examples

For examples, see `segmentedlist`.

■ seriesvolnums

Name

seriesvolnums – Numbers of the volumes in a series of books

Synopsis

Mixed Content Model

seriesvolnums ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

`SeriesVolNums` contains the numbers of the volumes of the books in a series. It is a wrapper for bibliographic information.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain seriesvolnums: appendixinfo, articleinfo, biblioentry, bibliographyinfo, bibliomixed, bibliomset, biblioset, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in seriesvolnums: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

biblioid, invpartnumber, isbn, issn, issuenum, productnumber, pubsnumber, volumenum.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
TBS.
</para>
```

TBS.

■ set

Name

set – A collection of books

Synopsis

Content Model

set ::= ((title,subtitle?,titleabbrev?)?, setinfo?,toc?, (book)+, setindex?)

Attributes Common attributes

Name Type Default

fpi CDATA *None*

status CDATA *None*

Description

A Set is a collection of Books. Placing multiple Books in a Set, as opposed to publishing each of them separately, has the advantage that ID/IDREF links can then be used across all books.

Set is the very top of the DocBook structural hierarchy. There's nothing that contains a Set.

Processing expectations

Formatted as a displayed block. A Set may generate additional front and back matter (tables of contents and SetIndexs, for example) around the Books it contains.

Children

The following elements occur in set: book, setindex, setinfo, subtitle, title, titleabbrev, toc.

Attributes

fpi FPI holds the Formal Public Identifier for the `Set`.

status Status identifies the editorial or publication status of the `Set`.

Publication status might be used to control formatting (for example, printing a “draft” watermark on drafts) or processing (perhaps a document with a status of “final” should not include any components that are not final).

See Also

`appendix`, `article`, `book`, `chapter`, `colophon`, `dedication`, `part`, `partintro`, `preface`.

Examples

```
<!DOCTYPE set PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">

<set><title>The Perl Series</title>
<setinfo>
  <corpauthor>O’Reilly &amp;amp; Associates, Inc.</corpauthor>
</setinfo>

<book><title>Learning Perl</title>
<chapter><title>...</title><para>...</para></chapter>
</book>

<book><title>Programming Perl</title>
<chapter><title>...</title><para>...</para></chapter>
</book>

<book><title>Advanced Perl Programming</title>
<chapter><title>...</title><para>...</para></chapter>
</book>

</set>
```

■ setindex

Name

setindex – An index to a set of books

Synopsis

Content Model

setindex ::= (`setindexinfo`?, (`title`,`subtitle`?,`titleabbrev`?)?, (`calloutlist`|`glosslist`|`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`|`msgset`|`procedure`|`sidebar`|`qandaset`|`productionset`|`constraintdef`|`anchor`|`bridgehead`|`remark`|`highlights`|`abstract`|`authorblurb`|`epigraph`|`indexterm`|`beginpage`)*, (`indexdiv`*|`indexentry`*))

Attributes Common attributes

Parameter Entities

`%index.class;`

Description

A `SetIndex` contains the formatted index of a complete `Set` of `Book`s. An index may begin with introductory material, followed by any number of `IndexEntry`s or `IndexDiv`s.

Processing expectations

Formatted as a displayed block. An `Index` in a `Set` usually causes a forced page break in print media.

In many processing systems, indexes are generated automatically or semiautomatically and never appear instantiated as DocBook markup.

Parents

These elements contain setindex: `book`, `set`.

Children

The following elements occur in setindex: `abstract`, `address`, `anchor`, `authorblurb`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constraintdef`, `constructorsynopsis`, `destructorsynopsis`, `epigraph`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexdiv`, `indexentry`, `indexterm`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `msgset`, `note`, `orderedlist`, `para`, `procedure`, `productionset`, `programlisting`, `programlistingco`, `qandaset`, `remark`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `setindexinfo`, `sidebar`, `simpara`, `simplelist`, `subtitle`, `synopsis`, `table`, `tip`, `title`, `titleabbrev`, `variablelist`, `warning`.

Examples

See `Index`.

■ setindexinfo

Name

setindexinfo – Meta-information for a `SetIndex`

Synopsis

Content Model

setindexinfo ::= ((`graphic`|`mediaobject`|`legalnotice`|`modespec`|`subjectset`|`keywordset`|`itermset`|`abbrev`|`abstract`|`address`|`artpagenums`|`author`|`authorgroup`|`authorinitials`|`bibliomisc`|`biblioset`|`collab`|`confgroup`|`contractnum`|`contractsponsor`|`copyright`|`corpauthor`|`corpname`|`date`|`edition`|`editor`|`invpartnumber`|`isbn`|`issn`|`issuenum`|`orgname`|`biblioid`|`citebiblioid`|`bibliosource`|`bibliorelation`|`bibliocoverage`|`othercredit`|`pagenums`|`printhistory`|`productname`|`productnumber`|`pubdate`|`publisher`|`publishername`|`pubsnumber`|`releaseinfo`|`revhistory`|`seriesvolnums`|`subtitle`|`title`|`titleabbrev`|`volumenum`|`citetitle`|`personname`|`honorific`|`firstname`|`surname`|`lineage`|`othername`|`affiliation`|`authorblurb`|`contrib`|`indexterm`)+)

Attributes Common attributes

Description

The SetIndexInfo element is a wrapper for a large collection of meta-information about a SetIndex. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain setindexinfo: setindex.

Children

The following elements occur in setindexinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

■ setinfo

Name

setinfo – Meta-information for a Set

Synopsis

Content Model

setinfo ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset| itermset|abbrev|abstract|address|artpagenums|author|authorgroup| authorinitials| bibliomisc|biblioset|collab|confgroup| contractnum|contractsponsor|copyright| corpauthor|corpname|date| edition|editor|invpartnumber|isbn|issn|issuenum| orgname| biblioid|citebiblioid|bibliosource|bibliorelation| bibliocoverage| othercredit|pagenums|printhistory|productname| productnumber|pubdate|publisher| publishername|pubsnumber| releaseinfo|revhistory|seriesvolnums|subtitle| title|titleabbrev| volumenum|citetitle|personname|honorific|firstname|surname| lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Name Type Default

contents IDREFS *None*

Description

SetInfo contains meta-information about an entire set of Books.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. The wrapper merely serves to identify where they occur.

Future Changes

AuthorBlurb and Affiliation will be removed from the inline content of SetInfo in DocBook V4.0. A new wrapper element will be created to associate this information with authors, editors, and other contributors.

Future Changes

The contents attribute will be removed.

Parents

These elements contain setinfo: set.

Children

The following elements occur in setinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

Attributes

contents Contents, if specified, should contain a list of all the IDs of the book-level subelements in the Set, presumably in their natural order.

See Also

appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, prefaceinfo, refsynopsisdivinfo, screeninfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo.

Examples

For examples, see set.

■ sgmltag

Name

sgmltag – A component of SGML markup

Synopsis

Mixed Content Model

sgmltag ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

class	attribute attvalue element emptytag endtag genentity numcharref paramentity pi sgmlcomment starttag xmlpi
None	

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

An `SGMLTag` identifies an SGML markup construct. The utility of this element is almost wholly constrained to books about SGML.

`SGMLTag` is sufficient for most XML constructs, which are identical to the corresponding SGML constructs, it but does not have any provisions for handling the special features of XML markup. A future version of DocBook will address this issue, probably by adding new Class values. In the meantime, you may get by by assigning a Role attribute for XML.

Processing expectations

Formatted inline. `SGMLTag` generates all the necessary punctuation before and after the construct it identifies. For example, it generates both the leading ampersand and the trailing semicolon when the Class is `genentity`.

Parents

These elements contain sgmltag: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in sgmltag: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

Attributes

class Class identifies the specific SGML construct represented.

See Also

`computeroutput`, `constant`, `literal`, `markup`, `option`, `optional`, `parameter`, `prompt`, `replaceable`, `userinput`, `varname`.

Examples

For examples, see `collab`, `link`, `simplelist`, `wordasword`, `xref`.

■ shortaffil

Name

shortaffil – A brief description of an affiliation

Synopsis

Mixed Content Model

shortaffil ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`ShortAffil` contains an abbreviated or brief description of an individual's `Affiliation`.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Sometimes suppressed.

Parents

These elements contain shortaffil: `affiliation`.

Children

The following elements occur in shortaffil: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `corpname`, `jobtitle`, `orgdiv`, `orgname`.

Examples

For examples, see `author`, `authorgroup`.

■ shortcut

Name

shortcut – A key combination for an action that is also accessible through a menu

Synopsis

Content Model

shortcut ::= ((`keycap`|`keycombo`|`keysym`|`mousebutton`)+)

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

otheraction CDATA *None*

action
- click
- double-click
- other
- press
- seq
- simul

None

Description

A `Shortcut` contains the key combination that is a shortcut for a `MenuChoice`. Users that are familiar with the shortcuts can access the functionality of the corresponding menu choice, without navigating through the menu structure to find the right menu item.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Parents

These elements contain shortcut: `menuchoice`.

Children

The following elements occur in shortcut: `keycap`, `keycombo`, `keysym`, `mousebutton`.

Attributes

action Action identifies the nature of the action taken. If `Shortcut` contains more than one action element, `Simul` is default value for Action, otherwise there is no default.

If `Other` is specified, OtherAction should be used to identify the nature of the action.

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `Shortcut`.

otheraction OtherAction should be used when Action is set to `Other`. It identifes the nature of the action in some application-specific way.

See Also

`accel`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `keycap`, `keycode`, `keycombo`, `keysym`, `menuchoice`, `mousebutton`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
You can exit from GNU Emacs with
<menuchoice>
  <shortcut>
    <keycombo><keysym>C-x</keysym><keysym>C-c</keysym></keycombo>
  </shortcut>
  <guimenu>Files</guimenu>
  <guimenuitem>Exit Emacs</guimenuitem>
```

```
</menuchoice>.
</para>
```

You can exit from GNU Emacs with **Files** → **Exit Emacs** (**C-x**–**C-c**). For additional examples, see also `menuchoice`.

■ sidebar

Name

sidebar – A portion of a document that is isolated from the main narrative flow

Synopsis

Content Model

sidebar ::= (`sidebarinfo`?, (`title`,`titleabbrev`?)?, (`calloutlist`|`glosslist`|`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`|`beginpage`)+)

Attributes Common attributes

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`

`%compound.class;` `%divcomponent.mix;` `%refcomponent.mix;`

Description

A `Sidebar` is a short piece of text, rarely longer than a single column or page, that is presented outside the narrative flow of the main text.

Sidebars are often used for digressions or interesting observations that are related, but not directly relevant, to the main text.

Processing expectations

Formatted as a displayed block. `Sidebar`s are sometimes boxed.

DocBook does not specify the location of the `Sidebar` within the final displayed flow of text. The wrapper may float or remain where it is located.

Parents

These elements contain sidebar: `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caution`, `chapter`, `constraintdef`, `glossary`, `glossdiv`, `important`, `index`, `listitem`, `msgexplan`, `msgtext`, `note`, `partintro`, `preface`, `procedure`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `simplesect`, `step`, `tip`, `warning`.

Children

The following elements occur in sidebar: `address`, `anchor`, `beginpage`, `blockquote`, `bridgehead`, `calloutlist`, `caution`, `classsynopsis`, `cmdsynopsis`, `constructorsynopsis`, `destructorsynopsis`, `equation`, `example`, `fieldsynopsis`, `figure`, `formalpara`, `funcsynopsis`, `glosslist`, `graphic`, `graphicco`, `highlights`, `important`, `indexterm`, `informalequation`, `informalexample`, `informalfigur` `informaltable`, `itemizedlist`, `literallayout`, `mediaobject`, `mediaobjectco`, `methodsynopsis`, `note`, `orderedlist`, `para`, `procedure`, `programlisting`, `programlistingco`, `remark`, `screen`,

screenco, screenshot, segmentedlist, sidebarinfo, simpara, simplelist, synopsis, table, tip, title, titleabbrev, variablelist, warning.

See Also

abstract, blockquote, epigraph, highlights.

Examples

```
<!DOCTYPE sect1 PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<sect1><title>An Example Section</title>

<para>
Some narrative text.
</para>

<sidebar><title>A Sidebar</title>
<para>
Sidebar content.
</para>
</sidebar>

<para>
The continuing flow of the narrative text, as if the
sidebar was not present.
</para>

</sect1>
```

■ sidebarinfo

Name

sidebarinfo – Meta-information for a Sidebar

Synopsis

Content Model

sidebarinfo ::= ((graphic|mediaobject|legalnotice|modespec|subjectset|keywordset|itermset|abbrev|abstract|address|artpagenums|author|authorgroup|authorinitials|bibliomisc|biblioset|collab|confgroup|contractnum|contractsponsor|copyright|corpauthor|corpname|date|edition|editor|invpartnumber|isbn|issn|issuenum|orgname|biblioid|citebiblioid|bibliosource|biblierelation|bibliocoverage|othercredit|pagenums|printhistory|productname|productnumber|pubdate|publisher|publishername|pubsnumber|releaseinfo|revhistory|seriesvolnums|subtitle|title|titleabbrev|volumenum|citetitle|personname|honorific|firstname|surname|lineage|othername|affiliation|authorblurb|contrib|indexterm)+)

Attributes Common attributes

Description

The SidebarInfo element is a wrapper for a large collection of meta-information about a Sidebar. Much of this data is bibliographic in nature.

Processing expectations

Suppressed. Many of the elements in this wrapper may be used in presentation, but they are not generally printed as part of the formatting of the wrapper. It merely serves to identify where they occur.

Parents

These elements contain sidebarinfo: sidebar.

Children

The following elements occur in sidebarinfo: abbrev, abstract, address, affiliation, artpagenums, author, authorblurb, authorgroup, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, biblioset, bibliosource, citebiblioid, citetitle, collab, confgroup, contractnum, contractsponsor, contrib, copyright, corpauthor, corpname, date, edition, editor, firstname, graphic, honorific, indexterm, invpartnumber, isbn, issn, issuenum, itermset, keywordset, legalnotice, lineage, mediaobject, modespec, orgname, othercredit, othername, pagenums, personname, printhistory, productname, productnumber, pubdate, publisher, publishername, pubsnumber, releaseinfo, revhistory, seriesvolnums, subjectset, subtitle, surname, title, titleabbrev, volumenum.

■ simpara

Name

simpara – A paragraph that contains only text and inline markup, no block elements

Synopsis

Mixed Content Model

simpara ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;
%divcomponent.mix; %example.mix; %footnote.mix;
%glossdef.mix; %highlights.mix; %indexdivcomponent.mix;
%legalnotice.mix; %listpreamble.mix; %para.class;
%qandaset.mix; %refcomponent.mix; %revdescription.mix;
%sidebar.mix; %tabentry.mix; %tbl.entry.mdl;
%textobject.mix;

Description

A SimPara is a “simple paragraph,” one that may contain only character data and inline elements. The Para element is less restrictive; it may also contain block level structures (lists, figures, and so on).

Processing expectations

Formatted as a displayed block.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain simpara: `abstract`, `answer`, `appendix`, `article`, `authorblurb`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `epigraph`, `example`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `highlights`, `important`, `index`, `indexdiv`, `informalexample`, `itemizedlist`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `orderedlist`, `partintro`, `personblurb`, `preface`, `printhistory`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `variablelist`, `warning`.

Children

The following elements occur in simpara: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

See Also

`formalpara`, `para`.

Examples

```
<!DOCTYPE simpara PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<simpara>
Just the text, ma'am.
</simpara>
```

Just the text, ma'am. For additional examples, see also `para`.

■ simplelist

Name

simplelist – An undecorated list of single words or short phrases

Synopsis

Content Model

simplelist ::= (`member`+)

Attributes Common attributes

Name Type Default

columns CDATA *None*
horiz
type inline
vert
"vert"

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%footnote.mix;`
`%glossdef.mix;` `%highlights.mix;` `%indexdivcomponent.mix;`
`%legalnotice.mix;` `%list.class;` `%para.mix;`
`%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;`
`%sidebar.mix;` `%tabentry.mix;` `%tbl.entry.mdl;`
`%textobject.mix;`

Description

A `SimpleList` is a list of words or phrases. It offers a convenient alternative to the other list elements for inline content.

Processing expectations

Ironically, the processing expectations of a `SimpleList` are quite complex.

The presentation of a `SimpleList` is controlled by the Type attribute, which has three possible values:

`Inline` Indicates that the `Member`s of the list should be rendered as a comma separated, inline list.

`Horiz` Indicates that the `Member`s of the list should be rendered in a tabular fashion with members running across the rows.

`Vert` Indicates that the `Member`s of the list should be rendered in a tabular fashion with members running down the columns. This is the default.

In both of the tabular cases, the number of columns in the table is controlled by the Columns attribute.

Parents

These elements contain simplelist: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caption`, `caution`, `chapter`, `colophon`, `constraintdef`, `dedication`, `entry`, `example`, `footnote`, `glossary`, `glossdef`, `glossdiv`, `highlights`, `important`, `index`, `indexdiv`, `informalexample`, `legalnotice`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `textobject`, `tip`, `warning`.

Children

The following elements occur in simplelist: `member`.

Attributes

columns Columns specifies the number of columns to be used in the presentation of a `SimpleList` with a Type of `Vert` or `Horiz`. If Columns is unspecified, `1` is assumed.

type Type specifies how the `Member`s of the `SimpleList` are to be formatted. (This attribute would better fit DocBook semantics if it were named “Class”).

See Also

`calloutlist`, `itemizedlist`, `listitem`, `orderedlist`, `segmentedlist`, `variablelist`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Here is a <sgmltag>SimpleList</sgmltag>, rendered inline:
<simplelist type='inline'>
<member>A</member>
<member>B</member>
<member>C</member>
<member>D</member>
<member>E</member>
<member>F</member>
<member>G</member>
</simplelist>
</para>
```

Here is a `SimpleList`, rendered inline: A, B, C, D, E, F, G

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Here is the same <sgmltag>SimpleList</sgmltag> rendered horizontally with
three columns:
<simplelist type='horiz' columns='3'>
<member>A</member>
<member>B</member>
<member>C</member>
<member>D</member>
<member>E</member>
<member>F</member>
<member>G</member>
</simplelist>
</para>
```

Here is the same `SimpleList` rendered horizontally with three columns:

A	B	C
D	E	F
G		

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Finally, here is the list rendered vertically:
<simplelist type='vert' columns='3'>
<member>A</member>
<member>B</member>
<member>C</member>
<member>D</member>
<member>E</member>
<member>F</member>
<member>G</member>
</simplelist>
</para>
```

Finally, here is the list rendered vertically:

A	D	G
B	E	
C	F	

■ simplemsgentry

Name

simplemsgentry – A wrapper for a simpler entry in a message set

Synopsis

Content Model

simplemsgentry ::= (`msgtext`,`msgexplan`+)

Attributes Common attributes

Name Type Default

origin CDATA *None*

audience CDATA *None*

level CDATA *None*

Description

`SimpleMsgEntry` is a simpler alternative to `MsgEntry`. In a `MsgSet`, each `SimpleMsgEntry` contains the text of a message and its explanation.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain simplemsgentry: `msgset`.

Children

The following elements occur in simplemsgentry: `msgexplan`, `msgtext`.

Attributes

audience Identifies the audience to which this message is relevant.

level Identifies the level of importance or severity of this message.

origin Indicates the origin of this message.

■ simplesect

Name

simplesect – A section of a document with no subdivisions

Synopsis

Content Model

simplesect ::= ((title,subtitle?,titleabbrev?), (calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset|constraintdef|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)+)

Attributes Common attributes

Parameter Entities

%bookcomponent.content;

Description

SimpleSect is one of the top-level sectioning elements in a component. There are three types of sectioning elements in DocBook:

- Explicitly numbered sections, Sect1...Sect5, which must be properly nested and can only be five levels deep.
- Recursive Sections, which are alternative to the numbered sections and have unbounded depth.
- SimpleSects, which are terminal. SimpleSects can occur as the "leaf" sections in either recursive sections or any of the numbered sections, or directly in components.

SimpleSects may be more convenient than numbered sections in some authoring environments because they can be moved around in the document hierarchy without renaming.

None of the sectioning elements is allowed to "float" in a component. You can place paragraphs and other block elements before a section, but you cannot place anything after it.

Processing expectations

Formatted as a displayed block. Sometimes sections are numbered.

Parents

These elements contain simplesect: appendix, article, chapter, partintro, preface, sect1, sect2, sect3, sect4, sect5, section.

Children

The following elements occur in simplesect: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, subtitle, synopsis, table, tip, title, titleabbrev, variablelist, warning.

See Also

bridgehead, sect1, sect2, sect3, sect4, sect5, section.

Examples

```
<!DOCTYPE sect1 PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<sect1><title>Additional Coding</title>

<para>
Support for the additional features requested will be provided.
</para>

<simplesect><title>Estimated Time</title>

<para>
2 to 3 weeks.
</para>

</simplesect>
</sect1>
```

For additional examples, see also `productionset`.

■ spanspec

Name

spanspec – Formatting information for a spanned column in a table

Synopsis

Content Model

spanspec ::= EMPTY

Attributes

Name Type Default

nameend CDATA *Required*

char CDATA *None*

colsep CDATA *None*

charoff CDATA *None*

spanname CDATA *Required*

namest CDATA *Required*

rowsep CDATA *None*

align (center | char | justify | left | right) *None*

Parameter Entities

`%tbl.entrytbl.mdl;` `%tbl.tgroup.mdl;`

Description

A `SpanSpec` associates a name with a span between two columns in a table. In the body of the table, cells can refer to the span by name. Cells that refer to a span will span horizontally from the first column to the last column, inclusive.

Cells can also form spans directly, by naming the start and end columns themselves. The added benefit of a SpanSpec is that it can associate formatting information (such as alignment and table rule specifications) with the span. This information does not need to be repeated then, on each spanning cell.

Processing expectations

Suppressed. This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* http://www.oasis-open.org/html/a502.htm.

The NameSt and NameEnd attributes of a SpanSpec must refer to named ColSpecs in the same table. In other words, if the SpanSpec

```
<spanspec spanname="fullyear" namest="jan" nameend="dec"/>
```

exists in a table, ColSpecs named “jan” and “dec” must also exist in the same table.

Parents

These elements contain spanspec: entrytbl, tgroup.

Attributes

align Align specifies the horizontal alignment of Entrys (or EntryTbls) in the span. The default alignment is inherited from the enclosing TGroup. If Char is specified, see also Char and CharOff. Individual Entrys and EntryTbls can specify an alternate alignment.

char Char specifies the alignment character when the Align attribute is set to Char.

charoff CharOff specifies the percentage of the column’s total width that should appear to the left of the first occurance of the character identified in Char when the Align attribute is set to Char. This attribute is inherited from the ColSpec of the column specified in Namest or from the enclosing TGroup.

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of the spanning column described by this SpanSpec. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute of the enclosing Table or InformalTable and the ColSep of the last column in the table is ignored. If unspecified, this attribute is inherited from enclosing elements.

nameend NameEnd is the name (defined in a ColSpec) of the rightmost column of the span.

namest NameSt is the name (defined in a ColSpec) of the leftmost column of the span.

rowsep If RowSep has the value 1 (true), then a rule will be drawn below all the cells in this Span. A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing Table or InformalTable and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements.

spanname SpanName specifies a name by which subsequent Entry and EntryTbls can refer to the span defined in this SpanSpec.

See Also

colspec, entry, entrytbl, informaltable, row, table, tbody, tfoot, tgroup, thead.

■ state

Name

state – A state or province in an address

Synopsis

Mixed Content Model

state ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Description

A State is the name or postal abbreviation for a state (or province) in an Address.

Processing expectations

Formatted inline. In an Address, this element may inherit the verbatim qualities of an address.

Parents

These elements contain state: address.

Children

The following elements occur in state: emphasis, indexterm, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, trademark, ulink.

See Also

address, city, country, email, fax, otheraddr, phone, pob, postcode, street.

Examples

For examples, see address, otheraddr.

■ step

Name

step – A unit of action in a procedure

Synopsis

Content Model

step ::= (title?, (((calloutlist|glosslist|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|msgset|procedure|sidebar|qandaset|productionset|constraintdef|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)+, (substeps, (calloutlist|glosslist|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| msgset|procedure|sidebar|qandaset|productionset| constraintdef| anchor|bridgehead|remark|highlights|abstract| authorblurb|epigraph|indexterm| beginpage)*)?)| (substeps, (calloutlist|glosslist|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|msgset|procedure| sidebar|qandaset|productionset| constraintdef|anchor|bridgehead|remark|highlights| abstract| authorblurb|epigraph|indexterm|beginpage)*)))

Attributes Common attributes

Name Type Default

performance optional
required
"required"

Description

A Step identifies a unit of action in a Procedure. If a finer level of granularity is required for some steps, you can embed SubSteps in a Step. Embedded SubSteps contain Steps, so that substeps can be nested to any depth.

Processing expectations

Formatted as a displayed block. Steps are almost always numbered.

Parents

These elements contain step: procedure, substeps.

Children

The following elements occur in step: abstract, address, anchor, authorblurb, beginpage, blockquote, bridgehead, calloutlist, caution, classsynopsis, cmdsynopsis, constraintdef, constructorsynopsis, destructorsynopsis, epigraph, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, msgset, note, orderedlist, para, procedure, productionset, programlisting, programlistingco, qandaset, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, substeps, synopsis, table, tip, title, variablelist, warning.

Attributes

performance Performance specifies whether particular Step in a Procedure must be performed or is optional.

Examples

For examples, see procedure.

■ street

Name

street – A street address in an address

Synopsis

Mixed Content Model

street ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`| `subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

In postal addresses, the `Street` element contains the street address portion of the `Address`. If an address contains more than one line of street address information, each line should appear in its own `Street`.

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain street: `address`.

Children

The following elements occur in street: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`address`, `city`, `country`, `email`, `fax`, `otheraddr`, `phone`, `pob`, `postcode`, `state`.

Examples

For examples, see `address`, `otheraddr`, `publisher`.

■ structfield

Name

structfield – A field in a structure (in the programming language sense)

Synopsis

Mixed Content Model

structfield ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`| `beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

A `StructField` is a wrapper for the name of a field in a `struct` (a syntactic element of the C programming language) or a field in an equivalent construct in another programming language.

Processing expectations

Formatted inline.

Parents

These elements contain structfield: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in structfield: `beginpage`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `replaceable`.

See Also

`classname`, `interface`, `property`, `structname`, `symbol`, `token`, `type`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <structfield>tm_isdst</structfield> field is non-zero when
the time reported is in daylight savings time.
</para>
```

The *`tm_isdst`* field is non-zero when the time reported is in daylight savings time.

■ structname

Name

structname – The name of a structure (in the programming language sense)

Synopsis

Mixed Content Model

structname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

StructName is an inline wrapper for the name of a struct (a syntactic element of the C programming language) or an equivalent construct in another programming language.

Processing expectations

Formatted inline.

Parents

These elements contain structname: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in structname: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

classname, interface, property, structfield, symbol, token, type.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <structname>tm</structname> structure, returned by
<function>_get_tm</function>, contains complete information
about the current time of day.
</para>
```

The tm structure, returned by _get_tm, contains complete information about the current time of day. For additional examples, see also refentry.

■ subject

Name

subject – One of a group of terms describing the subject matter of a document

Synopsis

Content Model

subject ::= (subjectterm+)

Attributes Common attributes

Name Type Default

weight CDATA *None*

Description

A “subject” categorizes or describes the topic of a document, or section of a document. In DocBook, a `Subject` is defined by the `SubjectTerm`s that it contains.

Subject terms should be drawn from a controlled vocabulary, such as the *Library of Congress Subject Headings*. If an outside vocabulary is not appropriate, a local or institutional subject set should be created.

The advantage of a controlled vocabulary is that it places the document into a known subject space. Searching the subject space with a particular subject term will find *all* of the documents that claim to have that subject. There’s no need to worry about terms that are synonymous with the search item, or homophones of the search term.

All of the `SubjectTerm`s in a `Subject` should describe the same subject, and be from the *same controlled vocabulary*.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Subjects are rarely displayed to a reader. Usually, they are reserved for searching and retrieval purposes.

Unlike `Keyword`s, which may be chosen freely, subject terms should come from a controlled vocabulary.

In order to assure that typographic or other errors are not introduced into the subject terms, they should be compared against the controlled vocabulary by an external process.

Parents

These elements contain subject: `subjectset`.

Children

The following elements occur in subject: `subjectterm`.

Attributes

weight Weight specifies a ranking for this `Subject` relative to other subjects in the same set.

See Also

`keyword`, `keywordset`, `subjectset`, `subjectterm`.

Examples

For examples, see `subjectset`.

■ subjectset

Name

subjectset – A set of terms describing the subject matter of a document

Synopsis

Content Model

subjectset ::= (`subject`+)

Attributes Common attributes

Name Type Default

scheme NMTOKEN *None*

Parameter Entities

```
%info.class;
```

Description

A SubjectSet is a container for a set of Subjects. All of the Subjects within a SubjectSet should come from the *same* controlled vocabulary.

A document can be described using terms from more than one controlled vocabulary. In order to do this, you should use the Scheme attribute to distinguish between controlled vocabularies.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Subjects are rarely displayed to a reader. Usually, they are reserved for searching and retrieval purposes.

DocBook does not specify a relationship between SubjectSets in different parts of a document or between a SubjectSet and the SubjectSets of enclosing parts of the document.

Parents

These elements contain subjectset: appendixinfo, articleinfo, bibliographyinfo, blockinfo, bookinfo, chapterinfo, glossaryinfo, indexinfo, objectinfo, partinfo, prefaceinfo, refentryinfo, referenceinfo, refsect1info, refsect2info, refsect3info, refsectioninfo, refsynopsisdivinfo, sect1info, sect2info, sect3info, sect4info, sect5info, sectioninfo, setindexinfo, setinfo, sidebarinfo.

Children

The following elements occur in subjectset: subject.

Attributes

scheme Scheme identifies the controlled vocabulary used by this SubjectSet's terms.

See Also

keyword, keywordset, subject, subjectterm.

Examples

```
<!DOCTYPE subjectset PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<subjectset scheme="libraryofcongress">
<subject>
  <subjectterm>Electronic Publishing</subjectterm>
</subject>
<subject>
  <subjectterm>SGML (Computer program language)</subjectterm>
</subject>
</subjectset>
```

■ subjectterm

Name

subjectterm – A term in a group of terms describing the subject matter of a document

Synopsis

Mixed Content Model

subjectterm ::= (#PCDATA)

Attributes Common attributes

Description

A SubjectTerm is an individual subject word or phrase that describes the subject matter of a document or the portion of a document in which it occurs.

Subject terms are not expected to contain any markup. They are external descriptions from a controlled vocabulary.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Subject terms are rarely displayed to a reader. Usually, they are reserved for searching and retrieval purposes.

Parents

These elements contain subjectterm: subject.

See Also

keyword, keywordset, subject, subjectset.

Examples

For examples, see subjectset.

■ subscript

Name

subscript – A subscript (as in H_2O, the molecular formula for water)

Synopsis

Mixed Content Model

subscript ::= (#PCDATA|link|olink|ulink|emphasis|replaceable|symbol| inlinegraphic| inlinemediaobject|anchor|remark|subscript| superscript)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %ndxterm.char.mix;

%other.char.class; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix; %word.char.mix;

Description

Subscript identifies text that is to be displayed as a subscript when rendered.

Processing expectations

Formatted inline. Subscripts are usually printed in a smaller font and shifted down with respect to the baseline.

Parents

These elements contain subscript: abbrev, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, fax, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, holder, honorific, interfacename,

invpartnumber, isbn, issn, issuenum, jobtitle, keycap, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, quote, refentrytitle, refmiscinfo, refpurpose, releaseinfo, remark, replaceable, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, shortaffil, simpara, state, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, volumenum, wordasword, year.

Children

The following elements occur in subscript: anchor, emphasis, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, symbol, ulink.

See Also

equation, informalequation, inlineequation, superscript.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
Thirsty?  Have some H<subscript>2</subscript>O.
</para>
```

Thirsty? Have some H_2O.

■ substeps

Name

substeps – A wrapper for steps that occur within steps in a procedure

Synopsis

Content Model

substeps ::= (step+)

Attributes Common attributes

Name	Type	Default
performance	optional required	"required"

Description

A Procedure describes a sequence of Steps that a reader is expected to perform. If a finer level of granularity is required for some steps, you can use SubSteps to embed substeps within a Step.

SubSteps contain Steps, so substeps can be nested to any depth.

Processing expectations

Formatted as a displayed block. SubSteps are almost always numbered.

Parents

These elements contain substeps: step.

Children

The following elements occur in substeps: `step`.

Attributes

performance Performance specifies whether particular set of `Substep` in a `Procedure` must be performed or is optional.

Examples

For examples, see `procedure`.

■ subtitle

Name

subtitle – The subtitle of a document

Synopsis

Mixed Content Model

subtitle ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`| `citetitle`| `emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|`trademark`| `wordasword`|`personname`| `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`|`constant`|`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|`property`| `replaceable`|`returnvalue`| `sgmltag`|`structfield`|`structname`|`symbol`| `systemitem`|`token`|`type`|`userinput`|`varname`| `nonterminal`|`anchor`| `author`|`authorinitials`|`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|`subscript`| `superscript`|`inlinegraphic`| `inlinemediaobject`|`inlineequation`| `indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%bookcomponent.title.content;` `%div.title.content;` `%info.class;` `%refsect.title.content;` `%sect.title.content;`

Description

A `Subtitle` identifies the subtitle of a document, or portion of a document.

Processing expectations

Formatted as a displayed block.

Parents

These elements contain subtitle: `appendix`, `appendixinfo`, `article`, `articleinfo`, `bibliodiv`, `biblioentry`, `bibliography`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `book`, `bookinfo`, `chapter`, `chapterinfo`, `colophon`, `dedication`, `glossary`, `glossaryinfo`, `glossdiv`, `index`, `indexdiv`, `indexinfo`, `lot`, `objectinfo`, `part`, `partinfo`, `partintro`, `preface`, `prefaceinfo`, `refentryinfo`, `reference`, `referenceinfo`, `refsect1`, `refsect1info`, `refsect2`, `refsect2info`, `refsect3`, `refsect3info`, `refsection`, `refsectioninfo`, `refsynopsisdiv`, `refsynopsisdivinfo`, `sect1`, `sect1info`, `sect2`, `sect2info`, `sect3`, `sect3info`, `sect4`, `sect4info`, `sect5`, `sect5info`,

section, sectioninfo, set, setindex, setindexinfo, setinfo, sidebarinfo, simplesect, toc.

Children

The following elements occur in subtitle: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

See Also

title, titleabbrev.

Examples

For examples, see bibliography, sect1info, section.

■ superscript

Name

superscript – A superscript (as in x^2, the mathematical notation for x multiplied by itself)

Synopsis

Mixed Content Model

superscript ::= (#PCDATA|link|olink|ulink|emphasis|replaceable|symbol|inlinegraphic|inlinemediaobject|anchor|remark|subscript|superscript)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %ndxterm.char.mix;

%other.char.class; %para.char.mix; %refinline.char.mix;

%tbl.entry.mdl; %title.char.mix; %word.char.mix;

Description

Superscript identifies text that is to be displayed as a superscript when rendered.

Processing expectations

Formatted inline. Superscripts are usually printed in a smaller font and shifted up with respect to the baseline.

Parents

These elements contain superscript: abbrev, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, fax, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, holder, honorific, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, quote, refentrytitle, refmiscinfo, refpurpose, releaseinfo, remark, replaceable, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, shortaffil, simpara, state, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, volumenum, wordasword, year.

Children

The following elements occur in superscript: anchor, emphasis, inlinegraphic, inlinemediaobject, link, olink, remark, replaceable, subscript, superscript, symbol, ulink.

See Also

equation, informalequation, inlineequation, subscript.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The equation e<superscript>&pi;i</superscript> + 1 = 0 ties together
five of the most important mathematical constants.
</para>
```

The equation $e^{i} + 1 = 0$ ties together five of the most important mathematical constants. For additional examples, see also inlinemediaobject.

■ surname

Name

surname – A family name; in western cultures the “last name”

Synopsis

Mixed Content Model

surname ::= (#PCDATA|link|olink|ulink|emphasis|trademark|replaceable|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm)*

Attributes Common attributes

Parameter Entities

%bibliocomponent.mix; %info.class; %person.ident.mix;

Description

A Surname is a family name; in Western cultures, the “last name.”

Processing expectations

Formatted inline. In an `Address`, this element may inherit the verbatim qualities of an address.

Parents

These elements contain surname: `address`, `appendixinfo`, `articleinfo`, `author`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `editor`, `glossaryinfo`, `indexinfo`, `objectinfo`, `othercredit`, `partinfo`, `personname`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in surname: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`affiliation`, `firstname`, `honorific`, `lineage`, `othername`.

Examples

For examples, see `ackno`, `article`, `author`, `authorgroup`, `bibliography`, `bibliomset`, `biblioset`, `book`, `bookinfo`, `othercredit`, `personname`.

■ svg:svg

Name

svg:svg – An SVG graphic

Synopsis

If the SVG Module is used, `ImageObject`s can include `svg:svg` graphics.

Description

The Scalable Vector Graphics `http://www.w3.org/TR/SVG/` Recommendation from the W3C defines the features and syntax for SVG, a language for describing two-dimensional vector and mixed vector/raster graphics in XML.

A complete description of SVG is outside the scope of this reference.

Examples

```
<!DOCTYPE mediaobject PUBLIC "-//OASIS//DTD DocBook SVG Module V1.0beta1//EN"
              "dbsvg.dtd">
<mediaobject>
<imageobject>
<svg:svg/>
</imageobject>
</mediaobject>
```

■ symbol

Name

symbol – A name that is replaced by a value before processing

Synopsis

Mixed Content Model

symbol ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

class limit
None

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

A Symbol is a name that represents a value. It should be used in contexts in which the name will actually be replaced by a value before processing. The canonical example is a #defined symbol in a C program where the C preprocessor replaces every occurance of the symbol with its value before compilation begins.

The Limit value of the Class attribute identifies those symbols that represent system limitations (for example, the number of characters allowed in a path name or the largest possible positive integer). DocBook V3.1 introduced the Constant element, which may be more suitable for some of these symbols.

Processing expectations

Formatted inline.

Parents

These elements contain symbol: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subscript, subtitle, superscript, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in symbol: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

Attributes

class Class indicates the type of Symbol.

See Also

classname, interface, property, structfield, structname, token, type.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
No filename may be more than <symbol class='limit'>MAXPATHLEN</symbol>
characters long.
</para>
```

No filename may be more than MAXPATHLEN characters long.

■ synopfragment

Name

synopfragment – A portion of a `CmdSynopsis` broken out from the main body of the synopsis

Synopsis

Content Model

synopfragment ::= ((`arg`|`group`)+)

Attributes Common attributes

Description

A complex `CmdSynopsis` can be made more manageable with `SynopFragments`. Rather than attempting to present the entire synopsis in one large piece, parts of the synopsis can be extracted out and presented elsewhere. These extracted pieces are placed in `SynopFragments` at the end of the `CmdSynopsis`.

At the point in which each piece was extracted, insert a `SynopFragmentRef` that points to the fragment. The content of the reference element will be presented inline.

Processing expectations

Formatted as a displayed block.

The presentation system is responsible for generating text that makes the reader aware of the link. This can be done with numbered bullets, or any other appropriate mechanism. Whatever mark is generated for the reference must also be generated for the fragment.

Online systems have additional flexibility. They may generate hot links between the references and the fragments, for example, or place the fragments in pop-up windows.

Parents

These elements contain synopfragment: `cmdsynopsis`.

Children

The following elements occur in synopfragment: `arg`, `group`.

See Also

`arg`, `cmdsynopsis`, `group`, `refsynopsisdiv`, `sbr`, `synopfragmentref`.

Examples

```
<!DOCTYPE cmdsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<cmdsynopsis>
  <command>cccp</command>
  <arg>-$</arg>
```

```
  <arg>-C</arg>
  <arg rep='repeat'>-D<replaceable>name</replaceable>
       <arg>=<replaceable>definition</replaceable></arg></arg>
  <arg>-dD</arg>
  <arg>-dM</arg>
  <sbr/>
  <arg rep='repeat'>-I <replaceable>directory</replaceable></arg>
  <arg>-H</arg>
  <arg>-I-</arg>
  <arg rep='repeat'>-imacros <replaceable>file</replaceable></arg>
  <sbr/>
  <arg rep='repeat'>-include <replaceable>file</replaceable></arg>
  <group>
    <synopfragmentref linkend="langs">languages</synopfragmentref>
  </group>
  <arg>-lint</arg>
  <sbr/>
  <group>
    <arg>-M</arg>
    <arg>-MD</arg>
    <arg>-MM</arg>
    <arg>-MMD</arg>
  </group>
  <arg>-nostdinc</arg>
  <arg>-P</arg>
  <arg>-pedantic</arg>
  <sbr/>
  <arg>-pedantic-errors</arg>
  <arg>-trigraphs</arg>
  <arg>-U<replaceable>name</replaceable></arg>
  <sbr/>
  <arg>-undef</arg>
  <arg choice="plain"><synopfragmentref linkend="warn">warnings
    </synopfragmentref></arg>
  <group choice='req'>
    <arg><replaceable>infile</replaceable></arg>
    <arg>-</arg>
  </group>
  <group choice='req'>
    <arg><replaceable>outfile</replaceable></arg>
    <arg>-</arg>
  </group>

  <synopfragment id="langs">
    <group choice="plain">
      <arg>-lang-c</arg>
      <arg>-lang-c++</arg>
      <arg>-lang-objc</arg>
    </group>
  </synopfragment>

  <synopfragment id="warn">
    <arg>-Wtrigraphs</arg>
    <arg>-Wcomment</arg>
    <arg>-Wall</arg>
    <arg>-Wtraditional</arg>
  </synopfragment>

</cmdsynopsis>
```

```
cccp [-$] [-C] [-Dname [=definition]...]  [-dD] [-dM] [-I directory...]
    [-H] [-I-] [-imacros file...]  [-include file...]  [ (1) ] [-lint] [-M
    | -MD | -MM | -MMD] [-nostdinc] [-P] [-pedantic] [-pedantic-errors]
    [-trigraphs] [-Uname] [-undef] (2)  infile | - outfile | -(1) -lang-c |
    -lang-c++ | -lang-objc (2) [-Wtrigraphs] [-Wcomment] [-Wall]
    [-Wtraditional]
```

■ synopfragmentref

Name

synopfragmentref – A reference to a fragment of a command synopsis

Synopsis

Mixed Content Model

synopfragmentref ::= (#PCDATA)

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

Description

A complex `CmdSynopsis` can be made more manageable with `SynopFragment`s. Rather than attempting to present the entire synopsis in one large piece, parts of the synopsis can be extracted out and presented elsewhere.

At the point where each piece was extracted, insert a `SynopFragmentRef` that points to the fragment. The content of the `SynopFragmentRef` will be presented inline.

The extracted pieces are placed in `SynopFragment`s at the end of the `CmdSynopsis`.

NOTE

The content model of `SynopFragmentRef` is unique in the SGML version of DocBook because it contains RCDATA declared content. What this means is that all markup inside a `SynopFragmentRef` is ignored, except for entity references.

How, you might ask, is this different from a content model that includes only #PCDATA? The difference is only apparent when you consider inclusions. Recall that an inclusion provides a list of elements that can occur *anywhere* inside an element. So, for example, the fact that `Chapter` lists `IndexTerm` as an inclusion means that `IndexTerm` can legally occur inside of a `SynopFragmentRef` that's nested inside a chapter, even if the content model of `SynopFragmentRef` does not explicitly allow `IndexTerm`s. Making the content RCDATA ensures that the markup will not be recognized, even if it's allowed by inclusion. A neat trick.

XML does not support RCDATA.

Processing expectations

Formatted as a displayed block.

The presentation system is responsible for generating text that makes the reader aware of the link. This can be done with numbered bullets, or any other appropriate mechanism.

Online systems have additional flexibility. They may generate hot links between the references and the fragments, for example, or place the fragments in pop-up windows.

Parents

These elements contain synopfragmentref: `arg`, `group`.

Attributes

linkend Linkend points to the `SynopFragment` referenced.

See Also

`arg`, `cmdsynopsis`, `group`, `refsynopsisdiv`, `sbr`, `synopfragment`.

Examples

For examples, see `synopfragment`.

synopsis

Name

synopsis – A general-purpose element for representing the syntax of commands or functions

Synopsis

Mixed Content Model

synopsis ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|
`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|
`trademark`|`wordasword`|`personname`|`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`|`constant`|
`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|
`property`| `replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|
`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`| `author`|`authorinitials`|
`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|
`subscript`| `superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`| `synopsis`|
`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`| `constructorsynopsis`|
`destructorsynopsis`|`methodsynopsis`|`indexterm`| `beginpage`|`graphic`|`mediaobject`|
`co`|`coref`|`textobject`|`lineannotation`)*

Attributes Common attributes

Name Type Default

format linespecific "linespecific"

linenumbering numbered | unnumbered

None

label CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;`
`%divcomponent.mix;` `%example.mix;` `%figure.mix;`
`%footnote.mix;` `%glossdef.mix;` `%indexdivcomponent.mix;`
`%listpreamble.mix;` `%para.char.mix;` `%qandaset.mix;`
`%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;`
`%synop.class;` `%tbl.entry.mdl;`

Description

A Synopsis is a verbatim environment for displaying command, function, and other syntax summaries.

Unlike CmdSynopsis and FuncSynopsis which have a complex interior structure, Synopsis is simply a verbatim environment.

Processing expectations

This element is displayed "verbatim"; whitespace and linebreaks within this element are significant. Synopsis elements are usually displayed in a fixed width font.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain synopsis: answer, appendix, application, article, attribution, bibliodiv, bibliography, bibliomisc, blockquote, callout, caution, chapter, citation, citetitle, constraintdef, emphasis, entry, example, figure, footnote, foreignphrase, glossary, glossdef, glossdiv, glosssee, glossseealso, glossterm, important, index, indexdiv, informalexample, informalfigure, itemizedlist, lineannotation, link, listitem, literallayout, lotentry, member, msgaud, msgexplan, msgtext, note, olink, orderedlist, para, partintro, phrase, preface, procedure, productname, programlisting, qandadiv, qandaset, question, quote, refentrytitle, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, remark, revdescription, screen, screeninfo, sect1, sect2, sect3, sect4, sect5, section, seg, setindex, sidebar, simpara, simplesect, step, synopsis, term, tip, tocback, tocentry, tocfront, ulink, variablelist, warning.

Children

The following elements occur in synopsis: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, co, command, computeroutput, constant, constructorsynopsis, coref, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, graphic, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, lineannotation, link, literal, markup, medialabel, mediaobject, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, textobject, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

format The Format attribute applies the linespecific notation to all synopses. All white space and line breaks must be preserved.

label Label specifies an identifying number or string that may be used in presentation.

linenumbering Line numbering indicates whether or not the lines of a Synopsis are to be automatically numbered. The details of numbering (every line or only selected lines, on the left or right, etc.) are left up to the processing application. Be aware that not all processors are capable of numbering lines.

See Also

cmdsynopsis, computeroutput, funcsynopsis, lineannotation, literallayout, programlisting, screen, screenshot, userinput.

Examples

```
<!DOCTYPE synopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<synopsis>
chgrp [-R [-H | -L | -P] [-f] group file...
</synopsis>
```

chgrp [-R [-H | -L | -P] [-f] group file...

```
<!DOCTYPE synopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<synopsis>
int max(int int1, int int2);
</synopsis>
```

int max(int int1, int int2); For additional examples, see also optional, refentry, reference.

■ systemitem

Name

systemitem – A system-related item or term

Synopsis

Mixed Content Model

systemitem ::= (#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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| remark|subscript|superscript|inlinegraphic|inlinemediaobject| indexterm|beginpage|acronym|co)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

constant
domainname
etheraddress
event
eventhandler
filesystem
fqdomainname
groupname
class ipaddress
library
macro
netmask
newsgroup
osname
resource
systemname
username

None

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

A `SystemItem` identifies any system-related item or term. The Class attribute defines a number of common system-related terms.

Many inline elements in DocBook are, in fact, system-related. Some of the objects identified by the Class attribute on `SystemItem` may eventually migrate out to be inline elements of their own accord... and vice versa.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information.

Future Changes

In DocBook V4.0, the content model of `SystemItem` will be constrained to `(#PCDATA | Replaceable | InlineGraphic)`.

Also, the `EnvironVar` and `Prompt` values of Class will be discarded (use `EnVar` and `Prompt` instead).

Future Changes

The `register` class value will be added to support CPU registers. The `constant` class value will be removed.

Parents

These elements contain systemitem: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

The following elements occur in systemitem: acronym, action, anchor, application, beginpage, classname, co, command, computeroutput, constant, database, email, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, function, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, parameter, prompt, property, remark, replaceable, returnvalue, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

class Class indicates the type of SystemItem.

moreinfo If MoreInfo is set to RefEntry, it implies that a RefEntry exists which further describes the SystemItem.

See Also

computeroutput, envar, filename, prompt, userinput.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
For many years, O'Reilly's primary web server,
<ulink url="http://www.oreilly.com/">http://www.oreilly.com/</ulink>,
was hosted by <application>WN</application> on
<systemitem class="systemname">helio.oreilly.com</systemitem>.
</para>
```

For many years, O'Reilly's primary web server, http://www.oreilly.com/, was hosted by WN on helio.oreilly.com. For additional examples, see also part.

■ table

Name

table – A formal table in a document

Synopsis

Content Model

table ::= ((blockinfo?, (title,titleabbrev?), (indexterm)*, textobject*, (graphic+|mediaobject+|tgroup+)))

Attributes Common attributes

Name Type Default

tocentry CDATA *None*

shortentry CDATA *None*

tabstyle CDATA *None*

orient land | port *None*

label CDATA *None*

colsep CDATA *None*

frame all | bottom | none | sides | top | topbot *None*

pgwide CDATA *None*

rowsep CDATA *None*

Parameter Entities

`%admon.mix;` `%bookcomponent.content;` `%component.mix;` `%divcomponent.mix;` `%formal.class;` `%glossdef.mix;` `%para.mix;` `%qandaset.mix;` `%refcomponent.mix;` `%revdescription.mix;` `%sidebar.mix;` `%tbl.table.name;`

Description

The `Table` element identifies a formal table. DocBook uses the CALS table model, which describes tables geometrically using rows, columns, and cells.

Tables may include column headers and footers, but there is no provision for row headers.

Processing expectations

Formatted as a displayed block. This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`.

Future Changes

The OASIS Exchange Table Model `http://www.oasis-open.org/specs/tm9901.html` will replace the full OASIS Table Model.

Parents

These elements contain table: `answer`, `appendix`, `article`, `bibliodiv`, `bibliography`, `blockquote`, `callout`, `caution`, `chapter`, `constraintdef`, `glossary`, `glossdef`, `glossdiv`, `important`, `index`, `listitem`, `msgexplan`, `msgtext`, `note`, `para`, `partintro`, `preface`, `procedure`, `qandadiv`, `qandaset`, `question`, `refsect1`, `refsect2`, `refsect3`, `refsection`, `refsynopsisdiv`, `revdescription`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `setindex`, `sidebar`, `simplesect`, `step`, `tip`, `warning`.

Children

The following elements occur in table: `blockinfo`, `graphic`, `indexterm`, `mediaobject`, `textobject`, `tgroup`, `title`, `titleabbrev`.

Attributes

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of all columns in this table. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute, not the ColSep.

frame Frame specifies how the table is to be framed:

Value	Meaning
`all`	Frame all four sides of the table. In some environments with limited control over table border formatting, such as HTML,
`bottom`	Frame only the bottom of the table.
`none`	Place no border on the table. In some environments with limited control over table border formatting, such as HTML, this
`sides`	Frame the left and right sides of the table.
`top`	Frame the top of the table.
`topbot`	Frame the top and bottom of the table.

There is no way to obtain a border on only the starting edge (left, in left-to-right writing systems) of the table.

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

orient Orient specifies the orientation of the `Table`. An orientation of `Port` is the "upright", the same orientation as the rest of the text flow. An orientation of `Land` is 90 degrees counterclockwise from the upright orientation.

pgwide If Pgwide has the value 0 (false), then the `Table` is rendered in the current text flow (with flow column width). A value of 1 (true) specifies that the table should be rendered across the full text page.

rowsep If RowSep has the value 1 (true), then a rule will be drawn below all the rows in the `Table` (unless other, interior elements, suppress some or all of the rules). A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute and the RowSep of the last row is ignored.

shortentry If `ShortEntry` has the value 1 (true), then the `Table`'s `TitleAbbrev` will be used in the `LoT`, `Index`, etc. A value of 0 (false) indicates that the full `Title` should be used in those places.

tabstyle TabStyle holds the name of a table style defined in a stylesheet (e.g., a FOSI) that will be used to process this document.

tocentry If ToCEntry has the value 1 (true), then the `Table` will appear in a generated List of Tables. The default value of 0 (false) indicates that it will not.

See Also

`colspec`, `entry`, `entrytbl`, `equation`, `example`, `figure`, `informalequation`, `informalexample`, `informalfigure`, `informaltable`, `row`, `spanspec`, `tbody`, `tfoot`, `tgroup`, `thead`.

Examples

```
<!DOCTYPE table PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<table frame='all'><title>Sample Table</title>
<tgroup cols='5' align='left' colsep='1' rowsep='1'>
```

```
<colspec colname='c1'/>
<colspec colname='c2'/>
<colspec colname='c3'/>
<colspec colnum='5' colname='c5'/>
<thead>
<row>
  <entry namest="c1" nameend="c2" align="center">Horizontal Span</entry>
  <entry>a3</entry>
  <entry>a4</entry>
  <entry>a5</entry>
</row>
</thead>
<tfoot>
<row>
  <entry>f1</entry>
  <entry>f2</entry>
  <entry>f3</entry>
  <entry>f4</entry>
  <entry>f5</entry>
</row>
</tfoot>
<tbody>
<row>
  <entry>b1</entry>
  <entry>b2</entry>
  <entry>b3</entry>
  <entry>b4</entry>
  <entry morerows='1' valign='middle'><para>  <!-- Pernicous Mixed Content -->
  Vertical Span</para></entry>
</row>
<row>
  <entry>c1</entry>
  <entry namest="c2" nameend="c3" align='center' morerows='1' valign='bottom'>Span Bot
  <entry>c4</entry>
</row>
<row>
  <entry>d1</entry>
  <entry>d4</entry>
  <entry>d5</entry>
</row>
</tbody>
</tgroup>
</table>
```

Horizontal Span		a3	a4	a5
b1	b2	b3	b4	Vertical Span
c1	Span Both		c4	
d1	d4	d5		
f1	f2	f3	f4	f5

■ tbody

Name

tbody – A wrapper for the rows of a table or informal table

Synopsis

Content Model

tbody ::= (row+)

Attributes Common attributes

Name Type Default

valign	bottom middle top	*None*

Parameter Entities

%tbl.entrytbl.mdl; %tbl.tgroup.mdl;

Description

The TBody wrapper identifies the Rows of a table that form the body of the table, as distinct from the header (THead) and footer (TFoot) rows.

In most tables, the TBody contains most of the rows.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* http://www.oasis-open.org/html/a502.htm.

Parents

These elements contain tbody: entrytbl, tgroup.

Children

The following elements occur in tbody: row.

Attributes

valign VAlign specifies the vertical alignment of text (and other elements) within the cells of this TBody.

See Also

colspec, entry, entrytbl, informaltable, row, spanspec, table, tfoot, tgroup, thead.

Examples

For examples, see entrytbl, footnoteref, informaltable, table.

■ term

Name

term – The word or phrase being defined or described in a variable list

Synopsis

Mixed Content Model

term ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Description

The Term in a VarListEntry identifies the thing that is described or defined by that entry.

Processing expectations

Formatted as a displayed block. Terms are usually formatted to make them stand out with respect to the text that follows. The best presentation depends on several factors, including the number and length of the terms. The TermLength attribute on the containing VariableList may influence the presentation of Terms. See VariableList.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain term: varlistentry.

Children

The following elements occur in term: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Examples

For examples, see variablelist.

■ tertiary

Name

tertiary – A tertiary word or phrase in an index term

Synopsis

Mixed Content Model

tertiary ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject)*

Attributes Common attributes

Name Type Default

sortas CDATA *None*

Description

`Tertiary` contains a third-level word or phrase in an `IndexTerm`. The text of a `Tertiary` term is less significant than the `Primary` and `Secondary` terms for sorting and display purposes.

DocBook does not define any additional levels. You cannot use `IndexTerm`s to construct indexes with more than three levels without extending the DTD.

In `IndexTerm`s, you can only have one primary, secondary, and tertiary term. If you want to index multiple tertiary terms for the same primary and secondary, you must repeat the primary and secondary in another `IndexTerm`. You cannot place several `Tertiary`s in the same primary.

Processing expectations

Suppressed. This element provides data for processing but is not rendered in the primary flow of text.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain tertiary: `indexterm`.

Children

The following elements occur in tertiary: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `citation`, `citerefentry`, `citetitle`, `classname`, `command`, `computeroutput`, `constant`, `corpauthor`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`,

quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

sortas SortAs specifies the string by which the element's content is to be sorted. If unspecified, the proper content is used.

See Also

indexentry, indexterm, primary, primaryie, secondary, secondaryie, see, seealso, seealsoie, seeie, tertiaryie.

■ tertiaryie

Name

tertiaryie – A tertiary term in an index entry, rather than in the text

Synopsis

Mixed Content Model

tertiaryie ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote| trademark|wordasword|personname|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|constant| markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt| property| replaceable|returnvalue|sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|nonterminal|anchor| author|authorinitials| corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark| subscript| superscript|inlinegraphic|inlinemediaobject)*

Attributes Common attributes

Name Type Default

linkends IDREFS *None*

Description

TertiaryIE identifies a third-level word or words in an IndexEntry.

In IndexEntrys, you can specify as many tertiary terms that are necessary. Secondary and tertiary terms can be mixed, following the primary.

Processing expectations

Formatted as a displayed block. TertiaryIEs occur below the SecondaryIE, and are usually aligned with each other and indented from the secondary.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain tertiaryie: indexentry.

Children

The following elements occur in tertiaryie: abbrev, acronym, action, anchor, application, author, authorinitials, citation, citerefentry, citetitle, classname, command, computeroutput, constant, corpauthor, database, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, filename, firstterm, footnote, footnoteref, foreignphrase, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkends Linkends, if used, points to the IndexTerms indexed by this entry.

See Also

indexentry, indexterm, primary, primaryie, secondary, secondaryie, see, seealso, seealsoie, seeie, tertiary.

■ textdata

Name

textdata – Pointer to external text data

Synopsis

Content Model

textdata ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

BMP
CGM-BINARY
CGM-CHAR
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
IGES
format JPEG
JPG
linespecific
PCX
PIC
PNG
PS
SGML
SVG
TBL
TEX
TIFF
WMF
WPG
None

entityref ENTITY *None*

fileref CDATA *None*

encoding CDATA *None*

Description

This element points to an external entity containing text to be inserted.

NOTE

Any markup in the text identified with `textdata` will be escaped. In other words, this element is for inserting things like program listings, not parts of an XML document. To break a document into separate physical files, use entities.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Insert the text.

There are two ways to provide content for `textdata`: entityref or fileref. It is best to use only one of these methods, however, if multiple sources are provided, entityref will be used in favor of fileref.

Parents

These elements contain textdata: `textobject`.

Attributes

encoding Identifies the encoding of the text in the external file. Recommended best practice suggests that character encodings registered (as charsets) `ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets` with the Internet Assigned Numbers Authority be referred to using their registered names.

entityref EntityRef identifies the general entity which contains (or points to) the external text.

fileref FileRef specifies the name of the file which contains the external text.

format Format identifies the format of the external text. The Format must be a defined notation.

srccredit SrcCredit contains details about the source of the external text.

■ textobject

Name

textobject – A wrapper for a text description of an object and its associated meta-information

Synopsis

Content Model

textobject ::= (`objectinfo`?, (`phrase`|`textdata`| (`calloutlist`|`glosslist`|`itemizedlist`|`orderedlist`|`segmentedlist`| `simplelist`|`variablelist`|`caution`|`important`|`note`|`tip`|`warning`| `literallayout`|`programlisting`|`programlistingco`|`screen`|`screenco`|`screenshot`|`formalpara`|`para`|`simpara`|`blockquote`)+))

Attributes Common attributes

Parameter Entities

`%mediaobject.mix;` `%tbl.table.mdl;`

Description

A `TextObject` is a wrapper containing a textual description of a media object, and its associated meta-information. `TextObject`s are only allowed in `MediaObject`s as a fall-back option, they cannot be the primary content.

There are two different forms of `TextObject`, and it is not unreasonable for a media object to contain both of them.

In the first form, the content of a `TextObject` is simply a `Phrase`. This form is a mechanism for providing a simple “alt text” for a media object. The phrase might be used, for example, as the value of the ALT attribute on an HTML `IMG`, with the primary content of the image coming from one of the other objects in the media object.

In the second form, the content of `TextObject` is a longer, prose description. This form could be used when rendering to devices that are incapable of displaying any of the other alternatives.

Processing expectations

May be formatted inline or as a displayed block, depending on context. It might not be rendered at all, depending on its placement within a `MediaObject` or `InlineMediaObject` and the constraints on the publishing system. For a more detailed description of the semantics involved, see `MediaObject`.

Parents

These elements contain textobject: `classsynopsisinfo`, `funcsynopsisinfo`, `informaltable`, `inlinemediaobject`, `literallayout`, `mediaobject`, `mediaobjectco`, `programlisting`, `screen`, `synopsis`, `table`.

Children

The following elements occur in textobject: `blockquote`, `calloutlist`, `caution`, `formalpara`, `glosslist`, `important`, `itemizedlist`, `literallayout`, `note`, `objectinfo`, `orderedlist`, `para`, `phrase`, `programlisting`, `programlistingco`, `screen`, `screenco`, `screenshot`, `segmentedlist`, `simpara`, `simplelist`, `textdata`, `tip`, `variablelist`, `warning`.

See Also

`alt`, `audioobject`, `caption`, `graphic`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `videoobject`.

Examples

For examples, see `audioobject`, `imageobject`, `informalfigure`, `inlinemediaobject`, `videoobject`.

■ tfoot

Name

tfoot – A table footer consisting of one or more rows

Synopsis

Content Model

tfoot ::= (`colspec*`,`row+`)

Attributes Common attributes

Name Type Default

valign	bottom middle top	*None*

Parameter Entities

```
%tbl.tgroup.mdl;
```

Description

The `TFoot` wrapper identifies the `Row`s of a table that form the foot of the table, as distinct from the header (`THead`) and body (`TBody`) rows.

Footer rows are always rendered at the end of the table, despite thier logical placement near the beginning.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`. Footer rows are often presented in an alternate typographic style, such as boldface.

In paged media, if a table spans across multiple pages, footer rows are printed on the bottom of each page.

In our experience, relatively few formatters handle footer rows correctly.

Parents

These elements contain tfoot: `tgroup`.

Children

The following elements occur in tfoot: `colspec`, `row`.

Attributes

valign VAlign specifies the vertical alignment of text (and other elements) within the cells of this `TFoot`.

See Also

`colspec`, `entry`, `entrytbl`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tgroup`, `thead`.

Examples

For examples, see `table`.

■ tgroup

Name

tgroup – A wrapper for the main content of a table, or part of a table

Synopsis

Content Model

tgroup ::= (`colspec*,spanspec*,thead?,tfoot?,tbody`)

Attributes Common attributes

Name Type Default

char CDATA *None*

charoff CDATA *None*

tgroupstyle CDATA *None*

cols CDATA *Required*

colsep CDATA *None*

rowsep CDATA *None*

align (center | char | justify | left | right) *None*

Parameter Entities

`%tbl.table-main.mdl; %tbl.table.mdl;`

Description

A `TGroup` surrounds a logically complete portion of a table. Most tables consist of a single `TGroup`, but complex tables with widely varying column specifications may be easier to code using multiple `TGroup`s.

The `TGroup` specifies the number of columns in the table, and contains all of the header, body, and footer rows, along with any additional column or span specifications necessary to express the geometry of the table.

Most of the properties of rows, columns, and cells inherit their default characteristics from the enclosing `TGroup`.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`.

Parents

These elements contain tgroup: `informaltable`, `table`.

Children

The following elements occur in tgroup: `colspec`, `spanspec`, `tbody`, `tfoot`, `thead`.

Attributes

align Align specifies the horizontal alignment of `Entry`s (or `EntryTbl`s) in cells of the `TGroup`. If `Char` is specified, see also Char and CharOff. Individual columns, spans, and cells can provide an alternate alignment.

char Char specifies the alignment character when the Align attribute is set to `Char`.

charoff CharOff specifies the percentage of the column's total width that should appear to the left of the first occurance of the character identified in Char when the Align attribute is set to `Char`.

cols Cols specifies the number of columns in the table.

colsep If ColSep has the value 1 (true), then a rule will be drawn to the right of all columns in this `TGroup`. A value of 0 (false) suppresses the rule. The rule to the right of the last column in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and not the ColSep.

rowsep If RowSep has the value 1 (true), then a rule will be drawn below all the rows in this `TGroup` (unless other, interior elements, suppress some or all of the rules). A value of 0 (false) suppresses the rule. The rule below the last row in the table is controlled by the Frame attribute of the enclosing `Table` or `InformalTable` and the RowSep of the last row is ignored. If unspecified, this attribute is inherited from enclosing elements.

tgroupstyle TGroupstyle holds the name of a table group style defined in a stylesheet (e.g., a FOSI) that will be used to process this document.

See Also

`colspec`, `entry`, `entrytbl`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `thead`.

Examples

For examples, see `entrytbl`, `footnoteref`, `informaltable`, `table`.

■ thead

Name

thead – A table header consisting of one or more rows

Synopsis

Content Model

thead ::= (`colspec*`,`row+`)

Attributes Common attributes

Name Type Default

valign	bottom middle top	*None*

Parameter Entities

`%tbl.entrytbl.mdl;` `%tbl.tgroup.mdl;`

Description

The `THead` wrapper identifies the `Row`s of a table that form the head of the table, as distinct from the body (`TBody`) and foot (`TFoot`) rows.

Header rows are always rendered at the beginning of the table.

Processing expectations

This element is expected to obey the semantics of the *CALS Table Model Document Type Definition*, as specified by *OASIS Technical Memorandum TM 9502:1995* `http://www.oasis-open.org/html/a502.htm`. Header rows are often presented in an alternate typographic style, such as boldface.

In paged media, if a table spans across multiple pages, header rows are printed at the top of each new page.

Parents

These elements contain thead: `entrytbl`, `tgroup`.

Children

The following elements occur in thead: `colspec`, `row`.

Attributes

valign VAlign specifies the vertical alignment of text (and other elements) within the cells of this `THead`.

See Also

`colspec`, `entry`, `entrytbl`, `informaltable`, `row`, `spanspec`, `table`, `tbody`, `tfoot`, `tgroup`.

Examples

For examples, see `table`.

■ tip

Name

tip – A suggestion to the user, set off from the text

Synopsis

Content Model

tip ::= (title?, (calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist|simplelist|variablelist|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|sidebar|anchor|bridgehead|remark|indexterm|beginpage)+)

Attributes Common attributes

Parameter Entities

%admon.class; %bookcomponent.content; %component.mix;

%divcomponent.mix; %highlights.mix; %legalnotice.mix;

%listpreamble.mix; %para.mix; %qandaset.mix;

%refcomponent.mix; %revdescription.mix; %sidebar.mix;

%tabentry.mix; %tbl.entry.mdl; %textobject.mix;

Description

A Tip is an admonition set off from the main text.

In some types of documentation, the semantics of admonitions are clearly defined (Caution might imply the possibility of harm to equipment whereas Warning might imply harm to a person), but DocBook makes no such assertions.

Processing expectations

Formatted as a displayed block. Often outputs the generated text “Tip” or some other visible indication of the type of admonition, especially if a Title is not present. Sometimes outputs a graphical icon or other symbol as well.

Parents

These elements contain tip: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, chapter, colophon, dedication, entry, glossary, glossdiv, highlights, index, itemizedlist, legalnotice, listitem, msgexplan, msgtext, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, variablelist.

Children

The following elements occur in tip: address, anchor, beginpage, blockquote, bridgehead, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayou mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, procedure, programlisting, programlistingco, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, title, variablelist.

See Also

caution, important, note, warning.

Examples

```
<!DOCTYPE tip PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<tip>
<para>
If you tie your shoelaces, you’re less likely to trip and
fall down.
</para>
</tip>
```

TIP

If you tie your shoelaces, you’re less likely to trip and fall down.

■ title

Name

title – The text of the title of a section of a document or of a formal block-level element

Synopsis

Mixed Content Model

title ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|indexterm)*

Attributes Common attributes

Name Type Default

pagenum CDATA *None*

Parameter Entities

%bibliocomponent.mix; %bookcomponent.title.content; %div.title.content; %formalobject.title.content; %info.class; %refsect.title.content; %sect.title.content; %tbl.table-titles.mdl; %tbl.table.mdl; %titles;

Description

Title is widely used in DocBook. It identifies the titles of documents and parts of documents, and is the required caption on formal objects. It is also allowed as an optional title or caption on many additional block elements.

Processing expectations

Formatted as a displayed block. Titles are often repeated in several locations, for example, at the location where the object occurs, in the *table of contents*, and in running headers and footers.

DocBook does not offer any mechanism for indicating where a line break should occur in long titles. Titles are often repeated and no single line break is likely to be correct in all of the places where a title is used. Instead, you will have to rely on your processing system to provide a mechanism, such as a processing instruction, for identifing the location of forced line breaks.

There are some contexts in which a `Title` can appear more than once. For example, it may appear in both `Book` and `BookInfo`:

```
<book><title>Some Book Title</title>
<bookinfo>
  <title>Some Book Title</title>
  <author><firstname>Some</firstname><surname>Author</surname>
    </author>
</bookinfo>
...
```

In these contexts, if the `Title` occurs more than once, *the same* title must be used in both places. It is an error to use different titles, although DocBook has no way to detect the error.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain title: `abstract`, `appendix`, `appendixinfo`, `article`, `articleinfo`, `authorblurb`, `bibliodiv`, `biblioentry`, `bibliography`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `blockquote`, `book`, `bookinfo`, `calloutlist`, `caution`, `chapter`, `chapterinfo`, `colophon`, `constraintdef`, `dedication`, `equation`, `example`, `figure`, `formalpara`, `glossary`, `glossaryinfo`, `glossdiv`, `important`, `index`, `indexdiv`, `indexinfo`, `itemizedlist`, `legalnotice`, `lot`, `msg`, `msgexplan`, `msgmain`, `msgrel`, `msgset`, `msgsub`, `note`, `objectinfo`, `orderedlist`, `part`, `partinfo`, `partintro`, `personblurb`, `preface`, `prefaceinfo`, `procedure`, `productionset`, `qandadiv`, `qandaset`, `refentryinfo`, `reference`, `referenceinfo`, `refsect1`, `refsect1info`, `refsect2`, `refsect2info`, `refsect3`, `refsect3info`, `refsection`, `refsectioninfo`, `refsynopsisdiv`, `refsynopsisdivinfo`, `sect1`, `sect1info`, `sect2`, `sect2info`, `sect3`, `sect3info`, `sect4`, `sect4info`, `sect5`, `sect5info`, `section`, `sectioninfo`, `segmentedlist`, `set`, `setindex`, `setindexinfo`, `setinfo`, `sidebar`, `sidebarinfo`, `simplesect`, `step`, `table`, `tip`, `toc`, `variablelist`, `warning`.

Children

The following elements occur in title: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `citation`, `citerefentry`, `citetitle`, `classname`, `command`, `computeroutput`, `constant`, `corpauthor`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

pagenum PageNum identifies the page on which this `Title` appears in some version of the printed document.

See Also

`subtitle`, `titleabbrev`.

Examples

```
<!DOCTYPE sect1 PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<sect1><title>A Sect One</title>
<para>
Actual content.
</para>
</sect1>
```

For additional examples, see also `abstract`, `appendix`, `article`, `audioobject`, `beginpage`, `bibliography`, `bibliomset`, `biblioset`, `book`, `bookinfo`, `caution`, `chapter`, `collab`, `contractsponsor`, `equation`, `example`, `figure`, `formalpara`, `glossary`, `graphic`, `highlights`, `html-form`, `index`, `indexterm`, `link`, `lot`, `note`, `olink`, `othercredit`, `para`, `part`, `procedure`, `productionset`, `qandaset`, `refentry`, `reference`, `remark`, `sect1info`, `section`, `segmentedlist`, `set`, `sidebar`, `simplesect`, `table`, `titleabbrev`, `variablelist`, `xref`.

■ titleabbrev

Name

titleabbrev – The abbreviation of a `Title`

Synopsis

Mixed Content Model

titleabbrev ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|
`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`| `phrase`|`quote`|
`trademark`|`wordasword`|`personname`|`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`|`constant`|
`markup`|`medialabel`|`menuchoice`| `mousebutton`|`option`|`optional`|`parameter`|`prompt`|
`property`| `replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|
`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`| `author`|`authorinitials`|
`corpauthor`|`modespec`|`othercredit`| `productname`|`productnumber`|`revhistory`|`remark`|
`subscript`| `superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`| `indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%bookcomponent.title.content;` `%div.title.content;`

`%formalobject.title.content;` `%info.class;` `%refsect.title.content;`

`%sect.title.content;` `%tbl.table.mdl;`

Description

`TitleAbbrev` holds an abbreviated version of a `Title`. One common use of `TitleAbbrev` is for the text used in running headers or footers, when the proper title is too long to be used conveniently.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Abbreviated titles are usually used only in specific contexts, such as headers and footers, and suppressed everywhere else.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain titleabbrev: `appendix`, `appendixinfo`, `article`, `articleinfo`, `bibliodiv`, `biblioentry`, `bibliography`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `book`, `bookinfo`, `calloutlist`, `chapter`, `chapterinfo`, `colophon`, `dedication`, `equation`, `example`, `figure`, `glossary`, `glossaryinfo`, `glossdiv`, `index`, `indexdiv`, `indexinfo`, `itemizedlist`, `lot`, `msgset`, `objectinfo`, `orderedlist`, `part`, `partinfo`, `partintro`, `preface`, `prefaceinfo`, `procedure`, `productionset`, `qandadiv`, `qandaset`, `refentryinfo`, `reference`, `referenceinfo`, `refsect1`, `refsect1info`, `refsect2`, `refsect2info`, `refsect3`, `refsect3info`, `refsection`, `refsectioninfo`, `refsynopsisdiv`, `refsynopsisdivinfo`, `sect1`, `sect1info`, `sect2`, `sect2info`, `sect3`, `sect3info`, `sect4`, `sect4info`, `sect5`, `sect5info`, `section`, `sectioninfo`, `segmentedlist`, `set`, `setindex`, `setindexinfo`, `setinfo`, `sidebar`, `sidebarinfo`, `simplesect`, `table`, `toc`, `variablelist`.

Children

The following elements occur in titleabbrev: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `citation`, `citerefentry`, `citetitle`, `classname`, `command`, `computeroutput`, `constant`, `corpauthor`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

See Also

`subtitle`, `title`.

Examples

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>How to Configure the Menu Subsystem
of the Graphical User Interface</title>
<titleabbrev>Configuring Menus</titleabbrev>
<para>
Actual content.
</para>
</chapter>
```

For additional examples, see also `article`, `book`, `xref`.

toc

Name

toc – A table of contents

Synopsis

Content Model

toc ::= (`beginpage`?, (`title`,`subtitle`?,`titleabbrev`?)?, `tocfront`*, (`tocpart`|`tocchap`)*, `tocback`*)

Attributes Common attributes

Name Type Default

pagenum CDATA *None*

Parameter Entities

`%nav.class;` `%partcontent.mix;`

Description

The `ToC` element defines a *table of contents* in a document.

The general structure of elements in a `ToC` is analogous to the structure of the document described. For example, a `ToC` for a `Book` might contain `TocFront` elements for the front-matter of the book, `TocChap` elements for the body of the book, and `TocBack` elements for the back matter. Inside each of these are additional elements reflecting the structure of each component.

Processing expectations

Formatted as a displayed block.

In real life, `ToC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toc: `appendix`, `article`, `book`, `chapter`, `part`, `preface`, `sect1`, `sect2`, `sect3`, `sect4`, `sect5`, `section`, `set`.

Children

The following elements occur in toc: `beginpage`, `subtitle`, `title`, `titleabbrev`, `tocback`, `tocchap`, `tocfront`, `tocpart`.

Attributes

pagenum PageNum indicates the page on which this Table of Contents appears in the printed document

Examples

```
<!DOCTYPE toc PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<toc>
<tocfront pagenum="i">Preface</tocfront>
<tocpart>
  <tocentry>Part I</tocentry>
  <tocchap>
    <tocentry pagenum="1">Getting Started with SGML/XML</tocentry>
    <toclevel1>
```

```
      <tocentry pagenum="1">HTML and SGML vs. XML</tocentry>
    </toclevel1>
    <toclevel1>
      <tocentry pagenum="3">How Does DocBook Fit In?</tocentry>
      <toclevel2>
        <tocentry pagenum="3">A Short DocBook History</tocentry>
      </toclevel2>
    </toclevel1>
    <!-- ... -->
  </tocchap>
</tocpart>
<!-- ... -->
<tocback pagenum="305">Bibliography</tocback>
</toc>
```

■ tocback

Name

tocback – An entry in a table of contents for a back matter component

Synopsis

Mixed Content Model

tocback ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry|citetitle|emphasis|firstterm|foreignphrase|glossterm|footnote|phrase|quote|trademark|wordasword|personname|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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|author|authorinitials|corpauthor|modespec|othercredit|productname|productnumber|revhistory|remark|subscript|superscript|inlinegraphic|inlinemediaobject|inlineequation|synopsis|cmdsynopsis|funcsynopsis|classsynopsis|fieldsynopsis|constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm|beginpage)*

Attributes Common attributes

Name Type Default

linkend IDREF *None*

label CDATA *None*

pagenum CDATA *None*

Description

The ToCback element is a chapter-level ToC element for back matter (Bibliographys, Indexs, and so on).

Processing expectations

Formatted as a displayed block.

In real life, ToCs are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain tocback: `toc`.

Children

The following elements occur in tocback: `abbrev`, `acronym`, `action`, `anchor`, `application`, `author`, `authorinitials`, `beginpage`, `citation`, `citerefentry`, `citetitle`, `classname`, `classsynopsis`, `cmdsynopsis`, `command`, `computeroutput`, `constant`, `constructorsynopsis`, `corpauthor`, `database`, `destructorsynopsis`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `fieldsynopsis`, `filename`, `firstterm`, `footnote`, `footnoteref`, `foreignphrase`, `funcsynopsis`, `function`, `glossterm`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `indexterm`, `inlineequation`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `methodsynopsis`, `modespec`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `othercredit`, `parameter`, `personname`, `phrase`, `productname`, `productnumber`, `prompt`, `property`, `quote`, `remark`, `replaceable`, `returnvalue`, `revhistory`, `sgmltag`, `structfield`, `structname`, `subscript`, `superscript`, `symbol`, `synopsis`, `systemitem`, `token`, `trademark`, `type`, `ulink`, `userinput`, `varname`, `wordasword`, `xref`.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

linkend Linkend points to the associated back matter element.

pagenum PageNum indicates the page on which the element of backmatter appears in some version of the printed document.

Examples

For examples, see `toc`.

■ tocchap

Name

tocchap – An entry in a table of contents for a component in the body of a document

Synopsis

Content Model

tocchap ::= (`tocentry`+,`toclevel1`*)

Attributes Common attributes

Name Type Default

label CDATA *None*

Description

The `ToCchap` element is a chapter-level `ToC` element for components in the main body of a document.

Processing expectations

Formatted as a displayed block.

In real life, `ToC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain tocchap: `appendix`, `article`, `chapter`, `preface`, `toc`, `tocpart`.

Children

The following elements occur in tocchap: `tocentry`, `toclevel1`.

Attributes

label Label specifies an identifying string for presentation purposes.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

Examples

For examples, see `toc`.

■ tocentry

Name

tocentry – A component title in a table of contents

Synopsis

Mixed Content Model

tocentry ::= (#PCDATA|`footnoteref`|`xref`|`abbrev`|`acronym`|`citation`|`citerefentry`|
`citetitle`|`emphasis`|`firstterm`|`foreignphrase`|`glossterm`|`footnote`|`phrase`|`quote`|
`trademark`|`wordasword`|`personname`|`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`|`constant`|
`markup`|`medialabel`|`menuchoice`|`mousebutton`|`option`|`optional`|`parameter`|`prompt`|
`property`|`replaceable`|`returnvalue`|`sgmltag`|`structfield`|`structname`|`symbol`|
`systemitem`|`token`|`type`|`userinput`|`varname`|`nonterminal`|`anchor`|`author`|`authorinitials`|
`corpauthor`|`modespec`|`othercredit`|`productname`|`productnumber`|`revhistory`|`remark`|
`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`inlineequation`|`synopsis`|
`cmdsynopsis`|`funcsynopsis`|`classsynopsis`|`fieldsynopsis`|`constructorsynopsis`|
`destructorsynopsis`|`methodsynopsis`|`indexterm`|`beginpage`)*

Attributes Common attributes

Name Type Default

linkend IDREF *None*

pagenum CDATA *None*

Description

A ToCentry contains the title of an entry in a ToC. In entries that allow nested structure, such as ToCchap, this additional wrapper is necessary in order to require that a title be present.

Processing expectations

Formatted as a displayed block.

In real life, ToCs are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain tocentry: tocchap, toclevel1, toclevel2, toclevel3, toclevel4, toclevel5, tocpart.

Children

The following elements occur in tocentry: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

linkend Linkend points to the associated element in the document.

pagenum PageNum indicates the page on which this ToC element appears in some version of the printed document.

Examples

For examples, see toc.

■ tocfront

Name

tocfront – An entry in a table of contents for a front matter component

Synopsis

Mixed Content Model

tocfront ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

linkend IDREF *None*

label CDATA *None*

pagenum CDATA *None*

Description

The ToCfront element is a chapter-level ToC element for front matter such as Prefaces.

Processing expectations

Formatted as a displayed block.

In real life, ToCs are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain tocfront: toc.

Children

The following elements occur in tocfront: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem, guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

label Label specifies an identifying number or string that may be used in presentation.

Generally, an explicit Label attribute is used only if the processing system is incapable of generating the label automatically. If present, the Label is normative; it will used even if the processing system is capable of automatic labelling.

linkend Linkend points to the associated front matter element.

pagenum PageNum indicates the page on which the element of frontmatter appears in som version of the printed document.

Examples

For examples, see `toc`.

■ toclevel1

Name

toclevel1 – A top-level entry within a table of contents entry for a chapter-like component

Synopsis

Content Model

toclevel1 ::= (`tocentry`+,`toclevel2`*)

Attributes Common attributes

Description

The `ToClevel1` element is a `ToC` entry for a first-level section in a component.

Processing expectations

Formatted as a displayed block.

In real life, `TOC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toclevel1: `tocchap`.

Children

The following elements occur in toclevel1: `tocentry`, `toclevel2`.

Examples

For examples, see `toc`.

■ toclevel2

Name

toclevel2 – A second-level entry within a table of contents entry for a chapter-like component

Synopsis

Content Model

toclevel2 ::= (`tocentry`+,`toclevel3`*)

Attributes Common attributes

Description

The `ToClevel2` element is a `ToC` entry for a second-level section in a component.

Processing expectations

Formatted as a displayed block.

In real life, `ToC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toclevel2: `toclevel1`.

Children

The following elements occur in toclevel2: `tocentry`, `toclevel3`.

Examples

For examples, see `toc`.

■ toclevel3

Name

toclevel3 – A third-level entry within a table of contents entry for a chapter-like component

Synopsis

Content Model

toclevel3 ::= (`tocentry`+,`toclevel4`*)

Attributes Common attributes

Description

The `ToClevel3` element is a `ToC` entry for a third-level section in a component.

Processing expectations

Formatted as a displayed block.

In real life, `ToC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toclevel3: `toclevel2`.

Children

The following elements occur in toclevel3: `tocentry`, `toclevel4`.

■ toclevel4

Name

toclevel4 – A fourth-level entry within a table of contents entry for a chapter-like component

Synopsis

Content Model

toclevel4 ::= (`tocentry`+,`toclevel5`*)

Attributes Common attributes

Description

The `ToClevel4` element is a `ToC` entry for a fourth-level section in a component.

Processing expectations

Formatted as a displayed block.

In real life, `TOC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toclevel4: `toclevel3`.

Children

The following elements occur in toclevel4: `tocentry`, `toclevel5`.

■ toclevel5

Name

toclevel5 – A fifth-level entry within a table of contents entry for a chapter-like component

Synopsis

Content Model

toclevel5 ::= (`tocentry`+)

Attributes Common attributes

Description

The `ToClevel5` element is a `ToC` entry for a fifth-level section in a component.

The `ToC` machinery in DocBook has not been extended to handle the infinitely recursive nature of `Section`s. It may never be extended.

Processing expectations

Formatted as a displayed block.

In real life, `TOC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain toclevel5: `toclevel4`.

Children

The following elements occur in toclevel5: `tocentry`.

■ tocpart

Name

tocpart – An entry in a table of contents for a part of a book

Synopsis

Content Model

tocpart ::= (`tocentry`+,`tocchap`*)

Attributes Common attributes

Description

The `ToCpart` element is a division-level `ToC` element for `Part`s and `Reference`s.

Processing expectations

Formatted as a displayed block.

In real life, `ToC`s are usually generated automatically by the presentation system and never have to be represented explicitly in the document source.

Parents

These elements contain tocpart: `toc`.

Children

The following elements occur in tocpart: `tocchap`, `tocentry`.

Examples

For examples, see `toc`.

■ token

Name

token – A unit of information

Synopsis

Mixed Content Model

token ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

A Token identifies a unit of information. Usually, "tokens" are the result of some processing pass that has performed lexical analysis and divided a data set into the smallest units of information used for subsequent processing.

Exactly what constitutes a token varies by context.

Processing expectations

Formatted inline.

Parents

These elements contain token: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in token: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

classname, interface, property, structfield, structname, symbol, type.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In parsing, line ends are turned into the <token>CRLF</token>, all other
whitespace becomes <token>WHITESP</token>.
</para>
```

In parsing, line ends are turned into the CRLF, all other whitespace becomes WHITESP.

■ trademark

Name

trademark – A trademark

Synopsis

Mixed Content Model

trademark ::= (#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|constant|markup|medialabel|menuchoice|
mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue|
sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname|
nonterminal|anchor| remark|subscript|superscript|inlinegraphic|inlinemediaobject|
emphasis)*

Attributes Common attributes

Name Type Default

class	copyright registered service trade	"trade"

Parameter Entities

`%docinfo.char.mix;` `%gen.char.class;` `%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;` `%tbl.entry.mdl;` `%title.char.mix;` `%word.char.mix;`

Description

`Trademark` identifies a legal trademark.

One of the values of the Class attribute on `Trademark` is `Copyright`. DocBook also has a `Copyright` element, but it is confined to meta-information. A copyright in running text is best represented as `<trademark class=copyright>`.

Processing expectations

Formatted inline.

In addition to `Trademark`, two of the values of the Class attribute on `ProductName` make assertions about trademarks; presumably the same markup is intended for both `Trademark` and `ProductName` when they make assertions about trademarks.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain trademark: `abbrev`, `ackno`, `acronym`, `application`, `artpagenums`, `attribution`, `authorinitials`, `bibliocoverage`, `biblioid`, `bibliomisc`, `bibliorelation`, `bibliosource`, `bridgehead`, `citation`, `citebiblioid`, `citetitle`, `city`, `collabname`, `confdates`, `confnum`, `confsponsor`, `conftitle`, `contractnum`, `contractsponsor`, `contrib`, `corpauthor`, `corpname`, `country`, `date`, `edition`, `email`, `emphasis`, `entry`, `fax`, `firstname`, `firstterm`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `holder`, `honorific`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `jobtitle`, `label`, `lineage`, `lineannotation`, `link`, `literallayout`, `lotentry`, `manvolnum`, `member`, `modespec`, `msgaud`, `olink`, `orgdiv`, `orgname`, `otheraddr`, `othername`, `pagenums`, `para`, `phone`, `phrase`, `pob`, `postcode`, `primary`, `primaryie`, `productname`, `productnumber`, `programlisting`, `pubdate`, `publishername`, `pubsnumber`, `quote`, `refentrytitle`, `refmiscinfo`, `refpurpose`, `releaseinfo`, `remark`, `revnumber`, `revremark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `seriesvolnums`, `shortaffil`, `simpara`, `state`, `street`, `subtitle`, `surname`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`, `volumenum`, `wordasword`, `year`.

Children

The following elements occur in trademark: `action`, `anchor`, `application`, `classname`, `command`, `computeroutput`, `constant`, `database`, `email`, `emphasis`, `envar`, `errorcode`, `errorname`, `errortext`, `errortype`, `exceptionname`, `filename`, `function`, `guibutton`, `guiicon`, `guilabel`, `guimenu`, `guimenuitem`, `guisubmenu`, `hardware`, `inlinegraphic`, `inlinemediaobject`, `interface`, `interfacename`, `keycap`, `keycode`, `keycombo`, `keysym`, `link`, `literal`, `markup`, `medialabel`, `menuchoice`, `methodname`, `mousebutton`, `nonterminal`, `olink`, `ooclass`, `ooexception`, `oointerface`, `option`, `optional`, `parameter`, `prompt`, `property`, `remark`, `replaceable`, `returnvalue`, `sgmltag`, `structfield`,

structname, subscript, superscript, symbol, systemitem, token, type, ulink, userinput, varname.

Attributes

class Class indicates the type of Trademark. The default is Trade.

See Also

copyright, legalnotice, productname.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The name <trademark class='registered'>WebSite</trademark> is a
registered trademark of O'Reilly &amp;amp; Associates, Inc.
</para>
```

The name WebSite™ is a registered trademark of O'Reilly & Associates, Inc. For additional examples, see also audioobject, productname, screenco.

■ type

Name

type – The classification of a value

Synopsis

Mixed Content Model

type ::= (#PCDATA|replaceable|inlinegraphic|inlinemediaobject|indexterm| beginpage)*

Attributes Common attributes

Parameter Entities

%cptr.char.mix; %ndxterm.char.mix; %para.char.mix;

%refinline.char.mix; %refname.char.mix; %tbl.entry.mdl;

%tech.char.class; %title.char.mix;

Description

In general usage, Type identifies one member of a class of values.

In documenting computer programs, it identifies specifically a "type," as might be declared with typedef in the C programming language.

Processing expectations

Formatted inline.

Parents

These elements contain type: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, fieldsynopsi filename, foreignphrase, funcdef, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, methodparam, methodsynopsis, msgaud, olink, option, optional, para, paramdef, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in type: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

classname, interface, property, structfield, structname, symbol, token.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
The <function>geteuid</function> function returns a <type>uid_t</type> that
contains the user's <emphasis>effective</emphasis> user id.
</para>
```

The geteuid function returns a uid_t that contains the user's *effective* user id.

■ ulink

Name

ulink – A link that addresses its target by means of a URL (Uniform Resource Locator)

Synopsis

Mixed Content Model

ulink ::= (#PCDATA|footnoteref|xref|abbrev|acronym|citation|citerefentry| citetitle| emphasis|firstterm|foreignphrase|glossterm|footnote| phrase|quote|trademark| wordasword|personname|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|constant|markup|medialabel|menuchoice| mousebutton|option|optional|parameter|prompt|property| replaceable|returnvalue| sgmltag|structfield|structname|symbol| systemitem|token|type|userinput|varname| nonterminal|anchor| author|authorinitials|corpauthor|modespec|othercredit| productname|productnumber|revhistory|remark|subscript| superscript|inlinegraphic| inlinemediaobject|inlineequation| synopsis|cmdsynopsis|funcsynopsis|classsynopsis| fieldsynopsis| constructorsynopsis|destructorsynopsis|methodsynopsis|indexterm| beginpage)*

Attributes Common attributes

Name Type Default

url CDATA *Required*

type CDATA *None*

Parameter Entities

%cptr.char.mix; %docinfo.char.mix; %indexdivcomponent.mix;
%link.char.class; %ndxterm.char.mix; %para.char.mix;
%refinline.char.mix; %tbl.entry.mdl; %title.char.mix;
%word.char.mix;

Description

The ULink element forms the equivalent of an HTML anchor (<A HREF="...">) for cross reference by a Uniform Resource Locator (URL).

Processing expectations

Formatted inline. When rendered online, it is natural to make the content of the ULink element an active link. When rendered in print media, the URL might be ignored, printed after the text of the link, or printed as a footnote.

When the content of the ULink element is empty, i.e., for either of the following cases: <ulink url="..."/> and <ulink url="..."></ulink>, the content of the url attribute should be rendered as the text of the link.

Linking elements must not be nested within other linking elements (including themselves). Because DocBook is harmonizing towards XML, this restriction cannot easily be enforced by the DTD. The processing of nested linking elements is undefined.

Future Changes

The InterfaceDefinition element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain ulink: abbrev, ackno, acronym, action, application, artpagenums, attribution, authorinitials, bibliocoverage, biblioid, bibliomisc, bibliorelation, bibliosource, bridgehead, citation, citebiblioid, citetitle, city, classsynopsisinfo, collabname, command, computeroutput, confdates, confnum, confsponsor, conftitle, contractnum, contractsponsor, contrib, corpauthor, corpname, country, database, date, edition, email, emphasis, entry, fax, figure, filename, firstname, firstterm, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, holder, honorific, indexdiv, informalfigure, interfacename, invpartnumber, isbn, issn, issuenum, jobtitle, keycap, label, lineage, lineannotation, link, literal, literallayout, lotentry, manvolnum, member, modespec, msgaud, olink, option, optional, orgdiv, orgname, otheraddr, othername, pagenums, para, parameter, phone, phrase, pob, postcode, primary, primaryie, productname, productnumber, programlisting, property, pubdate, publishername, pubsnumber, quote, refentry, refentrytitle, refmiscinfo, refnamediv, refpurpose, releaseinfo, remark, replaceable, revnumber, revremark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, seriesvolnums, shortaffil, simpara, state, street, subscript, subtitle, superscript, surname, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput, volumenum, wordasword, year.

Children

The following elements occur in ulink: abbrev, acronym, action, anchor, application, author, authorinitials, beginpage, citation, citerefentry, citetitle, classname, classsynopsis, cmdsynopsis, command, computeroutput, constant, constructorsynopsis, corpauthor, database, destructorsynopsis, email, emphasis, envar, errorcode, errorname, errortext, errortype, exceptionname, fieldsynopsis, filename, firstterm, footnote, footnoteref, foreignphrase, funcsynopsis, function, glossterm, guibutton, guiicon, guilabel, guimenu, guimenuitem,

guisubmenu, hardware, indexterm, inlineequation, inlinegraphic, inlinemediaobject, interface, interfacename, keycap, keycode, keycombo, keysym, link, literal, markup, medialabel, menuchoice, methodname, methodsynopsis, modespec, mousebutton, nonterminal, olink, ooclass, ooexception, oointerface, option, optional, othercredit, parameter, personname, phrase, productname, productnumber, prompt, property, quote, remark, replaceable, returnvalue, revhistory, sgmltag, structfield, structname, subscript, superscript, symbol, synopsis, systemitem, token, trademark, type, ulink, userinput, varname, wordasword, xref.

Attributes

type Type is available for application-specific customization of the linking behavior.

url URL specifies the Uniform Resource Locator that is the target of the ULink.

See Also

anchor, link, olink, xref.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
For more information, see the O'Reilly catalog entry for
<ulink url="http://www.ora.com/catalog/tex/"><citetitle>Making TeX
Work</citetitle></ulink>.
</para>
```

For more information, see the O'Reilly catalog entry for *Making TeX Work* http://www.ora.com/catalog/tex/. For additional examples, see also olink, productnumber, systemitem.

■ userinput

Name

userinput – Data entered by the user

Synopsis

Mixed Content Model

userinput ::= (#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|constant|markup|medialabel|menuchoice|mousebutton|option|optional|parameter|prompt|property|replaceable|returnvalue|sgmltag|structfield|structname|symbol|systemitem|token|type|userinput|varname|nonterminal|anchor|remark|subscript|superscript|inlinegraphic|inlinemediaobject|indexterm|beginpage|co)*

Attributes Common attributes

Name Type Default

moreinfo	none refentry	"none"

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`
`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`
`%tech.char.class;` `%title.char.mix;`

Description

The `UserInput` element identifies words or phrases that the user is expected to provide as input to a computer program.

Note that `UserInput` is not a verbatim environment, but an inline.

Processing expectations

Formatted inline. The MoreInfo attribute can help generate a link or query to retrieve additional information. Often presented in a fixed width font.

Future Changes

The `InterfaceDefinition` element will be discarded in DocBook V4.0. It will no longer be available in the content model of this element.

Parents

These elements contain userinput: `action`, `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `classsynopsisinfo`, `command`, `computeroutput`, `database`, `emphasis`, `entry`, `filename`, `foreignphrase`, `funcparams`, `funcsynopsisinfo`, `function`, `glosssee`, `glossseealso`, `glossterm`, `hardware`, `interfacename`, `keycap`, `lineannotation`, `link`, `literal`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `option`, `optional`, `para`, `parameter`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `property`, `quote`, `refdescriptor`, `refentrytitle`, `refname`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `systemitem`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `trademark`, `ulink`, `userinput`.

Children

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

Attributes

moreinfo If MoreInfo is set to `RefEntry`, it implies that a `RefEntry` exists which further describes the `UserInput`.

See Also

`computeroutput`, `constant`, `envar`, `filename`, `lineannotation`, `literal`, `literallayout`, `markup`, `option`, `optional`, `parameter`, `programlisting`, `prompt`, `replaceable`, `screen`, `screenshot`, `sgmltag`, `synopsis`, `systemitem`, `varname`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
At the system prompt, enter <userinput>xyzzy</userinput> to gain
supervisor access to the system.
</para>
```

At the system prompt, enter **`xyzzy`** to gain supervisor access to the system.

■ varargs

Name

varargs – An empty element in a function synopsis indicating a variable number of arguments

Synopsis

Content Model

varargs ::= EMPTY

Attributes Common attributes

Description

`VarArgs` indicates that a function takes a variable number of arguments.

Processing expectations

The `VarArgs` element produces generated text that indicates that the function takes a variable number of arguments. The exact generated text may vary. One common result is "`(...)`".

Parents

These elements contain varargs: `funcprototype`.

See Also

`funcdef`, `funcparams`, `funcprototype`, `funcsynopsisinfo`, `function`, `paramdef`, `parameter`, `returnvalue`, `void`.

Examples

```
<!DOCTYPE funcsynopsis PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<funcsynopsis>
<funcsynopsisinfo>
#include &lt;varargs.h&gt;
</funcsynopsisinfo>
<funcprototype>
  <funcdef>int <function>max</function></funcdef>
  <varargs/>
</funcprototype>
</funcsynopsis>


#include <varargs.h>
```

int **max** (...); For additional examples, see also `funcsynopsis`.

variablelist

Name

variablelist – A list in which each entry is composed of a set of one or more terms and an associated description

Synopsis

Content Model

variablelist ::= (blockinfo?, (title,titleabbrev?)?, (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|anchor|bridgehead|remark|highlights|abstract|authorblurb|epigraph|indexterm|beginpage)*, varlistentry+)

Attributes Common attributes

Name Type Default

termlength CDATA *None*

Parameter Entities

%admon.mix; %bookcomponent.content; %component.mix;

%divcomponent.mix; %example.mix; %footnote.mix;

%glossdef.mix; %highlights.mix; %indexdivcomponent.mix;

%legalnotice.mix; %list.class; %para.mix;

%qandaset.mix; %refcomponent.mix; %revdescription.mix;

%sidebar.mix; %tabentry.mix; %tbl.entry.mdl;

%textobject.mix;

Description

A VariableList is a list consisting of Terms and their definitions or descriptions.

Processing expectations

Formatted as a displayed block.

There are many ways to deal with the problems presented in formatting a variable list with long Terms. DocBook does not mandate any particular presentation. The TermLength attribute may influence the presentation of Terms.

Future Changes

Introductory material may appear before the first list item.

Parents

These elements contain variablelist: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, caution, chapter, colophon, constraintdef, dedication, entry, example, footnote, glossary, glossdef, glossdiv, highlights, important, index, indexdiv, informalexample, legalnotice, listitem, msgexplan, msgtext, note, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, tip, warning.

Children

The following elements occur in variablelist: abstract, address, anchor, authorblurb, beginpage, blockinfo, blockquote, bridgehead, caution, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, epigraph, fieldsynopsis, formalpara, funcsynopsis, graphic, graphicco, highlights, important, indexterm, informalequation, informalexample, informalfigure, informaltable, literallayout, mediaobject, mediaobjectco, methodsynopsis, note, para, programlisting, programlistingco, remark, screen, screenco, screenshot, simpara, synopsis, tip, title, titleabbrev, varlistentry, warning.

Attributes

termlength TermLength indicates a length beyond which the presentation engine may consider the Term(s) too long and select an alternate presentation of the Term(s) and/or, the associated ListItem.

See Also

calloutlist, itemizedlist, listitem, orderedlist, segmentedlist, simplelist.

Examples

```
<!DOCTYPE variablelist PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<variablelist><title>Font Filename Extensions</title>
<varlistentry><term><filename>TTF</filename></term>
<listitem>
<para>
TrueType fonts.
</para>
</listitem>
</varlistentry>
<varlistentry><term><filename>PFA</filename></term>
              <term><filename>PFB</filename></term>
<listitem>
<para>
PostScript fonts.  <filename>PFA</filename> files are common on
<acronym>UNIX</acronym> systems, <filename>PFB</filename> files
are more common on Windows systems.
</para>
</listitem>
</varlistentry>
</variablelist>
```

FONT FILENAME EXTENSIONS

TTF TrueType fonts.

PFA, PFB PostScript fonts. PFA files are common on UNIX systems, PFB files are more common on Windows systems.

■ varlistentry

Name

varlistentry – A wrapper for a set of terms and the associated description in a variable list

Synopsis

Content Model

varlistentry ::= (`term`+,`listitem`)

Attributes Common attributes

Description

A `VarListEntry` is an entry in a `VariableList`. Each `VarListEntry` contains one or more `Term`s and their description or definition.

Processing expectations

Formatted as a displayed block.

`Term`s are usually formatted to make them stand out with respect to the text that follows. The best presentation depends on several factors, including the number and length of the terms. See `VariableList`.

Parents

These elements contain varlistentry: `variablelist`.

Children

The following elements occur in varlistentry: `listitem`, `term`.

Examples

For examples, see `variablelist`.

■ varname

Name

varname – The name of a variable

Synopsis

Mixed Content Model

varname ::= (#PCDATA|`replaceable`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%cptr.char.mix;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%refname.char.mix;` `%tbl.entry.mdl;`

`%tech.char.class;` `%title.char.mix;`

Description

A `VarName` identifies a variable name in a programming or expression language. Variables most often get their values from `Literal`s, `Replaceable` values, `Constant`s, or `Symbol`s.

Processing expectations

Formatted inline.

Parents

These elements contain varname: action, application, attribution, bibliomisc, bridgehead, citation, citetitle, classsynopsisinfo, command, computeroutput, database, emphasis, entry, fieldsynopsis, filename, foreignphrase, funcparams, funcsynopsisinfo, function, glosssee, glossseealso, glossterm, hardware, interfacename, keycap, lineannotation, link, literal, literallayout, lotentry, member, msgaud, olink, option, optional, para, parameter, phrase, primary, primaryie, productname, programlisting, property, quote, refdescriptor, refentrytitle, refname, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, systemitem, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, trademark, ulink, userinput.

Children

The following elements occur in varname: beginpage, indexterm, inlinegraphic, inlinemediaobject, replaceable.

See Also

command, computeroutput, constant, literal, markup, option, optional, parameter, prompt, replaceable, sgmltag, userinput.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
In Perl, <varname>@ARGV</varname> contains the command line parameters
used when the script was run.
</para>
```

In Perl, @ARGV contains the command line parameters used when the script was run. For additional examples, see also literal.

■ videodata

Name

videodata – Pointer to external video data

Synopsis

Content Model

videodata ::= EMPTY

Attributes Common attributes

Name Type Default

srccredit CDATA *None*

valign bottom | middle | top *None*

width CDATA *None*

contentwidth CDATA *None*

BMP
CGM-BINARY
CGM-CHAR
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
IGES
format JPEG
JPG
linespecific
PCX
PIC
PNG
PS
SGML
SVG
TBL
TEX
TIFF
WMF
WPG
None

entityref ENTITY *None*

fileref CDATA *None*

scalefit CDATA *None*

depth CDATA *None*

scale CDATA *None*

contentdepth CDATA *None*

center
align left
right
None

Description

This element points to an external entity containing video data.

Processing expectations

May be formatted inline or as a displayed block, depending on context. Rendering a video is usually accomplished by reserving a rectangular area on the display and "running" the video in that frame.

There are two ways to provide content for `VideoData`: EntityRef or FileRef. It is best to use only one of these methods. However, if multiple sources are provided, EntityRef will be used in favor of FileRef.

Parents

These elements contain videodata: `videoobject`.

Attributes

align Align specifies the horizontal alignment of the image data on the page or within the element that frames it.

depth Depth specifies the desired depth (vertical distance, at least in horizontal writing systems) of the video data.

entityref EntityRef identifies the general entity which points to the content of the video data.

fileref FileRef specifies the name of the file which contains the content of the video data.

format Format identifies the format of the video data. The Format must be a defined notation.

scale Scale specifies integer representing a percentage scaling factor (retaining the relative dimensions of the original video frame). If unspecified, the value 100 (100%) is assumed.

scalefit If ScaleFit has the value 1 (true), then the video frame is to be scaled (uniformly) to the specified width or depth. The default value of 0 (false) indicates that the image will not be scaled to fit (although it may still be scaled by the Scale attribute).

srccredit SrcCredit contains details about the source of the video data.

width Width indicates the width of the graphic.

■ videoobject

Name

videoobject – A wrapper for video data and its associated meta-information

Synopsis

Content Model

videoobject ::= (`objectinfo`?,`videodata`)

Attributes Common attributes

Parameter Entities

`%mediaobject.mix;`

Description

A `VideoObject` is a wrapper containing `VideoData` and its associated meta-information.

Processing expectations

May be formatted inline or as a displayed block, depending on context. It might not be rendered at all, depending on its placement within a `MediaObject` or `InlineMediaObject` and the constraints on the publishing system. For a more detailed description of the semantics involved, see `MediaObject`.

Parents

These elements contain videoobject: `inlinemediaobject`, `mediaobject`.

Children

The following elements occur in videoobject: `objectinfo`, `videodata`.

See Also

`alt`, `audioobject`, `caption`, `graphic`, `imageobject`, `inlinegraphic`, `inlinemediaobject`, `mediaobject`, `textobject`.

Examples

```
<!DOCTYPE mediaobject PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<mediaobject>
<videoobject>
<videodata fileref='movie.avi'/>
</videoobject>
<imageobject>
<imagedata fileref='movie-frame.gif'/>
</imageobject>
<textobject>
<para>This video illustrates the proper way to assemble an
inverting time distortion device.
</para>
<warning>
<para>
It is imperative that the primary and secondary temporal
couplings not be mounted in the wrong order. Temporal
catastrophe is the likely result. The future you destroy
may be your own.
</para>
</warning>
</textobject>
</mediaobject>
```

■ void

Name

void – An empty element in a function synopsis indicating that the function in question takes no arguments

Synopsis

Content Model

void ::= EMPTY

Attributes Common attributes

Description

The `Void` element indicates explicitly that a `Function` has no arguments.

Processing expectations

The `Void` element produces generated text that indicates the function has no arguments (or returns nothing). The exact generated text may vary. One common result is `void`.

Parents

These elements contain void: `constructorsynopsis`, `destructorsynopsis`, `funcprototype`, `methodsynopsis`.

See Also

`funcdef`, `funcparams`, `funcprototype`, `funcsynopsisinfo`, `function`, `paramdef`, `parameter`, `returnvalue`, `varargs`.

Examples

For examples, see `funcdef`, `funcsynopsis`.

■ volumenum

Name

volumenum – The volume number of a document in a set (as of books in a set or articles in a journal)

Synopsis

Mixed Content Model

volumenum ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Parameter Entities

`%bibliocomponent.mix;` `%info.class;`

Description

`VolumeNum` identifies the volume number of a `Book` in a `Set`, or a periodical. It is a wrapper for bibliographic information.

Processing expectations

Formatted inline. Sometimes suppressed.

Parents

These elements contain volumenum: `appendixinfo`, `articleinfo`, `biblioentry`, `bibliographyinfo`, `bibliomixed`, `bibliomset`, `biblioset`, `blockinfo`, `bookinfo`, `chapterinfo`, `glossaryinfo`, `indexinfo`, `objectinfo`, `partinfo`, `prefaceinfo`, `refentryinfo`, `referenceinfo`, `refsect1info`, `refsect2info`, `refsect3info`, `refsectioninfo`, `refsynopsisdivinfo`, `sect1info`, `sect2info`, `sect3info`, `sect4info`, `sect5info`, `sectioninfo`, `setindexinfo`, `setinfo`, `sidebarinfo`.

Children

The following elements occur in volumenum: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`biblioid`, `invpartnumber`, `isbn`, `issn`, `issuenum`, `productnumber`, `pubsnumber`, `seriesvolnums`.

Examples

For examples, see `article`, `bibliography`, `bibliomset`.

■ warning

Name

warning – An admonition set off from the text

Synopsis

Content Model

warning ::= (title?, (calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist| simplelist|variablelist|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|sidebar|anchor|bridgehead|remark|indexterm| beginpage)+)

Attributes Common attributes

Parameter Entities

%admon.class; %bookcomponent.content; %component.mix;

%divcomponent.mix; %highlights.mix; %legalnotice.mix;

%listpreamble.mix; %para.mix; %qandaset.mix;

%refcomponent.mix; %revdescription.mix; %sidebar.mix;

%tabentry.mix; %tbl.entry.mdl; %textobject.mix;

Description

A Warning is an admonition, usually set off from the main text.

In some types of documentation, the semantics of admonitions are clearly defined (Caution might imply the possibility of harm to equipment whereas Warning might imply harm to a person), but DocBook makes no such assertions.

Processing expectations

Formatted as a displayed block. Often outputs the generated text “Warning” or some other visible indication of the type of admonition, especially if a Title is not present. Sometimes outputs a graphical icon or other symbol as well.

Parents

These elements contain warning: answer, appendix, article, bibliodiv, bibliography, blockquote, callout, caption, chapter, colophon, dedication, entry, glossary, glossdiv, highlights, index, itemizedlist, legalnotice, listitem, msgexplan, msgtext, orderedlist, para, partintro, preface, procedure, qandadiv, qandaset, question, refsect1, refsect2, refsect3, refsection, refsynopsisdiv, revdescription, sect1, sect2, sect3, sect4, sect5, section, setindex, sidebar, simplesect, step, textobject, variablelist.

Children

The following elements occur in warning: address, anchor, beginpage, blockquote, bridgehead, calloutlist, classsynopsis, cmdsynopsis, constructorsynopsis, destructorsynopsis, equation, example, fieldsynopsis, figure, formalpara, funcsynopsis, glosslist, graphic, graphicco, indexterm, informalequation, informalexample, informalfigure, informaltable, itemizedlist, literallayout, mediaobject, mediaobjectco, methodsynopsis, orderedlist, para, procedure, programlisting, programlistingco, remark, screen, screenco, screenshot, segmentedlist, sidebar, simpara, simplelist, synopsis, table, title, variablelist.

See Also

caution, important, note, tip.

Examples

```
<!DOCTYPE warning PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<warning>
<para>
Striking your thumb with a hammer may cause severe pain and discomfort.
</para>
</warning>
```

WARNING

Striking your thumb with a hammer may cause severe pain and discomfort.

For additional examples, see also `videoobject`.

■ wordasword

Name

wordasword – A word meant specifically as a word and not representing anything else

Synopsis

Mixed Content Model

wordasword ::= (#PCDATA|`acronym`|`emphasis`|`trademark`|`link`|`olink`|`ulink`|`anchor`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`|`beginpage`)*

Attributes Common attributes

Parameter Entities

`%gen.char.class;` `%ndxterm.char.mix;` `%para.char.mix;`

`%refinline.char.mix;` `%tbl.entry.mdl;` `%title.char.mix;`

Description

A lot of technical documentation contains words that have overloaded meanings. Sometimes it is useful to be able to use a word without invoking its technical meaning. The `WordAsWord` element identifies a word or phrase that might otherwise be interpreted in some specific way, and asserts that it should be interpreted simply as a word.

It is unlikely that the presentation of this element will be able to help readers understand the variation in meaning; good writing will have to achieve that goal. The real value of `WordAsWord` lies in the fact that full-text searching and indexing tools can use it to avoid false-positives.

Processing expectations

Formatted inline.

Parents

These elements contain wordasword: `application`, `attribution`, `bibliomisc`, `bridgehead`, `citation`, `citetitle`, `emphasis`, `entry`, `foreignphrase`, `glosssee`, `glossseealso`, `glossterm`, `lineannotation`, `link`, `literallayout`, `lotentry`, `member`, `msgaud`, `olink`, `para`, `phrase`, `primary`, `primaryie`, `productname`, `programlisting`, `quote`, `refentrytitle`, `refpurpose`, `remark`, `screen`, `screeninfo`, `secondary`, `secondaryie`, `see`, `seealso`, `seealsoie`, `seeie`, `seg`, `segtitle`, `simpara`, `subtitle`, `synopsis`, `term`, `tertiary`, `tertiaryie`, `title`, `titleabbrev`, `tocback`, `tocentry`, `tocfront`, `ulink`.

Children

The following elements occur in wordasword: `acronym`, `anchor`, `beginpage`, `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `subscript`, `superscript`, `trademark`, `ulink`.

See Also

`abbrev`, `acronym`, `emphasis`, `foreignphrase`, `phrase`, `quote`.

Examples

```
<!DOCTYPE para PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<para>
A <wordasword>Term</wordasword> in Algebra has a very different meaning
than a <sgmltag>Term</sgmltag> in DocBook.
</para>
```

A *Term* in Algebra has a very different meaning than a `Term` in DocBook.

■ xref

Name

xref – A cross reference to another part of the document

Synopsis

Content Model

xref ::= EMPTY

Attributes Common attributes

Name Type Default

linkend IDREF *Required*

endterm IDREF *None*

Parameter Entities

`%ndxterm.char.mix;` `%para.char.mix;` `%refinline.char.mix;`

`%tbl.entry.mdl;` `%title.char.mix;` `%xref.char.class;`

Description

The `XRef` element forms a cross-reference from the location of the `XRef` to the element to which it points. Unlike `Link` and the other cross-referencing elements, `XRef` is empty. The processing system has to generate appropriate cross-reference text for the reader.

Processing expectations

Under ordinary circumstances, the `xref` points to the some element with its linkend attribute and the processing system generates appropriate cross reference text. There are three ways for the author to influence the generated text:

1. If the endterm attribute is specified on `xref`, the content of the element pointed to by endterm will be used as the text of the cross-reference.
2. Otherwise, if the object *pointed to* has a specified XRefLabel, the content of that attribute will be used as the cross-reference text.

3. Finally, the author may specify a keyword (or other information) in the xrefstyle attribute. Unlike endterm and xreflabel which have rigid semantics, the content of the xrefstyle attribute is simply additional information for the processing system. What effect it has, if any, is dependent on the processing system.

Parents

These elements contain xref: application, attribution, bibliomisc, bridgehead, citation, citetitle, emphasis, entry, foreignphrase, glosssee, glossseealso, glossterm, lineannotation, link, literallayout, lotentry, member, msgaud, olink, para, phrase, primary, primaryie, productname, programlisting, quote, refentrytitle, refpurpose, remark, screen, screeninfo, secondary, secondaryie, see, seealso, seealsoie, seeie, seg, segtitle, simpara, subtitle, synopsis, term, tertiary, tertiaryie, title, titleabbrev, tocback, tocentry, tocfront, ulink.

Attributes

endterm Endterm points to the element whose content is to be used as the text of the link.

linkend Linkend points to the target of the cross reference.

See Also

anchor, link, olink, ulink.

Examples

Consider the following example:

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
          "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book><title>An Example Book</title>
<chapter id="ch01"><title>XRef Samples</title>
<para>
This paragraph demonstrates several features of
<sgmltag>XRef</sgmltag>.
</para>
<itemizedlist>
<listitem><para>A straight link generates the
cross-reference text: <xref linkend="ch02"/>.
</para></listitem>
<listitem><para>A link to an element with an
<sgmltag class="attribute">XRefLabel</sgmltag>:
<xref linkend="ch03"/>.
</para></listitem>
<listitem><para>A link with an
<sgmltag class="attribute">EndTerm</sgmltag>:
<xref linkend="ch04" endterm="ch04short"/>.
</para></listitem>
</itemizedlist>
</chapter>

<chapter id="ch02">
  <title>The Second Chapter</title>
  <para>Some content here</para>
</chapter>

<chapter id="ch03" xreflabel="Chapter the Third">
```

```
  <title>The Third Chapter</title>
  <para>Some content here</para>
</chapter>

<chapter id="ch04">
  <title>The Fourth Chapter</title>
  <titleabbrev id="ch04short">Chapter 4</titleabbrev>
  <para>Some content here</para>
</chapter>
</book>
```

One reasonable rendering for the content of the first chapter of this book is the following:
This paragraph demonstrates several features of `XRef`.

- A straight link generates the cross-reference text: Chapter 2, “The Second Chapter”
- A link to an element with an XRefLabel: Chapter the Third.
- A link with an EndTerm: Chapter 4.

Of course, in an online system, these references would also be links to the appropriate chapters.
For additional examples, see also `part`.

■ year

Name

year – The year of publication of a document

Synopsis

Mixed Content Model

year ::= (#PCDATA|`link`|`olink`|`ulink`|`emphasis`|`trademark`|`replaceable`|`remark`|`subscript`|`superscript`|`inlinegraphic`|`inlinemediaobject`|`indexterm`)*

Attributes Common attributes

Description

`Year` identifies a year. In DocBook V3.0, this is only used in `Copyright`, to identify the year or years in which copyright is asserted.

Processing expectations

Formatted inline.

Parents

These elements contain year: `copyright`.

Children

The following elements occur in year: `emphasis`, `indexterm`, `inlinegraphic`, `inlinemediaobject`, `link`, `olink`, `remark`, `replaceable`, `subscript`, `superscript`, `trademark`, `ulink`.

Examples

For examples, see `bibliography`, `biblioset`, `bookinfo`, `copyright`.

Chapter 7

DocBook Parameter Entity Reference

■ %*.attlist; Parameter Entities

Name

%*.attlist; Parameter Entities – Control individual attribute list declarations

Synopsis

Used to control marked sections around the declarations of individual attribute lists.

Description

The %*.attlist; parameter entities provide marked sections around individual attribute list declarations. You can selectively include or remove attribute list declarations from DocBook by changing these parameter entities.

The following sections identify all of the attlist module parameter entities in DocBook.

%abbrev.attlist;

Parameter Entity

INCLUDE

%abstract.attlist;

Parameter Entity

INCLUDE

%accel.attlist;

Parameter Entity

INCLUDE

%ackno.attlist;

Parameter Entity

INCLUDE

%acronym.attlist;

Parameter Entity

INCLUDE

%action.attlist;

Parameter Entity

INCLUDE

%address.attlist;

Parameter Entity

INCLUDE

%affiliation.attlist;

Parameter Entity

INCLUDE

%alt.attlist;

Parameter Entity

INCLUDE

%anchor.attlist;

Parameter Entity

INCLUDE

%answer.attlist;

Parameter Entity

INCLUDE

%appendix.attlist;

Parameter Entity

INCLUDE

%appendixinfo.attlist;

Parameter Entity

INCLUDE

%application.attlist;

Parameter Entity

INCLUDE

%area.attlist;

Parameter Entity

INCLUDE

%areaset.attlist;

Parameter Entity

INCLUDE

%areaspec.attlist;

Parameter Entity

INCLUDE

%arg.attlist;

Parameter Entity

INCLUDE

%article.attlist;

Parameter Entity

INCLUDE

%articleinfo.attlist;

Parameter Entity

INCLUDE

%artpagenums.attlist;

Parameter Entity

INCLUDE

%attribution.attlist;

Parameter Entity

INCLUDE

%audiodata.attlist;

Parameter Entity

INCLUDE

%audioobject.attlist;

Parameter Entity

INCLUDE

%author.attlist;

Parameter Entity

INCLUDE

%authorblurb.attlist;

Parameter Entity

INCLUDE

%authorgroup.attlist;

Parameter Entity

INCLUDE

%authorinitials.attlist;

Parameter Entity

INCLUDE

%beginpage.attlist;

Parameter Entity

INCLUDE

%bibliocoverage.attlist;

Parameter Entity

INCLUDE

%bibliodiv.attlist;

Parameter Entity

INCLUDE

%biblioentry.attlist;

Parameter Entity

INCLUDE

%bibliography.attlist;

Parameter Entity

INCLUDE

%bibliographyinfo.attlist;

Parameter Entity

INCLUDE

%biblioid.attlist;

Parameter Entity

INCLUDE

%bibliomisc.attlist;

Parameter Entity

INCLUDE

%bibliomixed.attlist;

Parameter Entity

INCLUDE

%bibliomset.attlist;

Parameter Entity

INCLUDE

%bibliorelation.attlist;

Parameter Entity

INCLUDE

%biblioset.attlist;

Parameter Entity

INCLUDE

%bibliosource.attlist;

Parameter Entity

INCLUDE

%blockinfo.attlist;

Parameter Entity

INCLUDE

%blockquote.attlist;

Parameter Entity

INCLUDE

%book.attlist;

Parameter Entity

INCLUDE

%bookinfo.attlist;

Parameter Entity

INCLUDE

%bridgehead.attlist;

Parameter Entity

INCLUDE

%callout.attlist;

Parameter Entity

INCLUDE

%calloutlist.attlist;

Parameter Entity

INCLUDE

%caption.attlist;

Parameter Entity

INCLUDE

%caution.attlist;

Parameter Entity

INCLUDE

%chapter.attlist;

Parameter Entity

INCLUDE

%chapterinfo.attlist;

Parameter Entity

INCLUDE

%citation.attlist;

Parameter Entity

INCLUDE

%citebiblioid.attlist;

Parameter Entity

INCLUDE

%citerefentry.attlist;

Parameter Entity

INCLUDE

%citetitle.attlist;

Parameter Entity

INCLUDE

%city.attlist;

Parameter Entity

INCLUDE

%classname.attlist;

Parameter Entity

INCLUDE

%classsynopsis.attlist;

Parameter Entity

INCLUDE

%classsynopsisinfo.attlist;

Parameter Entity

INCLUDE

%cmdsynopsis.attlist;

Parameter Entity

INCLUDE

%co.attlist;

Parameter Entity

INCLUDE

%collab.attlist;

Parameter Entity

INCLUDE

%collabname.attlist;

Parameter Entity

INCLUDE

%colophon.attlist;

Parameter Entity

INCLUDE

%command.attlist;

Parameter Entity

INCLUDE

%computeroutput.attlist;

Parameter Entity

INCLUDE

%confdates.attlist;

Parameter Entity

INCLUDE

%confgroup.attlist;

Parameter Entity

INCLUDE

%confnum.attlist;

Parameter Entity

INCLUDE

%confsponsor.attlist;

Parameter Entity

INCLUDE

%conftitle.attlist;

Parameter Entity

INCLUDE

%constant.attlist;

Parameter Entity

INCLUDE

%constructorsynopsis.attlist;

Parameter Entity

INCLUDE

%contractnum.attlist;

Parameter Entity

INCLUDE

%contractsponsor.attlist;

Parameter Entity

INCLUDE

%contrib.attlist;

Parameter Entity

INCLUDE

%copyright.attlist;

Parameter Entity

INCLUDE

%coref.attlist;

Parameter Entity

INCLUDE

%corpauthor.attlist;

Parameter Entity

INCLUDE

%corpname.attlist;

Parameter Entity

INCLUDE

%country.attlist;

Parameter Entity

INCLUDE

%database.attlist;

Parameter Entity

INCLUDE

%date.attlist;

Parameter Entity

INCLUDE

%dedication.attlist;

Parameter Entity

INCLUDE

%destructorsynopsis.attlist;

Parameter Entity

INCLUDE

%edition.attlist;

Parameter Entity

INCLUDE

%editor.attlist;

Parameter Entity

INCLUDE

%email.attlist;

Parameter Entity

INCLUDE

%emphasis.attlist;

Parameter Entity

INCLUDE

%envar.attlist;

Parameter Entity

INCLUDE

%epigraph.attlist;

Parameter Entity

INCLUDE

%equation.attlist;

Parameter Entity

INCLUDE

%errorcode.attlist;

Parameter Entity

INCLUDE

%errorname.attlist;

Parameter Entity

INCLUDE

%errortext.attlist;

Parameter Entity

INCLUDE

%errortype.attlist;

Parameter Entity

INCLUDE

%example.attlist;

Parameter Entity

INCLUDE

%exceptionname.attlist;

Parameter Entity

INCLUDE

%fax.attlist;

Parameter Entity

INCLUDE

%fieldsynopsis.attlist;

Parameter Entity

INCLUDE

%figure.attlist;

Parameter Entity

INCLUDE

%filename.attlist;

Parameter Entity

INCLUDE

%firstname.attlist;

Parameter Entity

INCLUDE

%firstterm.attlist;

Parameter Entity

INCLUDE

%footnote.attlist;

Parameter Entity

INCLUDE

%footnoteref.attlist;

Parameter Entity

INCLUDE

%foreignphrase.attlist;

Parameter Entity

INCLUDE

%formalpara.attlist;

Parameter Entity

INCLUDE

%funcdef.attlist;

Parameter Entity

INCLUDE

%funcparams.attlist;

Parameter Entity

INCLUDE

%funcprototype.attlist;

Parameter Entity

INCLUDE

%funcsynopsis.attlist;

Parameter Entity

INCLUDE

%funcsynopsisinfo.attlist;

Parameter Entity

INCLUDE

%function.attlist;

Parameter Entity

INCLUDE

%glossary.attlist;

Parameter Entity

INCLUDE

%glossaryinfo.attlist;

Parameter Entity

INCLUDE

%glossdef.attlist;

Parameter Entity

INCLUDE

%glossdiv.attlist;

Parameter Entity

INCLUDE

%glossentry.attlist;

Parameter Entity

INCLUDE

%glosslist.attlist;

Parameter Entity

INCLUDE

%glosssee.attlist;

Parameter Entity

INCLUDE

%glossseealso.attlist;

Parameter Entity

INCLUDE

%glossterm.attlist;

Parameter Entity

INCLUDE

%graphic.attlist;

Parameter Entity

INCLUDE

%graphicco.attlist;

Parameter Entity

INCLUDE

%group.attlist;

Parameter Entity

INCLUDE

%guibutton.attlist;

Parameter Entity

INCLUDE

%guiicon.attlist;

Parameter Entity

INCLUDE

%guilabel.attlist;

Parameter Entity

INCLUDE

%guimenu.attlist;

Parameter Entity

INCLUDE

%guimenuitem.attlist;

Parameter Entity

INCLUDE

%guisubmenu.attlist;

Parameter Entity

INCLUDE

%hardware.attlist;

Parameter Entity

INCLUDE

%highlights.attlist;

Parameter Entity

INCLUDE

%holder.attlist;

Parameter Entity

INCLUDE

%honorific.attlist;

Parameter Entity

INCLUDE

%imagedata.attlist;

Parameter Entity

INCLUDE

%imageobject.attlist;

Parameter Entity

INCLUDE

%imageobjectco.attlist;

Parameter Entity

INCLUDE

%important.attlist;

Parameter Entity

INCLUDE

%index.attlist;

Parameter Entity

INCLUDE

%indexdiv.attlist;

Parameter Entity

INCLUDE

%indexentry.attlist;

Parameter Entity

INCLUDE

%indexinfo.attlist;

Parameter Entity

INCLUDE

%indexterm.attlist;

Parameter Entity

INCLUDE

%informalequation.attlist;

Parameter Entity

INCLUDE

%informalexample.attlist;

Parameter Entity

INCLUDE

%informalfigure.attlist;

Parameter Entity

INCLUDE

%informaltable.attlist;

Parameter Entity

INCLUDE

%initializer.attlist;

Parameter Entity

INCLUDE

%inlineequation.attlist;

Parameter Entity

INCLUDE

%inlinegraphic.attlist;

Parameter Entity

INCLUDE

%inlinemediaobject.attlist;

Parameter Entity

INCLUDE

%interface.attlist;

Parameter Entity

INCLUDE

%interfacename.attlist;

Parameter Entity

INCLUDE

%invpartnumber.attlist;

Parameter Entity

INCLUDE

%isbn.attlist;

Parameter Entity

INCLUDE

%issn.attlist;

Parameter Entity

INCLUDE

%issuenum.attlist;

Parameter Entity

INCLUDE

%itemizedlist.attlist;

Parameter Entity

INCLUDE

%itermset.attlist;

Parameter Entity

INCLUDE

%jobtitle.attlist;

Parameter Entity

INCLUDE

%keycap.attlist;

Parameter Entity

INCLUDE

%keycode.attlist;

Parameter Entity

INCLUDE

%keycombo.attlist;

Parameter Entity

INCLUDE

%keysym.attlist;

Parameter Entity

INCLUDE

%keyword.attlist;

Parameter Entity

INCLUDE

%keywordset.attlist;

Parameter Entity

INCLUDE

%label.attlist;

Parameter Entity

INCLUDE

%legalnotice.attlist;

Parameter Entity

INCLUDE

%lineage.attlist;

Parameter Entity

INCLUDE

%lineannotation.attlist;

Parameter Entity

INCLUDE

%link.attlist;

Parameter Entity

INCLUDE

%listitem.attlist;

Parameter Entity

INCLUDE

%literal.attlist;

Parameter Entity

INCLUDE

%literallayout.attlist;

Parameter Entity

INCLUDE

%lot.attlist;

Parameter Entity

INCLUDE

%lotentry.attlist;

Parameter Entity

INCLUDE

%manvolnum.attlist;

Parameter Entity

INCLUDE

%markup.attlist;

Parameter Entity

INCLUDE

%medialabel.attlist;

Parameter Entity

INCLUDE

%mediaobject.attlist;

Parameter Entity

INCLUDE

%mediaobjectco.attlist;

Parameter Entity

INCLUDE

%member.attlist;

Parameter Entity

INCLUDE

%menuchoice.attlist;

Parameter Entity

INCLUDE

%methodname.attlist;

Parameter Entity

INCLUDE

%methodparam.attlist;

Parameter Entity

INCLUDE

%methodsynopsis.attlist;

Parameter Entity

INCLUDE

%modespec.attlist;

Parameter Entity

INCLUDE

%modifier.attlist;

Parameter Entity

INCLUDE

%mousebutton.attlist;

Parameter Entity

INCLUDE

%msg.attlist;

Parameter Entity

INCLUDE

%msgaud.attlist;

Parameter Entity

INCLUDE

%msgentry.attlist;

Parameter Entity

INCLUDE

%msgexplan.attlist;

Parameter Entity

INCLUDE

%msginfo.attlist;

Parameter Entity

INCLUDE

%msglevel.attlist;

Parameter Entity

INCLUDE

%msgmain.attlist;

Parameter Entity

INCLUDE

%msgorig.attlist;

Parameter Entity

INCLUDE

%msgrel.attlist;

Parameter Entity

INCLUDE

%msgset.attlist;

Parameter Entity

INCLUDE

%msgsub.attlist;

Parameter Entity

INCLUDE

%msgtext.attlist;

Parameter Entity

INCLUDE

%note.attlist;

Parameter Entity

INCLUDE

%objectinfo.attlist;

Parameter Entity

INCLUDE

%olink.attlist;

Parameter Entity

INCLUDE

%ooclass.attlist;

Parameter Entity

INCLUDE

%ooexception.attlist;

Parameter Entity

INCLUDE

%oointerface.attlist;

Parameter Entity

INCLUDE

%option.attlist;

Parameter Entity

INCLUDE

%optional.attlist;

Parameter Entity

INCLUDE

%orderedlist.attlist;

Parameter Entity

INCLUDE

%orgdiv.attlist;

Parameter Entity

INCLUDE

%orgname.attlist;

Parameter Entity

INCLUDE

%otheraddr.attlist;

Parameter Entity

INCLUDE

%othercredit.attlist;

Parameter Entity

INCLUDE

%othername.attlist;

Parameter Entity

INCLUDE

%pagenums.attlist;

Parameter Entity

INCLUDE

%para.attlist;

Parameter Entity

INCLUDE

%paramdef.attlist;

Parameter Entity

INCLUDE

%parameter.attlist;

Parameter Entity

INCLUDE

%part.attlist;

Parameter Entity

INCLUDE

%partinfo.attlist;

Parameter Entity

INCLUDE

%partintro.attlist;

Parameter Entity

INCLUDE

%personblurb.attlist;

Parameter Entity

INCLUDE

%personname.attlist;

Parameter Entity

INCLUDE

%phone.attlist;

Parameter Entity

INCLUDE

%phrase.attlist;

Parameter Entity

INCLUDE

%pob.attlist;

Parameter Entity

INCLUDE

%postcode.attlist;

Parameter Entity

INCLUDE

%preface.attlist;

Parameter Entity

INCLUDE

%prefaceinfo.attlist;

Parameter Entity

INCLUDE

%primary.attlist;

Parameter Entity

INCLUDE

%primaryie.attlist;

Parameter Entity

INCLUDE

%printhistory.attlist;

Parameter Entity

INCLUDE

%procedure.attlist;

Parameter Entity

INCLUDE

%productname.attlist;

Parameter Entity

INCLUDE

%productnumber.attlist;

Parameter Entity

INCLUDE

%programlisting.attlist;

Parameter Entity

INCLUDE

%programlistingco.attlist;

Parameter Entity

INCLUDE

%prompt.attlist;

Parameter Entity

INCLUDE

%property.attlist;

Parameter Entity

INCLUDE

%pubdate.attlist;

Parameter Entity

INCLUDE

%publisher.attlist;

Parameter Entity

INCLUDE

%publishername.attlist;

Parameter Entity

INCLUDE

%pubsnumber.attlist;

Parameter Entity

INCLUDE

%qandadiv.attlist;

Parameter Entity

INCLUDE

%qandaentry.attlist;

Parameter Entity

INCLUDE

%qandaset.attlist;

Parameter Entity

INCLUDE

%question.attlist;

Parameter Entity

INCLUDE

%quote.attlist;

Parameter Entity

INCLUDE

%refclass.attlist;

Parameter Entity

INCLUDE

%refdescriptor.attlist;

Parameter Entity

INCLUDE

%refentry.attlist;

Parameter Entity

INCLUDE

%refentryinfo.attlist;

Parameter Entity

INCLUDE

%refentrytitle.attlist;

Parameter Entity

INCLUDE

%reference.attlist;

Parameter Entity

INCLUDE

%referenceinfo.attlist;

Parameter Entity

INCLUDE

%refmeta.attlist;

Parameter Entity

INCLUDE

%refmiscinfo.attlist;

Parameter Entity

INCLUDE

%refname.attlist;

Parameter Entity

INCLUDE

%refnamediv.attlist;

Parameter Entity

INCLUDE

%refpurpose.attlist;

Parameter Entity

INCLUDE

%refsect1.attlist;

Parameter Entity

INCLUDE

%refsect1info.attlist;

Parameter Entity

INCLUDE

%refsect2.attlist;

Parameter Entity

INCLUDE

%refsect2info.attlist;

Parameter Entity

INCLUDE

%refsect3.attlist;

Parameter Entity

INCLUDE

%refsect3info.attlist;

Parameter Entity

INCLUDE

%refsection.attlist;

Parameter Entity

INCLUDE

%refsectioninfo.attlist;

Parameter Entity

INCLUDE

%refsynopsisdiv.attlist;

Parameter Entity

INCLUDE

%refsynopsisdivinfo.attlist;

Parameter Entity

INCLUDE

%releaseinfo.attlist;

Parameter Entity

INCLUDE

%remark.attlist;

Parameter Entity

INCLUDE

%replaceable.attlist;

Parameter Entity

INCLUDE

%returnvalue.attlist;

Parameter Entity

INCLUDE

%revdescription.attlist;

Parameter Entity

INCLUDE

%revhistory.attlist;

Parameter Entity

INCLUDE

%revision.attlist;

Parameter Entity

INCLUDE

%revnumber.attlist;

Parameter Entity

INCLUDE

%revremark.attlist;

Parameter Entity

INCLUDE

%sbr.attlist;

Parameter Entity

INCLUDE

%screen.attlist;

Parameter Entity

INCLUDE

%screenco.attlist;

Parameter Entity

INCLUDE

%screeninfo.attlist;

Parameter Entity

INCLUDE

%screenshot.attlist;

Parameter Entity

INCLUDE

%secondary.attlist;

Parameter Entity

INCLUDE

%secondaryie.attlist;

Parameter Entity

INCLUDE

%sect1.attlist;

Parameter Entity

INCLUDE

%sect1info.attlist;

Parameter Entity

INCLUDE

%sect2.attlist;

Parameter Entity

INCLUDE

%sect2info.attlist;

Parameter Entity

INCLUDE

%sect3.attlist;

Parameter Entity

INCLUDE

%sect3info.attlist;

Parameter Entity

INCLUDE

%sect4.attlist;

Parameter Entity

INCLUDE

%sect4info.attlist;

Parameter Entity

INCLUDE

%sect5.attlist;

Parameter Entity

INCLUDE

%sect5info.attlist;

Parameter Entity

INCLUDE

%section.attlist;

Parameter Entity

INCLUDE

%sectioninfo.attlist;

Parameter Entity

INCLUDE

%see.attlist;

Parameter Entity

INCLUDE

%seealso.attlist;

Parameter Entity

INCLUDE

%seealsoie.attlist;

Parameter Entity

INCLUDE

%seeie.attlist;

Parameter Entity

INCLUDE

%seg.attlist;

Parameter Entity

INCLUDE

%seglistitem.attlist;

Parameter Entity

INCLUDE

%segmentedlist.attlist;

Parameter Entity

INCLUDE

%segtitle.attlist;

Parameter Entity

INCLUDE

%seriesvolnums.attlist;

Parameter Entity

INCLUDE

%set.attlist;

Parameter Entity

INCLUDE

%setindex.attlist;

Parameter Entity

INCLUDE

%setindexinfo.attlist;

Parameter Entity

INCLUDE

%setinfo.attlist;

Parameter Entity

INCLUDE

%sgmltag.attlist;

Parameter Entity

INCLUDE

%shortaffil.attlist;

Parameter Entity

INCLUDE

%shortcut.attlist;

Parameter Entity

INCLUDE

%sidebar.attlist;

Parameter Entity

INCLUDE

%sidebarinfo.attlist;

Parameter Entity

INCLUDE

%simpara.attlist;

Parameter Entity

INCLUDE

%simplelist.attlist;

Parameter Entity

INCLUDE

%simplemsgentry.attlist;

Parameter Entity

INCLUDE

%simplesect.attlist;

Parameter Entity

INCLUDE

%state.attlist;

Parameter Entity

INCLUDE

%step.attlist;

Parameter Entity

INCLUDE

%street.attlist;

Parameter Entity

INCLUDE

%structfield.attlist;

Parameter Entity

INCLUDE

%structname.attlist;

Parameter Entity

INCLUDE

%subject.attlist;

Parameter Entity

INCLUDE

%subjectset.attlist;

Parameter Entity

INCLUDE

%subjectterm.attlist;

Parameter Entity

INCLUDE

%subscript.attlist;

Parameter Entity

INCLUDE

%substeps.attlist;

Parameter Entity

INCLUDE

%subtitle.attlist;

Parameter Entity

INCLUDE

%superscript.attlist;

Parameter Entity

INCLUDE

%surname.attlist;

Parameter Entity

INCLUDE

%symbol.attlist;

Parameter Entity

INCLUDE

%synopfragment.attlist;

Parameter Entity

INCLUDE

%synopfragmentref.attlist;

Parameter Entity

INCLUDE

%synopsis.attlist;

Parameter Entity

INCLUDE

%systemitem.attlist;

Parameter Entity

INCLUDE

%term.attlist;

Parameter Entity

INCLUDE

%tertiary.attlist;

Parameter Entity

INCLUDE

%tertiaryie.attlist;

Parameter Entity

INCLUDE

%textdata.attlist;

Parameter Entity

INCLUDE

%textobject.attlist;

Parameter Entity

INCLUDE

%tip.attlist;

Parameter Entity

INCLUDE

%title.attlist;

Parameter Entity

INCLUDE

%titleabbrev.attlist;

Parameter Entity

INCLUDE

%toc.attlist;

Parameter Entity

INCLUDE

%tocback.attlist;

Parameter Entity

INCLUDE

%tocchap.attlist;

Parameter Entity

INCLUDE

%tocentry.attlist;

Parameter Entity

INCLUDE

%tocfront.attlist;

Parameter Entity

INCLUDE

%toclevel1.attlist;

Parameter Entity

INCLUDE

%toclevel2.attlist;

Parameter Entity

INCLUDE

%toclevel3.attlist;

Parameter Entity

INCLUDE

%toclevel4.attlist;

Parameter Entity

INCLUDE

%toclevel5.attlist;

Parameter Entity

INCLUDE

%tocpart.attlist;

Parameter Entity

INCLUDE

%token.attlist;

Parameter Entity

INCLUDE

%trademark.attlist;

Parameter Entity

INCLUDE

%type.attlist;

Parameter Entity

INCLUDE

%ulink.attlist;

Parameter Entity

INCLUDE

%userinput.attlist;

Parameter Entity

INCLUDE

%varargs.attlist;

Parameter Entity

INCLUDE

%variablelist.attlist;

Parameter Entity

INCLUDE

%varlistentry.attlist;

Parameter Entity

INCLUDE

%varname.attlist;

Parameter Entity

INCLUDE

%videodata.attlist;

Parameter Entity

INCLUDE

%videoobject.attlist;

Parameter Entity

INCLUDE

%void.attlist;

Parameter Entity

INCLUDE

%volumenum.attlist;

Parameter Entity

INCLUDE

%warning.attlist;

Parameter Entity

INCLUDE

%wordasword.attlist;

Parameter Entity

INCLUDE

%xref.attlist;

Parameter Entity

INCLUDE

%year.attlist;

Parameter Entity

INCLUDE

■ %*.attrib; Parameter Entities

Name

%*.attrib; Parameter Entities – Define attributes on selected elements

Synopsis

These entities parameterize the attribute declarations on selected elements.

Description

Each of the following sections describes an individual parameter entity.

%bibliorelation.type.attrib;

Parameter Entity

Name Type Default

type *Enumerated:* isversionof | hasversion | isreplacedby | replaces | isrequiredby | requires | ispartof | haspart | isreferencedby | references | isformatof | hasformat | othertype

None

othertype CDATA *None*

%bibliorelation.type.attrib; appears in:

`bibliorelation`

%common.table.attribs;

Parameter Entity

label CDATA #IMPLIED id ID #IMPLIED lang CDATA #IMPLIED remap CDATA #IMPLIED xreflabel CDATA #IMPLIED revisionflag (changed |added |deleted |off) #IMPLIED arch CDATA #IMPLIED condition CDATA #IMPLIED conformance NMTOKENS #IMPLIED os CDATA #IMPLIED revision CDATA #IMPLIED security CDATA #IMPLIED userlevel CDATA #IMPLIED vendor CDATA #IMPLIED role CDATA #IMPLIED

%common.table.attribs; appears in:

`informaltable`

%condition.attrib;

Parameter Entity

Name Type Default

condition CDATA *None*

%graphics.attrib;

Parameter Entity

Name Type Default

entityref ENTITY *None*

fileref CDATA *None*

BMP
CGM-CHAR
CGM-BINARY
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
JPG
format *Enumerated:* JPEG
IGES
PCX
PIC
PNG
PS
SGML
TBL
TEX
TIFF
WMF
WPG
SVG
linespecific

None

srccredit CDATA *None*

width CDATA *None*

contentwidth CDATA *None*

depth CDATA *None*

contentdepth CDATA *None*

left
align *Enumerated:* right
center

None

top
valign *Enumerated:* middle
bottom

None

scale CDATA *None*

scalefit CDATA *None*

%graphics.attrib; appears in:

`graphic` `inlinegraphic`

%keyaction.attrib;

Parameter Entity

Name Type Default

click
double-click
action *Enumerated:* press
seq
simul
other

None

otheraction CDATA *None*

%keyaction.attrib; appears in:

`keycombo` `shortcut`

%label.attrib;

Parameter Entity

Name Type Default

label CDATA *None*

%label.attrib; appears in:

`appendix` `area` `areaset`
`book` `chapter` `cmdsynopsis`
`co` `coref` `equation`
`example` `figure` `footnote`
`footnoteref` `funcsynopsis` `informalfigure`
`lot` `part` `partintro`
`reference` `sect1` `sect2`
`sect3` `sect4` `sect5`
`section` `synopsis` `tocback`
`tocchap` `tocfront`

%linespecific.attrib;

Parameter Entity

Name Type Default

format *Enumerated notation:* linespecific
'linespecific'

linenumbering *Enumerated:* numbered
unnumbered
None

%linespecific.attrib; appears in:

`address` `classsynopsisinfo` `funcsynopsisinfo`
`literallayout` `programlisting` `screen`
`synopsis`

%linkend.attrib;

Parameter Entity

Name Type Default

linkend IDREF *None*

%linkend.attrib; appears in:

`firstterm` `glossterm` `lotentry`
`seeie` `tocback` `tocentry`
`tocfront`

%linkendreq.attrib;

Parameter Entity

Name Type Default

linkend IDREF *Required*

%linkendreq.attrib; appears in:

`coref` `footnoteref` `link`

`synopfragmentref` `xref`

%linkends.attrib;

Parameter Entity

Name Type Default

linkends IDREFS *None*

%linkends.attrib; appears in:

`area` `co` `primaryie`

`secondaryie` `seealsoie` `tertiaryie`

%mark.attrib;

Parameter Entity

Name Type Default

mark CDATA *None*

%mark.attrib; appears in:

`itemizedlist`

%moreinfo.attrib;

Parameter Entity

Name Type Default

moreinfo *Enumerated:* refentry
none

'none'

%moreinfo.attrib; appears in:

`action` `application` `command`

`computeroutput` `database` `errorcode`

`filename` `function` `guibutton`

`guiicon` `guilabel` `guimenu`

`guimenuitem` `guisubmenu` `hardware`

`interface` `keycap` `keycombo`

`literal` `menuchoice` `mousebutton`

`parameter` `prompt` `property`

`shortcut` `systemitem` `userinput`

%objectdata.attrib;

Parameter Entity

Name Type Default

entityref ENTITY *None*

fileref CDATA *None*

format *Enumerated:* BMP
CGM-CHAR
CGM-BINARY
CGM-CLEAR
DITROFF
DVI
EPS
EQN
FAX
GIF
GIF87a
GIF89a
JPG
JPEG
IGES
PCX
PIC
PNG
PS
SGML
TBL
TEX
TIFF
WMF
WPG
SVG
linespecific

None

srccredit CDATA *None*

%objectdata.attrib; appears in:

`audiodata` `imagedata` `textdata` `videodata`

%pagenum.attrib;

Parameter Entity

Name Type Default

pagenum CDATA *None*

%pagenum.attrib; appears in:

`anchor` `beginpage` `indexterm` `lotentry` `title` `toc` `tocback` `tocentry` `tocfront`

%required-IDREF-to-constraintdef-ID;

Parameter Entity

linkend IDREF #REQUIRED

%required-IDREF-to-constraintdef-ID; appears in:

`constraint`

%required-IDREF-to-production-ID;

Parameter Entity

linkend IDREF #REQUIRED

%required-IDREF-to-production-ID; appears in:

`productionrecap`

%required-XLink-to-production-ID;

Parameter Entity

def CDATA #REQUIRED

%required-XLink-to-production-ID; appears in:

`nonterminal`

%role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%role.attrib; appears in:

`constraint` `constraintdef` `lhs`
`nonterminal` `production` `productionrecap`
`productionset` `rhs`

%security.attrib;

Parameter Entity

Name Type Default

security CDATA *None*

%status.attrib;

Parameter Entity

Name Type Default

status CDATA *None*

%status.attrib; appears in:

`appendix` `article` `bibliodiv`
`bibliography` `book` `chapter`
`colophon` `dedication` `glossary`
`glossdiv` `part` `preface`
`refentry` `reference` `refsect1`
`refsect2` `refsect3` `refsection`
`sect1` `sect2` `sect3`
`sect4` `sect5` `section`
`set`

%width.attrib;

Parameter Entity

Name Type Default

width CDATA *None*

%width.attrib; appears in:

`example` `informalexample` `literallayout` `programlisting` `screen`

■ %*.attval; Parameter Entities

Name

`%*.attval;` Parameter Entities – Define attribute values

Synopsis

These parameter entities define attribute values used on selected attributes.

Description

The following parameter entities define attribute values in DocBook:

%yesorno.attvals;

Parameter Entity

CDATA

%yesorno.attvals; appears in:

`figure` `imagedata` `informalfigure` `informaltable` `videodata`

■ CALS Table Model Parameter Entities

Name

CALS Table Model Parameter Entities – Parameter entities which control the CALS Table Model

Synopsis

The CALS Table Model is incorporated into DocBook by reference to the standard CALS Table Model DTD. The CALS Table Model DTD, like DocBook, can be customized by defining a number of parameter entities before including it. The parameter entities in the following sections are the table-model related parameter entities defined by DocBook for use in the CALS Table Model.

Description

The following parameter entities are related to the CALS Table Model:

%bodyatt;

Parameter Entity

Name Type Default

label CDATA *None*

%bodyatt; appears in:

`table`

%paracon;

Parameter Entity

#PCDATA

%secur;

Parameter Entity

Name Type Default

id ID *None*

lang CDATA *None*

remap CDATA *None*

xreflabel CDATA *None*

revisionflag *Enumerated:* changed added deleted off

None

arch CDATA *None*

condition CDATA *None*

conformance NMTOKENS *None*

os CDATA *None*

revision CDATA *None*

security CDATA *None*

userlevel CDATA *None*

vendor CDATA *None*

role CDATA *None*

%secur; appears in:

`entry` `entrytbl` `row`
`table` `tbody` `tfoot`
`tgroup` `thead`

%tabentry.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |caution|important|note| tip|warning |literallayout|programlisting|programlistingco|screen |screenco|screenshot |formalpara|para| simpara |graphic|mediaobject

%tablemodel;

External Entity

Public identifier: -//OASIS//DTD DocBook CALS Table Model V4.2CR3//EN **System identifier**: calstblx.dtd

%tbl.entry.mdl;

Parameter Entity

#PCDATA |footnoteref|xref |abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase| glossterm|footnote|phrase |quote|trademark|wordasword|personname |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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt| property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type|userinput| varname |nonterminal |anchor |author|authorinitials|corpauthor|modespec|othercredit |productname| productnumber|revhistory |remark|subscript|superscript |inlinegraphic|inlinemediaobject|inlineequation |synopsis|cmdsynopsis|funcsynopsis |classsynopsis|fieldsynopsis |constructorsynopsis |destructorsynopsis |methodsynopsis |indexterm |beginpage | calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist | simplelist|variablelist |caution|important|note|tip|warning |literallayout|programlisting|programlistingco| screen |screenco|screenshot |formalpara|para|simpara |graphic|mediaobject

%tbl.entry.mdl; appears in:

`entry`

%tbl.entrytbl.mdl;

Parameter Entity

colspec*,spanspec*,thead?,tbody

%tbl.entrytbl.mdl; appears in:

`entrytbl`

%tbl.hdft.mdl;

Parameter Entity

colspec*,row

%tbl.hdft.mdl; appears in:

`tfoot` `thead`

%tbl.row.mdl;

Parameter Entity

(entry|entrytbl

%tbl.row.mdl; appears in:

`row`

%tbl.table-main.mdl;

Parameter Entity

(tgroup+|graphic

%tbl.table-titles.mdl;

Parameter Entity

title

%tbl.table.att;

Parameter Entity

Name Type Default

tabstyle CDATA *None*

tocentry CDATA *None*

shortentry CDATA *None*

orient *Enumerated:* port
land

None

pgwide CDATA *None*

%tbl.table.att; appears in:

`informaltable` `table`

%tbl.table.mdl;

Parameter Entity

(blockinfo?, (title, titleabbrev?), (indexterm)*, textobject*, (graphic+|mediaobject+|tgroup

%tbl.table.mdl; appears in:

`table`

%tbl.table.name;

Parameter Entity

(table|chart

%tbl.tgroup.att;

Parameter Entity

Name Type Default

tgroupstyle CDATA *None*

%tbl.tgroup.att; appears in:

`entrytbl` `tgroup`

%tbl.tgroup.mdl;

Parameter Entity

colspec*,spanspec*,thead?,tfoot?,tbody

%tbl.tgroup.mdl; appears in:

`tgroup`

%titles;

Parameter Entity

title

%yesorno;

Parameter Entity

CDATA

%yesorno; appears in:

`colspec` `entry` `entrytbl`
`row` `spanspec` `table`
`tgroup`

■ %*.class; Parameter Entities

Name

%*.class; Parameter Entities – Parameter entities which define the DocBook classes

Synopsis

Classes group elements of a similar type, for example all the lists are in the %list.class;

If you want to add a new kind of something (a new kind of list or a new kind of verbatim environment, for example), you generally want to add the name of the new element to the appropriate class.

Description

The following sections identify all of the class parameter entities in DocBook.

%admon.class;

Parameter Entity

caution|important|note|tip|warning

%appendix.class;

Parameter Entity

appendix

%appendix.class; appears in:

`article` `book`

%article.class;

Parameter Entity

article

%article.class; appears in:

`book`

%base.char.class;

Parameter Entity

anchor

%base.char.class; appears in:

`replaceable` `subscript` `superscript` `trademark`

%biblio.class.attrib;

Parameter Entity

Name Type Default

		uri doi isbn
class	*Enumerated:*	issn libraryofcongress pubnumber other
		None
otherclass	CDATA	*None*

%biblio.class.attrib; appears in:

`biblioid` `bibliorelation` `bibliosource` `citebiblioid`

%book.class;

Parameter Entity

book

%book.class; appears in:

`set`

%chapter.class;

Parameter Entity

chapter

%chapter.class; appears in:

book

%compound.class;

Parameter Entity

msgset|procedure|sidebar|qandaset |productionset|constraintdef

%descobj.class;

Parameter Entity

abstract|authorblurb|epigraph

%docinfo.char.class;

Parameter Entity

author|authorinitials|corpauthor|modespec|othercredit |productname|productnumber|revhistory

%formal.class;

Parameter Entity

equation|example|figure|table

%gen.char.class;

Parameter Entity

abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase|glossterm|footnote|phrase |quote|trademark|wordasword|personname

%genobj.class;

Parameter Entity

anchor|bridgehead|remark|highlights

%index.class;

Parameter Entity

index|setindex

%index.class; appears in:

book

%info.class;

Parameter Entity

graphic | mediaobject | legalnotice | modespec | subjectset | keywordset | itermset | abbrev|abstract| address|artpagenums|author |authorgroup|authorinitials|bibliomisc|biblioset |collab|confgroup|contractnum| contractsponsor |copyright|corpauthor|corpname|date|edition |editor|invpartnumber|isbn|issn|issuenum| orgname |biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage |othercredit|pagenums|printhistory| productname |productnumber|pubdate|publisher|publishername |pubsnumber|releaseinfo|revhistory| seriesvolnums |subtitle|title|titleabbrev|volumenum|citetitle |personname|honorific|firstname|surname| lineage|othername|affiliation |authorblurb|contrib |indexterm

%info.class; appears in:

`appendixinfo` `articleinfo` `bibliographyinfo`

`blockinfo` `bookinfo` `chapterinfo`

`glossaryinfo` `indexinfo` `objectinfo`

`partinfo` `prefaceinfo` `refentryinfo`

`referenceinfo` `refsect1info` `refsect2info`

`refsect3info` `refsectioninfo` `refsynopsisdivinfo`

`sect1info` `sect2info` `sect3info`

`sect4info` `sect5info` `sectioninfo`

`setindexinfo` `setinfo` `sidebarinfo`

%informal.class;

Parameter Entity

address|blockquote |graphic|graphicco|mediaobject|mediaobjectco |informalequation |informalexample |informalfigure |informaltable

%inlineobj.char.class;

Parameter Entity

inlinegraphic|inlinemediaobject|inlineequation

%linespecific.class;

Parameter Entity

literallayout|programlisting|programlistingco|screen |screenco|screenshot

%link.char.class;

Parameter Entity

link|olink|ulink

%link.char.class; appears in:

`figure` `informalfigure` `refentry`

`refnamediv` `replaceable` `subscript`

`superscript` `trademark`

%list.class;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist

%method.synop.class;

Parameter Entity

constructorsynopsis |destructorsynopsis |methodsynopsis

%method.synop.class; appears in:

`classsynopsis`

%nav.class;

Parameter Entity

toc|lot|index|glossary|bibliography

%nav.class; appears in:

`appendix` `article` `chapter`
`preface` `sect1` `sect2`
`sect3` `sect4` `sect5`
`section`

%ndxterm.class;

Parameter Entity

indexterm

%ndxterm.class; appears in:

`formalpara` `glossentry` `refentry`
`refmeta`

%notation.class;

Parameter Entity

BMP| CGM-CHAR | CGM-BINARY | CGM-CLEAR | DITROFF | DVI | EPS | EQN | FAX | GIF | GIF87a | GIF89a | JPG | JPEG | IGES | PCX | PIC | PNG | PS | SGML | TBL | TEX | TIFF | WMF | WPG | SVG | linespecific

%notation.class; appears in:

`modespec`

%other.char.class;

Parameter Entity

remark|subscript|superscript

%other.char.class; appears in:

`replaceable` `subscript` `superscript`
`trademark`

%para.class;

Parameter Entity

formalpara|para|simpara

%para.class; appears in:

`abstract` `authorblurb` `epigraph`
`personblurb` `printhistory`

%refentry.class;

Parameter Entity

refentry

%refentry.class; appears in:

`reference` `sect1` `sect2`
`sect3` `sect4` `sect5`
`section`

%section.class;

Parameter Entity

section

%section.class; appears in:

`section`

%synop.class;

Parameter Entity

synopsis|cmdsynopsis|funcsynopsis |classsynopsis|fieldsynopsis |constructorsynopsis |destructorsynopsis |methodsynopsis

%tech.char.class;

Parameter Entity

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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt|property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type| userinput|varname |nonterminal

%tech.char.class; appears in:

`trademark`

%xref.char.class;

Parameter Entity

footnoteref|xref

■ Content Model Parameter Entities

Name

Content Model Parameter Entities – Control groups of element definitions

Synopsis

The content model parameter entities provide marked sections around groups of related elements. You can selectively include or remove these elements from DocBook by changing these parameter entities.

Description

The following sections identify all of the content model parameter entities in DocBook.

FIXME: This is probably a DTD typo. This should be the sidebar.content.module parameter entity, surely?

%sidebar.content.model;

Parameter Entity

INCLUDE

■ Content Module Parameter Entities

Name

Content Module Parameter Entities – Control groups of element definitions

Synopsis

The content module parameter entities provide marked sections around groups of related elements. You can selectively include or remove these elements from DocBook by changing these parameter entities.

Description

The following sections identify all of the content module parameter entities in DocBook.

%address.content.module;

Parameter Entity

INCLUDE

%affiliation.content.module;

Parameter Entity

INCLUDE

%areaspec.content.module;

Parameter Entity

INCLUDE

%authorgroup.content.module;

Parameter Entity

INCLUDE

%bibliography.content.module;

Parameter Entity

INCLUDE

%book.content.module;

Parameter Entity

INCLUDE

%calloutlist.content.module;

Parameter Entity

INCLUDE

%classsynopsis.content.module;

Parameter Entity

INCLUDE

%cmdsynopsis.content.module;

Parameter Entity

INCLUDE

%collab.content.module;

Parameter Entity

INCLUDE

%confgroup.content.module;

Parameter Entity

INCLUDE

%copyright.content.module;

Parameter Entity

INCLUDE

%docinfo.content.module;

Parameter Entity

INCLUDE

%funcsynopsis.content.module;

Parameter Entity

INCLUDE

%glossary.content.module;

Parameter Entity

INCLUDE

%glossentry.content.module;

Parameter Entity

INCLUDE

%index.content.module;

Parameter Entity

INCLUDE

%indexterm.content.module;

Parameter Entity

INCLUDE

%keywordset.content.module;

Parameter Entity

INCLUDE

%lot.content.module;

Parameter Entity

INCLUDE

%mediaobject.content.module;

Parameter Entity

INCLUDE

%menuchoice.content.module;

Parameter Entity

INCLUDE

%msgset.content.module;

Parameter Entity

INCLUDE

%procedure.content.module;

Parameter Entity

INCLUDE

%publisher.content.module;

Parameter Entity

INCLUDE

%qandaset.content.module;

Parameter Entity

INCLUDE

%refentry.content.module;

Parameter Entity

INCLUDE

%revhistory.content.module;

Parameter Entity

INCLUDE

%screenshot.content.module;

Parameter Entity

INCLUDE

%section.content.module;

Parameter Entity

INCLUDE

%segmentedlist.content.module;

Parameter Entity

INCLUDE

%set.content.module;

Parameter Entity

INCLUDE

%simplelist.content.module;

Parameter Entity

INCLUDE

%subjectset.content.module;

Parameter Entity

INCLUDE

%toc.content.module;

Parameter Entity

INCLUDE

%variablelist.content.module;

Parameter Entity

INCLUDE

■ Common Attribute Parameter Entities

Name

Common Attribute Parameter Entities – Parameter entities which define the common attributes

Synopsis

These parameter entities define the attributes that are considered “common”. Common attributes occur on every element. They are provided by either the `%common.attrib;` parameter entity or the `%idreq.common.attrib;` parameter entity. Those parameter entities are defined in terms of the others, listed here.

Description

The following parameter entities define the common attributes in DocBook:

%arch.attrib;

Parameter Entity

Name Type Default

arch CDATA *None*

%common.attrib;

Parameter Entity

Name Type Default

id ID *None*

lang CDATA *None*

remap CDATA *None*

xreflabel CDATA *None*

revisionflag *Enumerated:* changed, added, deleted, off

None

arch CDATA *None*

condition CDATA *None*
conformance NMTOKENS *None*
os CDATA *None*
revision CDATA *None*
security CDATA *None*
userlevel CDATA *None*
vendor CDATA *None*

%common.attrib; appears in:

abbrev abstract accel
ackno acronym action
address affiliation alt
answer appendix appendixinfo
application areaspec arg
article articleinfo artpagenums
attribution audiodata audioobject
author authorblurb authorgroup
authorinitials beginpage bibliocoverage
bibliodiv biblioentry bibliography
bibliographyinfo biblioid bibliomisc
bibliomixed bibliomset bibliorelation
biblioset bibliosource blockinfo
blockquote book bookinfo
bridgehead callout calloutlist
caption caution chapter
chapterinfo citation citebiblioid
citerefentry citetitle city
classname classsynopsis classsynopsisinfo
cmdsynopsis collab collabname
colophon command computeroutput
confdates confgroup confnum
confsponsor conftitle constant
constraint constructorsynopsis contractnum
contractsponsor contrib copyright
coref corpauthor corpname
country database date
dedication destructorsynopsis edition
editor email emphasis
envar epigraph equation
errorcode errorname errortext
errortype example exceptionname
fax fieldsynopsis figure
filename firstname firstterm
footnote footnoteref foreignphrase
formalpara funcdef funcparams
funcprototype funcsynopsis funcsynopsisinfo

function glossary glossaryinfo
glossdef glossdiv glossentry
glosslist glosssee glossseealso
glossterm graphic graphicco
group guibutton guiicon
guilabel guimenu guimenuitem
guisubmenu hardware highlights
holder honorific imagedata
imageobject imageobjectco important
index indexdiv indexentry
indexinfo indexterm informalequation
informalexample informalfigure initializer
inlineequation inlinegraphic inlinemediaobject
interface interfacename invpartnumber
isbn issn issuenum
itemizedlist itermset jobtitle
keycap keycode keycombo
keysym keyword keywordset
label legalnotice lhs
lineage lineannotation link
listitem literal literallayout
lot lotentry manvolnum
markup medialabel mediaobject
mediaobjectco member menuchoice
methodname methodparam methodsynopsis
modespec modifier mousebutton
msg msgaud msgentry
msgexplan msginfo msglevel
msgmain msgorig msgrel
msgset msgsub msgtext
nonterminal note objectinfo
olink ooclass ooexception
oointerface option optional
orderedlist orgdiv orgname
otheraddr othercredit othername
pagenums para paramdef
parameter part partinfo
partintro personblurb personname
phone phrase pob
postcode preface prefaceinfo
primary primaryie printhistory
procedure productionrecap productionset
productname productnumber programlisting
programlistingco prompt property
pubdate publisher publishername

pubsnumber qandadiv qandaentry
qandaset question quote
refclass refdescriptor refentry
refentryinfo refentrytitle reference
referenceinfo refmeta refmiscinfo
refname refnamediv refpurpose
refsect1 refsect1info refsect2
refsect2info refsect3 refsect3info
refsection refsectioninfo refsynopsisdiv
refsynopsisdivinfo releaseinfo remark
replaceable returnvalue revdescription
revhistory revision revnumber
revremark rhs sbr
screen screenco screeninfo
screenshot secondary secondaryie
sect1 sect1info sect2
sect2info sect3 sect3info
sect4 sect4info sect5
sect5info section sectioninfo
see seealso seealsoie
seeie seg seglistitem
segmentedlist segtitle seriesvolnums
set setindex setindexinfo
setinfo sgmltag shortaffil
shortcut sidebar sidebarinfo
simpara simplelist simplemsgentry
simplesect state step
street structfield structname
subject subjectset subjectterm
subscript substeps subtitle
superscript surname symbol
synopfragmentref synopsis systemitem
term tertiary tertiaryie
textdata textobject tip
title titleabbrev toc
tocback tocchap tocentry
tocfront toclevel1 toclevel2
toclevel3 toclevel4 toclevel5
tocpart token trademark
type ulink userinput
varargs variablelist varlistentry
varname videodata videoobject
void volumenum warning
wordasword xref year

%conformance.attrib;

Parameter Entity
Name Type Default
conformance NMTOKENS *None*

%effectivity.attrib;

Parameter Entity
Name Type Default
arch CDATA *None*
condition CDATA *None*
conformance NMTOKENS *None*
os CDATA *None*
revision CDATA *None*
security CDATA *None*
userlevel CDATA *None*
vendor CDATA *None*
%effectivity.attrib; appears in:
`anchor`

%id.attrib;

Parameter Entity
Name Type Default
id ID *None*

%idreq.attrib;

Parameter Entity
Name Type Default
id ID *Required*
%idreq.attrib; appears in:
`anchor`

%idreq.common.attrib;

Parameter Entity
Name Type Default
id ID *Required*
lang CDATA *None*
remap CDATA *None*
xreflabel CDATA *None*
revisionflag *Enumerated:* changed, added, deleted, off
None
arch CDATA *None*
condition CDATA *None*
conformance NMTOKENS *None*

os CDATA *None*

revision CDATA *None*

security CDATA *None*

userlevel CDATA *None*

vendor CDATA *None*

%idreq.common.attrib; appears in:

`area` `areaset` `co`
`constraintdef` `production` `synopfragment`

%lang.attrib;

Parameter Entity

Name Type Default

lang CDATA *None*

%os.attrib;

Parameter Entity

Name Type Default

os CDATA *None*

%remap.attrib;

Parameter Entity

Name Type Default

remap CDATA *None*

%remap.attrib; appears in:

`anchor`

%revision.attrib;

Parameter Entity

Name Type Default

revision CDATA *None*

%revisionflag.attrib;

Parameter Entity

Name Type Default

revisionflag *Enumerated:* changed, added, deleted, off

None

%revisionflag.attrib; appears in:

`anchor`

%userlevel.attrib;

Parameter Entity

Name Type Default

userlevel CDATA *None*

%vendor.attrib;

Parameter Entity

Name Type Default

vendor CDATA *None*

%xreflabel.attrib;

Parameter Entity

Name Type Default

xreflabel CDATA *None*

%xreflabel.attrib; appears in:

`anchor`

■ DocBook Content Parameter Entities

Name

DocBook Content Parameter Entities – Specify content of selected elements

Synopsis

These parameter entities control the content of selected elements.

Description

The content model of several elements in DocBook are controlled by parameter entities.

The following sections identify all of the module parameter entities in DocBook.

%bookcomponent.content;

Parameter Entity

((calloutlist|glosslist|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 |msgset|procedure|sidebar| qandaset |productionset|constraintdef |anchor|bridgehead|remark|highlights |abstract|authorblurb|epigraph |indexterm |beginpage)+, (sect1*|(refentry)*|simplesect*|(section)*)) | (sect1+|(refentry)+|simplesect+| (section

%bookcomponent.content; appears in:

`appendix` `article` `chapter`

`partintro` `preface`

%bookcomponent.title.content;

Parameter Entity

title, subtitle?, titleabbrev

%bookcomponent.title.content; appears in:

`appendix` `bibliography` `chapter`

`glossary` `index` `lot`

`part` `preface` `reference`

`setindex` `toc`

%div.title.content;

Parameter Entity

title, subtitle?, titleabbrev

%div.title.content; appears in:

`article` `book` `partintro`
`set`

%equation.content;

Parameter Entity

(alt?, (graphic+|mediaobject

%equation.content; appears in:

`equation` `informalequation`

%formalobject.title.content;

Parameter Entity

title, titleabbrev

%formalobject.title.content; appears in:

`calloutlist` `equation` `example`
`figure` `itemizedlist` `msgset`
`orderedlist` `procedure` `productionset`
`qandadiv` `qandaset` `segmentedlist`
`sidebar` `variablelist`

%inlineequation.content;

Parameter Entity

(alt?, (graphic+|inlinemediaobject

%inlineequation.content; appears in:

`inlineequation`

%refsect.title.content;

Parameter Entity

title, subtitle?, titleabbrev

%refsect.title.content; appears in:

`refsect1` `refsect2` `refsect3`
`refsection` `refsynopsisdiv`

%sect.title.content;

Parameter Entity

title, subtitle?, titleabbrev

%sect.title.content; appears in:

`bibliodiv` `colophon` `dedication`
`glossdiv` `indexdiv` `sect1`
`sect2` `sect3` `sect4`
`sect5` `section` `simplesect`

%db*; Parameter Entities

Name

%db*; Parameter Entities – Control inclusion of DocBook modules

Synopsis

These parameter entities control the inclusion of DocBook modules.

Description

The modules listed here are included in DocBook DTD via parameter entity reference. For more information about these modules, and additional modules that can be defined, see Chapter 5, *Customizing DocBook*.

The following sections identify all of the module parameter entities in DocBook.

%dbcent.module;

Parameter Entity

INCLUDE

%dbcent;

External Entity

Public identifier: -//OASIS//ENTITIES DocBook Character Entities V4.2CR3//EN **System identifier**: dbcentx.mod

%dbgenent.module;

Parameter Entity

INCLUDE

%dbgenent;

External Entity

Public identifier: -//OASIS//ENTITIES DocBook Additional General Entities V4.2CR3//EN **System identifier**: dbgenent.mod

%dbhier.module;

Parameter Entity

INCLUDE

%dbhier;

External Entity

Public identifier: -//OASIS//ELEMENTS DocBook Document Hierarchy V4.2CR3//EN **System identifier**: dbhierx.mod

%dbhier.redecl.module;

Parameter Entity

IGNORE

%dbhier.redecl2.module;

Parameter Entity

IGNORE

%dbnotn.module;

Parameter Entity

INCLUDE

%dbnotn;

External Entity

Public identifier: -//OASIS//ENTITIES DocBook Notations V4.2CR3//EN **System identifier**: dbnotnx.mod

%dbpool.module;

Parameter Entity

INCLUDE

%dbpool;

External Entity

Public identifier: -//OASIS//ELEMENTS DocBook Information Pool V4.2CR3//EN **System identifier**: dbpoolx.mod

%dbpool.redecl.module;

Parameter Entity

IGNORE

■ %*.element; Parameter Entities

Name

%*.element; Parameter Entities – Control individual element declarations

Synopsis

Used to control marked sections around the declarations of individual elements.

Description

The %*.element; parameter entities provide marked sections around individual element declarations. You can selectively include or remove element declarations from DocBook by changing these parameter entities.

The following sections identify all of the element parameter entities in DocBook.

%abbrev.element;

Parameter Entity

INCLUDE

%abstract.element;

Parameter Entity

INCLUDE

%accel.element;

Parameter Entity

INCLUDE

%ackno.element;

Parameter Entity

INCLUDE

%acronym.element;

Parameter Entity

INCLUDE

%action.element;

Parameter Entity

INCLUDE

%address.element;

Parameter Entity

INCLUDE

%affiliation.element;

Parameter Entity

INCLUDE

%alt.element;

Parameter Entity

INCLUDE

%anchor.element;

Parameter Entity

INCLUDE

%answer.element;

Parameter Entity

INCLUDE

%appendix.element;

Parameter Entity

INCLUDE

%appendixinfo.element;

Parameter Entity

INCLUDE

%application.element;

Parameter Entity

INCLUDE

%area.element;

Parameter Entity

INCLUDE

%areaset.element;

Parameter Entity

INCLUDE

%areaspec.element;

Parameter Entity

INCLUDE

%arg.element;

Parameter Entity

INCLUDE

%article.element;

Parameter Entity

INCLUDE

%articleinfo.element;

Parameter Entity

INCLUDE

%artpagenums.element;

Parameter Entity

INCLUDE

%attribution.element;

Parameter Entity

INCLUDE

%audiodata.element;

Parameter Entity

INCLUDE

%audioobject.element;

Parameter Entity

INCLUDE

%author.element;

Parameter Entity

INCLUDE

%authorblurb.element;

Parameter Entity

INCLUDE

%authorgroup.element;

Parameter Entity

INCLUDE

%authorinitials.element;

Parameter Entity

INCLUDE

%beginpage.element;

Parameter Entity

INCLUDE

%bibliocoverage.element;

Parameter Entity

INCLUDE

%bibliodiv.element;

Parameter Entity

INCLUDE

%biblioentry.element;

Parameter Entity

INCLUDE

%bibliography.element;

Parameter Entity

INCLUDE

%bibliographyinfo.element;

Parameter Entity

INCLUDE

%biblioid.element;

Parameter Entity

INCLUDE

%bibliomisc.element;

Parameter Entity

INCLUDE

%bibliomixed.element;

Parameter Entity

INCLUDE

%bibliomset.element;

Parameter Entity

INCLUDE

%bibliorelation.element;

Parameter Entity

INCLUDE

%biblioset.element;

Parameter Entity

INCLUDE

%bibliosource.element;

Parameter Entity

INCLUDE

%blockinfo.element;

Parameter Entity

INCLUDE

%blockquote.element;

Parameter Entity

INCLUDE

%book.element;

Parameter Entity

INCLUDE

%bookinfo.element;

Parameter Entity

INCLUDE

%bridgehead.element;

Parameter Entity

INCLUDE

%callout.element;

Parameter Entity

INCLUDE

%calloutlist.element;

Parameter Entity

INCLUDE

%caption.element;

Parameter Entity

INCLUDE

%caution.element;

Parameter Entity

INCLUDE

%chapter.element;

Parameter Entity

INCLUDE

%chapterinfo.element;

Parameter Entity

INCLUDE

%citation.element;

Parameter Entity

INCLUDE

%citebiblioid.element;

Parameter Entity

INCLUDE

%citerefentry.element;

Parameter Entity

INCLUDE

%citetitle.element;

Parameter Entity

INCLUDE

%city.element;

Parameter Entity

INCLUDE

%classname.element;

Parameter Entity

INCLUDE

%classsynopsis.element;

Parameter Entity

INCLUDE

%classsynopsisinfo.element;

Parameter Entity

INCLUDE

%cmdsynopsis.element;

Parameter Entity

INCLUDE

%co.element;

Parameter Entity

INCLUDE

%collab.element;

Parameter Entity

INCLUDE

%collabname.element;

Parameter Entity

INCLUDE

%colophon.element;

Parameter Entity

INCLUDE

%command.element;

Parameter Entity

INCLUDE

%computeroutput.element;

Parameter Entity

INCLUDE

%confdates.element;

Parameter Entity

INCLUDE

%confgroup.element;

Parameter Entity

INCLUDE

%confnum.element;

Parameter Entity

INCLUDE

%confsponsor.element;

Parameter Entity

INCLUDE

%conftitle.element;

Parameter Entity

INCLUDE

%constant.element;

Parameter Entity

INCLUDE

%constructorsynopsis.element;

Parameter Entity

INCLUDE

%contractnum.element;

Parameter Entity

INCLUDE

%contractsponsor.element;

Parameter Entity

INCLUDE

%contrib.element;

Parameter Entity

INCLUDE

%copyright.element;

Parameter Entity

INCLUDE

%coref.element;

Parameter Entity

INCLUDE

%corpauthor.element;

Parameter Entity

INCLUDE

%corpname.element;

Parameter Entity

INCLUDE

%country.element;

Parameter Entity

INCLUDE

%database.element;

Parameter Entity

INCLUDE

%date.element;

Parameter Entity

INCLUDE

%dedication.element;

Parameter Entity

INCLUDE

%destructorsynopsis.element;

Parameter Entity

INCLUDE

%edition.element;

Parameter Entity

INCLUDE

%editor.element;

Parameter Entity

INCLUDE

%email.element;

Parameter Entity

INCLUDE

%emphasis.element;

Parameter Entity

INCLUDE

%envar.element;

Parameter Entity

INCLUDE

%epigraph.element;

Parameter Entity

INCLUDE

%equation.element;

Parameter Entity

INCLUDE

%errorcode.element;

Parameter Entity

INCLUDE

%errorname.element;

Parameter Entity

INCLUDE

%errortext.element;

Parameter Entity

INCLUDE

%errortype.element;

Parameter Entity

INCLUDE

%example.element;

Parameter Entity

INCLUDE

%exceptionname.element;

Parameter Entity

INCLUDE

%fax.element;

Parameter Entity

INCLUDE

%fieldsynopsis.element;

Parameter Entity

INCLUDE

%figure.element;

Parameter Entity

INCLUDE

%filename.element;

Parameter Entity

INCLUDE

%firstname.element;

Parameter Entity

INCLUDE

%firstterm.element;

Parameter Entity

INCLUDE

%footnote.element;

Parameter Entity

INCLUDE

%footnoteref.element;

Parameter Entity

INCLUDE

%foreignphrase.element;

Parameter Entity

INCLUDE

%formalpara.element;

Parameter Entity

INCLUDE

%funcdef.element;

Parameter Entity

INCLUDE

%funcparams.element;

Parameter Entity

INCLUDE

%funcprototype.element;

Parameter Entity

INCLUDE

%funcsynopsis.element;

Parameter Entity

INCLUDE

%funcsynopsisinfo.element;

Parameter Entity

INCLUDE

%function.element;

Parameter Entity

INCLUDE

%glossary.element;

Parameter Entity

INCLUDE

%glossaryinfo.element;

Parameter Entity

INCLUDE

%glossdef.element;

Parameter Entity

INCLUDE

%glossdiv.element;

Parameter Entity

INCLUDE

%glossentry.element;

Parameter Entity

INCLUDE

%glosslist.element;

Parameter Entity

INCLUDE

%glosssee.element;

Parameter Entity

INCLUDE

%glossseealso.element;

Parameter Entity

INCLUDE

%glossterm.element;

Parameter Entity

INCLUDE

%graphic.element;

Parameter Entity

INCLUDE

%graphicco.element;

Parameter Entity

INCLUDE

%group.element;

Parameter Entity

INCLUDE

%guibutton.element;

Parameter Entity

INCLUDE

%guiicon.element;

Parameter Entity

INCLUDE

%guilabel.element;

Parameter Entity

INCLUDE

%guimenu.element;

Parameter Entity

INCLUDE

%guimenuitem.element;

Parameter Entity

INCLUDE

%guisubmenu.element;

Parameter Entity

INCLUDE

%hardware.element;

Parameter Entity

INCLUDE

%highlights.element;

Parameter Entity

INCLUDE

%holder.element;

Parameter Entity

INCLUDE

%honorific.element;

Parameter Entity

INCLUDE

%imagedata.element;

Parameter Entity

INCLUDE

%imageobject.element;

Parameter Entity

INCLUDE

%imageobjectco.element;

Parameter Entity

INCLUDE

%important.element;

Parameter Entity

INCLUDE

%index.element;

Parameter Entity

INCLUDE

%indexdiv.element;

Parameter Entity

INCLUDE

%indexentry.element;

Parameter Entity

INCLUDE

%indexinfo.element;

Parameter Entity

INCLUDE

%indexterm.element;

Parameter Entity

INCLUDE

%informalequation.element;

Parameter Entity

INCLUDE

%informalexample.element;

Parameter Entity

INCLUDE

%informalfigure.element;

Parameter Entity

INCLUDE

%informaltable.element;

Parameter Entity

INCLUDE

%initializer.element;

Parameter Entity

INCLUDE

%inlineequation.element;

Parameter Entity

INCLUDE

%inlinegraphic.element;

Parameter Entity

INCLUDE

%inlinemediaobject.element;

Parameter Entity

INCLUDE

%interface.element;

Parameter Entity

INCLUDE

%interfacename.element;

Parameter Entity

INCLUDE

%invpartnumber.element;

Parameter Entity

INCLUDE

%isbn.element;

Parameter Entity

INCLUDE

%issn.element;

Parameter Entity

INCLUDE

%issuenum.element;

Parameter Entity

INCLUDE

%itemizedlist.element;

Parameter Entity

INCLUDE

%itermset.element;

Parameter Entity

INCLUDE

%jobtitle.element;

Parameter Entity

INCLUDE

%keycap.element;

Parameter Entity

INCLUDE

%keycode.element;

Parameter Entity

INCLUDE

%keycombo.element;

Parameter Entity

INCLUDE

%keysym.element;

Parameter Entity

INCLUDE

%keyword.element;

Parameter Entity

INCLUDE

%keywordset.element;

Parameter Entity

INCLUDE

%label.element;

Parameter Entity

INCLUDE

%legalnotice.element;

Parameter Entity

INCLUDE

%lineage.element;

Parameter Entity

INCLUDE

%lineannotation.element;

Parameter Entity

INCLUDE

%link.element;

Parameter Entity

INCLUDE

%listitem.element;

Parameter Entity

INCLUDE

%literal.element;

Parameter Entity

INCLUDE

%literallayout.element;

Parameter Entity

INCLUDE

%lot.element;

Parameter Entity

INCLUDE

%lotentry.element;

Parameter Entity

INCLUDE

%manvolnum.element;

Parameter Entity

INCLUDE

%markup.element;

Parameter Entity

INCLUDE

%medialabel.element;

Parameter Entity

INCLUDE

%mediaobject.element;

Parameter Entity

INCLUDE

%mediaobjectco.element;

Parameter Entity

INCLUDE

%member.element;

Parameter Entity

INCLUDE

%menuchoice.element;

Parameter Entity

INCLUDE

%methodname.element;

Parameter Entity

INCLUDE

%methodparam.element;

Parameter Entity

INCLUDE

%methodsynopsis.element;

Parameter Entity

INCLUDE

%modespec.element;

Parameter Entity

INCLUDE

%modifier.element;

Parameter Entity

INCLUDE

%mousebutton.element;

Parameter Entity

INCLUDE

%msg.element;

Parameter Entity

INCLUDE

%msgaud.element;

Parameter Entity

INCLUDE

%msgentry.element;

Parameter Entity

INCLUDE

%msgexplan.element;

Parameter Entity

INCLUDE

%msginfo.element;

Parameter Entity

INCLUDE

%msglevel.element;

Parameter Entity

INCLUDE

%msgmain.element;

Parameter Entity

INCLUDE

%msgorig.element;

Parameter Entity

INCLUDE

%msgrel.element;

Parameter Entity

INCLUDE

%msgset.element;

Parameter Entity

INCLUDE

%msgsub.element;

Parameter Entity

INCLUDE

%msgtext.element;

Parameter Entity

INCLUDE

%note.element;

Parameter Entity

INCLUDE

%objectinfo.element;

Parameter Entity

INCLUDE

%olink.element;

Parameter Entity

INCLUDE

%ooclass.element;

Parameter Entity

INCLUDE

%ooexception.element;

Parameter Entity

INCLUDE

%oointerface.element;

Parameter Entity

INCLUDE

%option.element;

Parameter Entity

INCLUDE

%optional.element;

Parameter Entity

INCLUDE

%orderedlist.element;

Parameter Entity

INCLUDE

%orgdiv.element;

Parameter Entity

INCLUDE

%orgname.element;

Parameter Entity

INCLUDE

%otheraddr.element;

Parameter Entity

INCLUDE

%othercredit.element;

Parameter Entity

INCLUDE

%othername.element;

Parameter Entity

INCLUDE

%pagenums.element;

Parameter Entity

INCLUDE

%para.element;

Parameter Entity

INCLUDE

%paramdef.element;

Parameter Entity

INCLUDE

%parameter.element;

Parameter Entity

INCLUDE

%part.element;

Parameter Entity

INCLUDE

%partinfo.element;

Parameter Entity

INCLUDE

%partintro.element;

Parameter Entity

INCLUDE

%personblurb.element;

Parameter Entity

INCLUDE

%personname.element;

Parameter Entity

INCLUDE

%phone.element;

Parameter Entity

INCLUDE

%phrase.element;

Parameter Entity

INCLUDE

%pob.element;

Parameter Entity

INCLUDE

%postcode.element;

Parameter Entity

INCLUDE

%preface.element;

Parameter Entity

INCLUDE

%prefaceinfo.element;

Parameter Entity

INCLUDE

%primary.element;

Parameter Entity

INCLUDE

%primaryie.element;

Parameter Entity

INCLUDE

%printhistory.element;

Parameter Entity

INCLUDE

%procedure.element;

Parameter Entity

INCLUDE

%productname.element;

Parameter Entity

INCLUDE

%productnumber.element;

Parameter Entity

INCLUDE

%programlisting.element;

Parameter Entity

INCLUDE

%programlistingco.element;

Parameter Entity

INCLUDE

%prompt.element;

Parameter Entity

INCLUDE

%property.element;

Parameter Entity

INCLUDE

%pubdate.element;

Parameter Entity

INCLUDE

%publisher.element;

Parameter Entity

INCLUDE

%publishername.element;

Parameter Entity

INCLUDE

%pubsnumber.element;

Parameter Entity

INCLUDE

%qandadiv.element;

Parameter Entity

INCLUDE

%qandaentry.element;

Parameter Entity

INCLUDE

%qandaset.element;

Parameter Entity

INCLUDE

%question.element;

Parameter Entity

INCLUDE

%quote.element;

Parameter Entity

INCLUDE

%refclass.element;

Parameter Entity

INCLUDE

%refdescriptor.element;

Parameter Entity

INCLUDE

%refentry.element;

Parameter Entity

INCLUDE

%refentryinfo.element;

Parameter Entity

INCLUDE

%refentrytitle.element;

Parameter Entity

INCLUDE

%reference.element;

Parameter Entity

INCLUDE

%referenceinfo.element;

Parameter Entity

INCLUDE

%refmeta.element;

Parameter Entity

INCLUDE

%refmiscinfo.element;

Parameter Entity

INCLUDE

%refname.element;

Parameter Entity

INCLUDE

%refnamediv.element;

Parameter Entity

INCLUDE

%refpurpose.element;

Parameter Entity

INCLUDE

%refsect1.element;

Parameter Entity

INCLUDE

%refsect1info.element;

Parameter Entity

INCLUDE

%refsect2.element;

Parameter Entity

INCLUDE

%refsect2info.element;

Parameter Entity

INCLUDE

%refsect3.element;

Parameter Entity

INCLUDE

%refsect3info.element;

Parameter Entity

INCLUDE

%refsection.element;

Parameter Entity

INCLUDE

%refsectioninfo.element;

Parameter Entity

INCLUDE

%refsynopsisdiv.element;

Parameter Entity

INCLUDE

%refsynopsisdivinfo.element;

Parameter Entity

INCLUDE

%releaseinfo.element;

Parameter Entity

INCLUDE

%remark.element;

Parameter Entity

INCLUDE

%replaceable.element;

Parameter Entity

INCLUDE

%returnvalue.element;

Parameter Entity

INCLUDE

%revdescription.element;

Parameter Entity

INCLUDE

%revhistory.element;

Parameter Entity

INCLUDE

%revision.element;

Parameter Entity

INCLUDE

%revnumber.element;

Parameter Entity

INCLUDE

%revremark.element;

Parameter Entity

INCLUDE

%sbr.element;

Parameter Entity

INCLUDE

%screen.element;

Parameter Entity

INCLUDE

%screenco.element;

Parameter Entity

INCLUDE

%screeninfo.element;

Parameter Entity

INCLUDE

%screenshot.element;

Parameter Entity

INCLUDE

%secondary.element;

Parameter Entity

INCLUDE

%secondaryie.element;

Parameter Entity

INCLUDE

%sect1.element;

Parameter Entity

INCLUDE

%sect1info.element;

Parameter Entity

INCLUDE

%sect2.element;

Parameter Entity

INCLUDE

%sect2info.element;

Parameter Entity

INCLUDE

%sect3.element;

Parameter Entity

INCLUDE

%sect3info.element;

Parameter Entity

INCLUDE

%sect4.element;

Parameter Entity

INCLUDE

%sect4info.element;

Parameter Entity

INCLUDE

%sect5.element;

Parameter Entity

INCLUDE

%sect5info.element;

Parameter Entity

INCLUDE

%section.element;

Parameter Entity

INCLUDE

%sectioninfo.element;

Parameter Entity

INCLUDE

%see.element;

Parameter Entity

INCLUDE

%seealso.element;

Parameter Entity

INCLUDE

%seealsoie.element;

Parameter Entity

INCLUDE

%seeie.element;

Parameter Entity

INCLUDE

%seg.element;

Parameter Entity

INCLUDE

%seglistitem.element;

Parameter Entity

INCLUDE

%segmentedlist.element;

Parameter Entity

INCLUDE

%segtitle.element;

Parameter Entity

INCLUDE

%seriesvolnums.element;

Parameter Entity

INCLUDE

%set.element;

Parameter Entity

INCLUDE

%setindex.element;

Parameter Entity

INCLUDE

%setindexinfo.element;

Parameter Entity

INCLUDE

%setinfo.element;

Parameter Entity

INCLUDE

%sgmltag.element;

Parameter Entity

INCLUDE

%shortaffil.element;

Parameter Entity

INCLUDE

%shortcut.element;

Parameter Entity

INCLUDE

%sidebar.element;

Parameter Entity

INCLUDE

%sidebarinfo.element;

Parameter Entity

INCLUDE

%simpara.element;

Parameter Entity

INCLUDE

%simplelist.element;

Parameter Entity

INCLUDE

%simplemsgentry.element;

Parameter Entity

INCLUDE

%simplesect.element;

Parameter Entity

INCLUDE

%state.element;

Parameter Entity

INCLUDE

%step.element;

Parameter Entity

INCLUDE

%street.element;

Parameter Entity

INCLUDE

%structfield.element;

Parameter Entity

INCLUDE

%structname.element;

Parameter Entity

INCLUDE

%subject.element;

Parameter Entity

INCLUDE

%subjectset.element;

Parameter Entity

INCLUDE

%subjectterm.element;

Parameter Entity

INCLUDE

%subscript.element;

Parameter Entity

INCLUDE

%substeps.element;

Parameter Entity

INCLUDE

%subtitle.element;

Parameter Entity

INCLUDE

%superscript.element;

Parameter Entity

INCLUDE

%surname.element;

Parameter Entity

INCLUDE

%symbol.element;

Parameter Entity

INCLUDE

%synopfragment.element;

Parameter Entity

INCLUDE

%synopfragmentref.element;

Parameter Entity

INCLUDE

%synopsis.element;

Parameter Entity

INCLUDE

%systemitem.element;

Parameter Entity

INCLUDE

%term.element;

Parameter Entity

INCLUDE

%tertiary.element;

Parameter Entity

INCLUDE

%tertiaryie.element;

Parameter Entity

INCLUDE

%textdata.element;

Parameter Entity

INCLUDE

%textobject.element;

Parameter Entity

INCLUDE

%tip.element;

Parameter Entity

INCLUDE

%title.element;

Parameter Entity

INCLUDE

%titleabbrev.element;

Parameter Entity

INCLUDE

%toc.element;

Parameter Entity

INCLUDE

%tocback.element;

Parameter Entity

INCLUDE

%tocchap.element;

Parameter Entity

INCLUDE

%tocentry.element;

Parameter Entity

INCLUDE

%tocfront.element;

Parameter Entity

INCLUDE

%toclevel1.element;

Parameter Entity

INCLUDE

%toclevel2.element;

Parameter Entity

INCLUDE

%toclevel3.element;

Parameter Entity

INCLUDE

%toclevel4.element;

Parameter Entity

INCLUDE

%toclevel5.element;

Parameter Entity

INCLUDE

%tocpart.element;

Parameter Entity

INCLUDE

%token.element;

Parameter Entity

INCLUDE

%trademark.element;

Parameter Entity

INCLUDE

%type.element;

Parameter Entity

INCLUDE

%ulink.element;

Parameter Entity

INCLUDE

%userinput.element;

Parameter Entity

INCLUDE

%varargs.element;

Parameter Entity

INCLUDE

%variablelist.element;

Parameter Entity

INCLUDE

%varlistentry.element;

Parameter Entity

INCLUDE

%varname.element;

Parameter Entity

INCLUDE

%videodata.element;

Parameter Entity

INCLUDE

%videoobject.element;

Parameter Entity

INCLUDE

%void.element;

Parameter Entity

INCLUDE

%volumenum.element;

Parameter Entity

INCLUDE

%warning.element;

Parameter Entity

INCLUDE

%wordasword.element;

Parameter Entity

INCLUDE

%xref.element;

Parameter Entity

INCLUDE

%year.element;

Parameter Entity

INCLUDE

■ %*.exclusion; Parameter Entities

Name

%*.exclusion; Parameter Entities – Control SGML exclusions

Synopsis

Used to control the exclusion declarations on elements with exclusions.

Description

The %*.exclusion; parameter entities parameterize element exclusions. You can selectively include or remove elements from the exclusion by changing the appropriate parameter entity.

In the XML version of DocBook V4, there are no inclusion entities because inclusions are not part of XML.

■ %*.hook; Parameter Entities

Name

%*.hook; Parameter Entities – Control access to additional modules

Synopsis

The hook parameter entities provide a mechanism for including additional modules (EBNF, MathML, HTML Forms, etc.). Each of these hooks exists so that the module can insert additional content in appropriate places without interfering with the normal user-level customization provided by the local.* entities.

Description

The following sections identify all of the hook parameter entities in DocBook.

%ebnf.block.hook;

Parameter Entity

|productionset|constraintdef

%ebnf.inline.hook;

Parameter Entity

|nonterminal

%ebnf;

External Entity

System identifier: dbebnf.dtd

%forminlines.hook;

Parameter Entity

The replacement text for this entity is empty.

%forms.hook;

Parameter Entity

The replacement text for this entity is empty.

■ %*.inclusion; Parameter Entities

Name

%*.inclusion; Parameter Entities – Control SGML inclusions

Synopsis

Used to control the inclusion declarations on elements with inclusions.

Description

The %*.inclusion; parameter entities parameterize element inclusions. You can selectively include or remove elements from the inclusion by changing the appropriate parameter entity. For example, if you remove BeginPage from the %ubiq.inclusion; parameter entity, it is no longer included everywhere in DocBook.

In the XML version of DocBook V4, there are no inclusion entities because inclusions are not part of XML.

■ %iso*; Parameter Entities

Name

%iso*; Parameter Entities – Parameter entities which control ISO Entity Sets

Synopsis

These parameter entities pull in the corresponding ISO entity set.

Description

The following sections identify all of the ISO entity sets in DocBook.

%ISOamsa;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Arrow Relations//EN//XML
System identifier: ent/iso-amsa.ent

%ISOamsb;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Binary Operators//EN//XML
System identifier: ent/iso-amsb.ent

%ISOamsc;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Delimiters//EN//XML **System identifier**: ent/iso-amsc.ent

%ISOamsn;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Negated Relations//EN//XML
System identifier: ent/iso-amsn.ent

%ISOamso;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Ordinary//EN//XML **System identifier**: ent/iso-amso.ent

%ISOamsr;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Math Symbols: Relations//EN//XML **System identifier**: ent/iso-amsr.ent

%ISObox;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Box and Line Drawing//EN//XML **System identifier**: ent/iso-box.ent

%ISOcyr1;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Russian Cyrillic//EN//XML **System identifier**: ent/iso-cyr1.ent

%ISOcyr2;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Non-Russian Cyrillic//EN//XML **System identifier**: ent/iso-cyr2.ent

%ISOdia;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Diacritical Marks//EN//XML **System identifier**: ent/iso-dia.ent

%ISOgrk1;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Greek Letters//EN//XML **System identifier**: ent/iso-grk1.ent

%ISOgrk2;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Monotoniko Greek//EN//XML **System identifier**: ent/iso-grk2.ent

%ISOgrk3;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Greek Symbols//EN//XML **System identifier**: ent/iso-grk3.ent

%ISOgrk4;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Alternative Greek Symbols//EN//XML **System identifier**: ent/iso-grk4.ent

%ISOlat1;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Latin 1//EN//XML **System identifier**: ent/iso-lat1.ent

%ISOlat2;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Added Latin 2//EN//XML **System identifier**: ent/iso-lat2.ent

%ISOnum;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN//XML **System identifier**: ent/iso-num.ent

%ISOpub;

External Entity

Public identifier: ISO 8879:1986//ENTITIES Publishing//EN//XML **System identifier**: ent/iso-pub.ent

%ISOtech;

External Entity

Public identifier: ISO 8879:1986//ENTITIES General Technical//EN//XML **System identifier**: ent/iso-tech.ent

■ %local.*.attrib; Parameter Entities

Name

`%local.*.attrib;` Parameter Entities – Allow attribute extension

Synopsis

These parameter entities offer a place where you can easily add new attributes to DocBook elements.

Description

The following sections identify all of the local attribute parameter entities in DocBook. Each of the entities is defined as empty.

%local.abbrev.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.abbrev.attrib; appears in:

`abbrev`

%local.abstract.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.abstract.attrib; appears in:

`abstract`

%local.accel.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.accel.attrib; appears in:

`accel`

%local.ackno.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.ackno.attrib; appears in:

`ackno`

%local.acronym.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.acronym.attrib; appears in:

`acronym`

%local.action.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.action.attrib; appears in:

`action`

%local.address.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.address.attrib; appears in:

`address`

%local.admon.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.admon.attrib; appears in:

`caution` `important` `note`
`tip` `warning`

%local.affiliation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.affiliation.attrib; appears in:

`affiliation`

%local.alt.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.alt.attrib; appears in:

`alt`

%local.anchor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.anchor.attrib; appears in:

`anchor`

%local.answer.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.answer.attrib; appears in:

`answer`

%local.appendix.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.appendix.attrib; appears in:

`appendix`

%local.appendixinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.appendixinfo.attrib; appears in:

`appendixinfo`

%local.application.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.application.attrib; appears in:

`application`

%local.area.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.area.attrib; appears in:

area

%local.areaset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.areaset.attrib; appears in:

areaset

%local.areaspec.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.areaspec.attrib; appears in:

areaspec

%local.arg.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.arg.attrib; appears in:

arg

%local.article.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.article.attrib; appears in:

article

%local.articleinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.articleinfo.attrib; appears in:

articleinfo

%local.artpagenums.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.artpagenums.attrib; appears in:

artpagenums

%local.attribution.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.attribution.attrib; appears in:

`attribution`

%local.audiodata.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.audiodata.attrib; appears in:

`audiodata`

%local.audioobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.audioobject.attrib; appears in:

`audioobject`

%local.author.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.author.attrib; appears in:

`author`

%local.authorblurb.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.authorblurb.attrib; appears in:

`authorblurb`

%local.authorgroup.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.authorgroup.attrib; appears in:

`authorgroup`

%local.authorinitials.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.authorinitials.attrib; appears in:

`authorinitials`

%local.beginpage.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.beginpage.attrib; appears in:

beginpage

%local.bibliocoverage.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliocoverage.attrib; appears in:

bibliocoverage

%local.bibliodiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliodiv.attrib; appears in:

bibliodiv

%local.biblioentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.biblioentry.attrib; appears in:

biblioentry

%local.bibliography.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliography.attrib; appears in:

bibliography

%local.bibliographyinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliographyinfo.attrib; appears in:

bibliographyinfo

%local.biblioid.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.biblioid.attrib; appears in:

biblioid

%local.bibliomisc.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliomisc.attrib; appears in:

bibliomisc

%local.bibliomixed.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliomixed.attrib; appears in:

`bibliomixed`

%local.bibliomset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliomset.attrib; appears in:

`bibliomset`

%local.biblиorelation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.biblиorelation.attrib; appears in:

`biblиorelation`

%local.biblиorelation.types;

Parameter Entity

The replacement text for this entity is empty.

%local.biblioset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.biblioset.attrib; appears in:

`biblioset`

%local.bibliosource.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliosource.attrib; appears in:

`bibliosource`

%local.blockinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.blockinfo.attrib; appears in:

`blockinfo`

%local.blockquote.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.blockquote.attrib; appears in:

`blockquote`

%local.book.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.book.attrib; appears in:

book

%local.bookinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bookinfo.attrib; appears in:

bookinfo

%local.bridgehead.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.bridgehead.attrib; appears in:

bridgehead

%local.callout.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.callout.attrib; appears in:

callout

%local.calloutlist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.calloutlist.attrib; appears in:

calloutlist

%local.caption.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.caption.attrib; appears in:

caption

%local.chapter.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.chapter.attrib; appears in:

chapter

%local.chapterinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.chapterinfo.attrib; appears in:

`chapterinfo`

%local.citation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.citation.attrib; appears in:

`citation`

%local.citebiblioid.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.citebiblioid.attrib; appears in:

`citebiblioid`

%local.citerefentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.citerefentry.attrib; appears in:

`citerefentry`

%local.citetitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.citetitle.attrib; appears in:

`citetitle`

%local.city.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.city.attrib; appears in:

`city`

%local.classname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.classname.attrib; appears in:

`classname`

%local.classsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.classsynopsis.attrib; appears in:

`classsynopsis`

%local.classsynopsisinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.classsynopsisinfo.attrib; appears in:

`classsynopsisinfo`

%local.cmdsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.cmdsynopsis.attrib; appears in:

`cmdsynopsis`

%local.co.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.co.attrib; appears in:

`co`

%local.collab.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.collab.attrib; appears in:

`collab`

%local.collabname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.collabname.attrib; appears in:

`collabname`

%local.colophon.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.colophon.attrib; appears in:

`colophon`

%local.command.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.command.attrib; appears in:

`command`

%local.common.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.computeroutput.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.computeroutput.attrib; appears in:

`computeroutput`

%local.confdates.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.confdates.attrib; appears in:

`confdates`

%local.confgroup.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.confgroup.attrib; appears in:

`confgroup`

%local.confnum.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.confnum.attrib; appears in:

`confnum`

%local.confsponsor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.confsponsor.attrib; appears in:

`confsponsor`

%local.conftitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.conftitle.attrib; appears in:

`conftitle`

%local.constant.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.constant.attrib; appears in:

`constant`

%local.constructorsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.constructorsynopsis.attrib; appears in:

`constructorsynopsis`

%local.contractnum.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.contractnum.attrib; appears in:

`contractnum`

%local.contractsponsor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.contractsponsor.attrib; appears in:

`contractsponsor`

%local.contrib.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.contrib.attrib; appears in:

`contrib`

%local.copyright.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.copyright.attrib; appears in:

`copyright`

%local.coref.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.coref.attrib; appears in:

`coref`

%local.corpauthor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.corpauthor.attrib; appears in:

`corpauthor`

%local.corpname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.corpname.attrib; appears in:

`corpname`

%local.country.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.country.attrib; appears in:

`country`

%local.database.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.database.attrib; appears in:

`database`

%local.date.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.date.attrib; appears in:

`date`

%local.dedication.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.dedication.attrib; appears in:

`dedication`

%local.destructorsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.destructorsynopsis.attrib; appears in:

`destructorsynopsis`

%local.edition.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.edition.attrib; appears in:

`edition`

%local.editor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.editor.attrib; appears in:

`editor`

%local.effectivity.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.email.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.email.attrib; appears in:

`email`

%local.emphasis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.emphasis.attrib; appears in:

`emphasis`

%local.envar.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.envar.attrib; appears in:

`envar`

%local.epigraph.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.epigraph.attrib; appears in:

`epigraph`

%local.equation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.equation.attrib; appears in:

`equation`

%local.errorcode.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.errorcode.attrib; appears in:

errorcode

%local.errorname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.errorname.attrib; appears in:

errorname

%local.errortext.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.errortext.attrib; appears in:

errortext

%local.errortype.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.errortype.attrib; appears in:

errortype

%local.example.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.example.attrib; appears in:

example

%local.exceptionname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.exceptionname.attrib; appears in:

exceptionname

%local.fax.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.fax.attrib; appears in:

fax

%local.fieldsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.fieldsynopsis.attrib; appears in:

`fieldsynopsis`

%local.figure.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.figure.attrib; appears in:

`figure`

%local.filename.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.filename.attrib; appears in:

`filename`

%local.firstname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.firstname.attrib; appears in:

`firstname`

%local.firstterm.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.firstterm.attrib; appears in:

`firstterm`

%local.footnote.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.footnote.attrib; appears in:

`footnote`

%local.footnoteref.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.footnoteref.attrib; appears in:

`footnoteref`

%local.foreignphrase.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.foreignphrase.attrib; appears in:

`foreignphrase`

%local.formalpara.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.formalpara.attrib; appears in:

`formalpara`

%local.funcdef.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.funcdef.attrib; appears in:

`funcdef`

%local.funcparams.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.funcparams.attrib; appears in:

`funcparams`

%local.funcprototype.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.funcprototype.attrib; appears in:

`funcprototype`

%local.funcsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.funcsynopsis.attrib; appears in:

`funcsynopsis`

%local.funcsynopsisinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.funcsynopsisinfo.attrib; appears in:

`funcsynopsisinfo`

%local.function.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.function.attrib; appears in:

`function`

%local.glossary.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossary.attrib; appears in:

`glossary`

%local.glossaryinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossaryinfo.attrib; appears in:

`glossaryinfo`

%local.glossdef.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossdef.attrib; appears in:

`glossdef`

%local.glossdiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossdiv.attrib; appears in:

`glossdiv`

%local.glossentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossentry.attrib; appears in:

`glossentry`

%local.glosslist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glosslist.attrib; appears in:

`glosslist`

%local.glosssee.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glosssee.attrib; appears in:

`glosssee`

%local.glossseealso.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossseealso.attrib; appears in:

`glossseealso`

%local.glossterm.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.glossterm.attrib; appears in:

`glossterm`

%local.graphic.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.graphic.attrib; appears in:

`graphic`

%local.graphicco.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.graphicco.attrib; appears in:

`graphicco`

%local.graphics.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.group.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.group.attrib; appears in:

`group`

%local.guibutton.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guibutton.attrib; appears in:

`guibutton`

%local.guiicon.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guiicon.attrib; appears in:

guiicon

%local.guilabel.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guilabel.attrib; appears in:

guilabel

%local.guimenu.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guimenu.attrib; appears in:

guimenu

%local.guimenuitem.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guimenuitem.attrib; appears in:

guimenuitem

%local.guisubmenu.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.guisubmenu.attrib; appears in:

guisubmenu

%local.hardware.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.hardware.attrib; appears in:

hardware

%local.highlights.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.highlights.attrib; appears in:

highlights

%local.holder.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.holder.attrib; appears in:

`holder`

%local.honorific.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.honorific.attrib; appears in:

`honorific`

%local.imagedata.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.imagedata.attrib; appears in:

`imagedata`

%local.imageobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.imageobject.attrib; appears in:

`imageobject`

%local.imageobjectco.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.imageobjectco.attrib; appears in:

`imageobjectco`

%local.indexdiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.indexdiv.attrib; appears in:

`indexdiv`

%local.indexentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.indexentry.attrib; appears in:

`indexentry`

%local.indexes.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.indexes.attrib; appears in:

`index` `setindex`

%local.indexinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.indexinfo.attrib; appears in:

`indexinfo`

%local.indexterm.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.indexterm.attrib; appears in:

`indexterm`

%local.informalequation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.informalequation.attrib; appears in:

`informalequation`

%local.informalexample.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.informalexample.attrib; appears in:

`informalexample`

%local.informalfigure.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.informalfigure.attrib; appears in:

`informalfigure`

%local.informaltable.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.informaltable.attrib; appears in:

`informaltable`

%local.initializer.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.initializer.attrib; appears in:

`initializer`

%local.inlineequation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.inlineequation.attrib; appears in:

`inlineequation`

%local.inlinegraphic.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.inlinegraphic.attrib; appears in:

`inlinegraphic`

%local.inlinemediaobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.inlinemediaobject.attrib; appears in:

`inlinemediaobject`

%local.interface.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.interface.attrib; appears in:

`interface`

%local.interfacename.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.interfacename.attrib; appears in:

`interfacename`

%local.invpartnumber.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.invpartnumber.attrib; appears in:

`invpartnumber`

%local.isbn.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.isbn.attrib; appears in:

`isbn`

%local.issn.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.issn.attrib; appears in:

`issn`

%local.issuenum.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.issuenum.attrib; appears in:

`issuenum`

%local.itemizedlist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.itemizedlist.attrib; appears in:

`itemizedlist`

%local.itermset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.itermset.attrib; appears in:

`itermset`

%local.jobtitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.jobtitle.attrib; appears in:

`jobtitle`

%local.keyaction.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keycap.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keycap.attrib; appears in:

`keycap`

%local.keycode.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keycode.attrib; appears in:

`keycode`

%local.keycombo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keycombo.attrib; appears in:

`keycombo`

%local.keysym.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keysym.attrib; appears in:

`keysym`

%local.keyword.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keyword.attrib; appears in:

`keyword`

%local.keywordset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.keywordset.attrib; appears in:

`keywordset`

%local.label.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.label.attrib; appears in:

`label`

%local.legalnotice.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.legalnotice.attrib; appears in:

`legalnotice`

%local.lineage.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.lineage.attrib; appears in:

`lineage`

%local.lineannotation.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.lineannotation.attrib; appears in:

`lineannotation`

%local.link.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.link.attrib; appears in:

`link`

%local.listitem.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.listitem.attrib; appears in:

`listitem`

%local.literal.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.literal.attrib; appears in:

`literal`

%local.literallayout.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.literallayout.attrib; appears in:

`literallayout`

%local.lot.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.lot.attrib; appears in:

`lot`

%local.lotentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.lotentry.attrib; appears in:

`lotentry`

%local.manvolnum.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.manvolnum.attrib; appears in:

`manvolnum`

%local.mark.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.markup.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.markup.attrib; appears in:

`markup`

%local.medialabel.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.medialabel.attrib; appears in:

`medialabel`

%local.mediaobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.mediaobject.attrib; appears in:

`mediaobject`

%local.mediaobjectco.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.mediaobjectco.attrib; appears in:

`mediaobjectco`

%local.member.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.member.attrib; appears in:

`member`

%local.menuchoice.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.menuchoice.attrib; appears in:

`menuchoice`

%local.methodname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.methodname.attrib; appears in:

`methodname`

%local.methodparam.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.methodparam.attrib; appears in:

`methodparam`

%local.methodsynopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.methodsynopsis.attrib; appears in:

`methodsynopsis`

%local.modespec.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.modespec.attrib; appears in:

`modespec`

%local.modifier.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.modifier.attrib; appears in:

`modifier`

%local.mousebutton.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.mousebutton.attrib; appears in:

`mousebutton`

%local.msg.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msg.attrib; appears in:

`msg`

%local.msgaud.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgaud.attrib; appears in:

`msgaud`

%local.msgentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgentry.attrib; appears in:

`msgentry`

%local.msgexplan.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgexplan.attrib; appears in:

msgexplan

%local.msginfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msginfo.attrib; appears in:

msginfo

%local.msglevel.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msglevel.attrib; appears in:

msglevel

%local.msgmain.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgmain.attrib; appears in:

msgmain

%local.msgorig.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgorig.attrib; appears in:

msgorig

%local.msgrel.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgrel.attrib; appears in:

msgrel

%local.msgset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgset.attrib; appears in:

msgset

%local.msgsub.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgsub.attrib; appears in:

`msgsub`

%local.msgtext.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.msgtext.attrib; appears in:

`msgtext`

%local.objectdata.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.objectinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.objectinfo.attrib; appears in:

`objectinfo`

%local.olink.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.olink.attrib; appears in:

`olink`

%local.ooclass.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.ooclass.attrib; appears in:

`ooclass`

%local.ooexception.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.ooexception.attrib; appears in:

`ooexception`

%local.oointerface.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.oointerface.attrib; appears in:

`oointerface`

%local.option.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.option.attrib; appears in:

`option`

%local.optional.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.optional.attrib; appears in:

`optional`

%local.orderedlist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.orderedlist.attrib; appears in:

`orderedlist`

%local.orgdiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.orgdiv.attrib; appears in:

`orgdiv`

%local.orgname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.orgname.attrib; appears in:

`orgname`

%local.otheraddr.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.otheraddr.attrib; appears in:

`otheraddr`

%local.othercredit.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.othercredit.attrib; appears in:

`othercredit`

%local.othername.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.othername.attrib; appears in:

`othername`

%local.pagenums.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.pagenums.attrib; appears in:

`pagenums`

%local.para.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.para.attrib; appears in:

`para`

%local.paramdef.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.paramdef.attrib; appears in:

`paramdef`

%local.parameter.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.parameter.attrib; appears in:

`parameter`

%local.part.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.part.attrib; appears in:

`part`

%local.partinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.partinfo.attrib; appears in:

`partinfo`

%local.partintro.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.partintro.attrib; appears in:

`partintro`

%local.personblurb.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.personblurb.attrib; appears in:

`personblurb`

%local.personname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.personname.attrib; appears in:

`personname`

%local.phone.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.phone.attrib; appears in:

`phone`

%local.phrase.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.phrase.attrib; appears in:

`phrase`

%local.pob.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.pob.attrib; appears in:

`pob`

%local.postcode.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.postcode.attrib; appears in:

`postcode`

%local.preface.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.preface.attrib; appears in:

`preface`

%local.prefaceinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.prefaceinfo.attrib; appears in:

`prefaceinfo`

%local.primsecter.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.primsecter.attrib; appears in:

`primary` `secondary` `tertiary`

%local.primsecterie.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.primsecterie.attrib; appears in:

`primaryie` `secondaryie` `tertiaryie`

%local.printhistory.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.printhistory.attrib; appears in:

`printhistory`

%local.procedure.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.procedure.attrib; appears in:

`procedure`

%local.productname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.productname.attrib; appears in:

`productname`

%local.productnumber.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.productnumber.attrib; appears in:

`productnumber`

%local.programlisting.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.programlisting.attrib; appears in:

programlisting

%local.programlistingco.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.programlistingco.attrib; appears in:

programlistingco

%local.prompt.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.prompt.attrib; appears in:

prompt

%local.property.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.property.attrib; appears in:

property

%local.pubdate.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.pubdate.attrib; appears in:

pubdate

%local.publisher.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.publisher.attrib; appears in:

publisher

%local.publishername.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.publishername.attrib; appears in:

publishername

%local.pubsnumber.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.pubsnumber.attrib; appears in:

pubsnumber

%local.qandadiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.qandadiv.attrib; appears in:

qandadiv

%local.qandaentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.qandaentry.attrib; appears in:

qandaentry

%local.qandaset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.qandaset.attrib; appears in:

qandaset

%local.question.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.question.attrib; appears in:

question

%local.quote.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.quote.attrib; appears in:

quote

%local.refclass.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refclass.attrib; appears in:

refclass

%local.refdescriptor.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refdescriptor.attrib; appears in:

`refdescriptor`

%local.refentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refentry.attrib; appears in:

`refentry`

%local.refentryinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refentryinfo.attrib; appears in:

`refentryinfo`

%local.refentrytitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refentrytitle.attrib; appears in:

`refentrytitle`

%local.reference.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.reference.attrib; appears in:

`reference`

%local.referenceinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.referenceinfo.attrib; appears in:

`referenceinfo`

%local.refmeta.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refmeta.attrib; appears in:

`refmeta`

%local.refmiscinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refmiscinfo.attrib; appears in:

`refmiscinfo`

%local.refname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refname.attrib; appears in:

`refname`

%local.refnamediv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refnamediv.attrib; appears in:

`refnamediv`

%local.refpurpose.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refpurpose.attrib; appears in:

`refpurpose`

%local.refsect1.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect1.attrib; appears in:

`refsect1`

%local.refsect1info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect1info.attrib; appears in:

`refsect1info`

%local.refsect2.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect2.attrib; appears in:

`refsect2`

%local.refsect2info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect2info.attrib; appears in:

`refsect2info`

%local.refsect3.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect3.attrib; appears in:

`refsect3`

%local.refsect3info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsect3info.attrib; appears in:

`refsect3info`

%local.refsection.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsection.attrib; appears in:

`refsection`

%local.refsectioninfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsectioninfo.attrib; appears in:

`refsectioninfo`

%local.refsynopsisdiv.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsynopsisdiv.attrib; appears in:

`refsynopsisdiv`

%local.refsynopsisdivinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.refsynopsisdivinfo.attrib; appears in:

`refsynopsisdivinfo`

%local.releaseinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.releaseinfo.attrib; appears in:

`releaseinfo`

%local.remark.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.remark.attrib; appears in:

`remark`

%local.replaceable.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.replaceable.attrib; appears in:

`replaceable`

%local.returnvalue.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.returnvalue.attrib; appears in:

`returnvalue`

%local.revdescription.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.revdescription.attrib; appears in:

`revdescription`

%local.revhistory.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.revhistory.attrib; appears in:

`revhistory`

%local.revision.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.revision.attrib; appears in:

`revision`

%local.revnumber.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.revnumber.attrib; appears in:

`revnumber`

%local.revremark.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.revremark.attrib; appears in:

`revremark`

%local.sbr.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sbr.attrib; appears in:

`sbr`

%local.screen.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.screen.attrib; appears in:

screen

%local.screenco.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.screenco.attrib; appears in:

screenco

%local.screeninfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.screeninfo.attrib; appears in:

screeninfo

%local.screenshot.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.screenshot.attrib; appears in:

screenshot

%local.sect1.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect1.attrib; appears in:

sect1

%local.sect1info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect1info.attrib; appears in:

sect1info

%local.sect2.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect2.attrib; appears in:

sect2

%local.sect2info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect2info.attrib; appears in:

sect2info

%local.sect3.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect3.attrib; appears in:

sect3

%local.sect3info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect3info.attrib; appears in:

sect3info

%local.sect4.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect4.attrib; appears in:

sect4

%local.sect4info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect4info.attrib; appears in:

sect4info

%local.sect5.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect5.attrib; appears in:

sect5

%local.sect5info.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sect5info.attrib; appears in:

sect5info

%local.section.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.section.attrib; appears in:

section

%local.sectioninfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sectioninfo.attrib; appears in:

sectioninfo

%local.seealsoie.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seealsoie.attrib; appears in:

`seealsoie`

%local.seeie.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seeie.attrib; appears in:

`seeie`

%local.seeseealso.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seeseealso.attrib; appears in:

`see` `seealso`

%local.seg.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seg.attrib; appears in:

`seg`

%local.seglistitem.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seglistitem.attrib; appears in:

`seglistitem`

%local.segmentedlist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.segmentedlist.attrib; appears in:

`segmentedlist`

%local.segtitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.segtitle.attrib; appears in:

`segtitle`

%local.seriesvolnums.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.seriesvolnums.attrib; appears in:

`seriesvolnums`

%local.set.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.set.attrib; appears in:

`set`

%local.setindexinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.setindexinfo.attrib; appears in:

`setindexinfo`

%local.setinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.setinfo.attrib; appears in:

`setinfo`

%local.sgmltag.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sgmltag.attrib; appears in:

`sgmltag`

%local.shortaffil.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.shortaffil.attrib; appears in:

`shortaffil`

%local.shortcut.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.shortcut.attrib; appears in:

`shortcut`

%local.sidebar.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sidebar.attrib; appears in:

`sidebar`

%local.sidebarinfo.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.sidebarinfo.attrib; appears in:

`sidebarinfo`

%local.simpara.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.simpara.attrib; appears in:

`simpara`

%local.simplelist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.simplelist.attrib; appears in:

`simplelist`

%local.simplemsgentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.simplemsgentry.attrib; appears in:

`simplemsgentry`

%local.simplesect.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.simplesect.attrib; appears in:

`simplesect`

%local.ssscript.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.ssscript.attrib; appears in:

`subscript` `superscript`

%local.state.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.state.attrib; appears in:

`state`

%local.status.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.step.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.step.attrib; appears in:

`step`

%local.street.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.street.attrib; appears in:

`street`

%local.structfield.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.structfield.attrib; appears in:

`structfield`

%local.structname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.structname.attrib; appears in:

`structname`

%local.subject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.subject.attrib; appears in:

`subject`

%local.subjectset.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.subjectset.attrib; appears in:

`subjectset`

%local.subjectterm.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.subjectterm.attrib; appears in:

`subjectterm`

%local.substeps.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.substeps.attrib; appears in:

`substeps`

%local.subtitle.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.subtitle.attrib; appears in:

`subtitle`

%local.surname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.surname.attrib; appears in:

`surname`

%local.symbol.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.symbol.attrib; appears in:

`symbol`

%local.synopfragment.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.synopfragment.attrib; appears in:

`synopfragment`

%local.synopfragmentref.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.synopfragmentref.attrib; appears in:

`synopfragmentref`

%local.synopsis.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.synopsis.attrib; appears in:

`synopsis`

%local.systemitem.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.systemitem.attrib; appears in:

`systemitem`

%local.term.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.term.attrib; appears in:

`term`

%local.textdata.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.textdata.attrib; appears in:

`textdata`

%local.textobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.textobject.attrib; appears in:

`textobject`

%local.title.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.title.attrib; appears in:

`title`

%local.titleabbrev.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.titleabbrev.attrib; appears in:

`titleabbrev`

%local.toc.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toc.attrib; appears in:

`toc`

%local.tocback.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.tocback.attrib; appears in:

`tocback`

%local.tocchap.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.tocchap.attrib; appears in:

tocchap

%local.tocentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.tocentry.attrib; appears in:

tocentry

%local.tocfront.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.tocfront.attrib; appears in:

tocfront

%local.toclevel1.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toclevel1.attrib; appears in:

toclevel1

%local.toclevel2.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toclevel2.attrib; appears in:

toclevel2

%local.toclevel3.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toclevel3.attrib; appears in:

toclevel3

%local.toclevel4.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toclevel4.attrib; appears in:

toclevel4

%local.toclevel5.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.toclevel5.attrib; appears in:

toclevel5

%local.tocpart.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.tocpart.attrib; appears in:

`tocpart`

%local.token.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.token.attrib; appears in:

`token`

%local.trademark.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.trademark.attrib; appears in:

`trademark`

%local.type.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.type.attrib; appears in:

`type`

%local.ulink.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.ulink.attrib; appears in:

`ulink`

%local.userinput.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.userinput.attrib; appears in:

`userinput`

%local.varargs.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.varargs.attrib; appears in:

`varargs`

%local.variablelist.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.variablelist.attrib; appears in:

`variablelist`

%local.varlistentry.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.varlistentry.attrib; appears in:

`varlistentry`

%local.varname.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.varname.attrib; appears in:

`varname`

%local.videodata.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.videodata.attrib; appears in:

`videodata`

%local.videoobject.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.videoobject.attrib; appears in:

`videoobject`

%local.void.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.void.attrib; appears in:

`void`

%local.volumenum.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.volumenum.attrib; appears in:

`volumenum`

%local.wordasword.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.wordasword.attrib; appears in:

`wordasword`

%local.xref.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.xref.attrib; appears in:

xref

%local.year.attrib;

Parameter Entity

The replacement text for this entity is empty.

%local.year.attrib; appears in:

year

%local.*.class; Parameter Entities

Name

%local.*.class; Parameter Entities – Allow class extension

Synopsis

These parameter entities offer a place where you can easily add new elements to the DocBook classes.

Description

The following sections identify all of the local class parameter entities in DocBook. Each of the entities is defined as empty.

%local.admon.class;

Parameter Entity

The replacement text for this entity is empty.

%local.appendix.class;

Parameter Entity

The replacement text for this entity is empty.

%local.article.class;

Parameter Entity

The replacement text for this entity is empty.

%local.base.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.book.class;

Parameter Entity

The replacement text for this entity is empty.

%local.chapter.class;

Parameter Entity

The replacement text for this entity is empty.

%local.compound.class;

Parameter Entity

The replacement text for this entity is empty.

%local.descobj.class;

Parameter Entity

The replacement text for this entity is empty.

%local.docinfo.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.formal.class;

Parameter Entity

The replacement text for this entity is empty.

%local.gen.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.genobj.class;

Parameter Entity

The replacement text for this entity is empty.

%local.index.class;

Parameter Entity

The replacement text for this entity is empty.

%local.info.class;

Parameter Entity

The replacement text for this entity is empty.

%local.informal.class;

Parameter Entity

The replacement text for this entity is empty.

%local.inlineobj.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.linespecific.class;

Parameter Entity

The replacement text for this entity is empty.

%local.link.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.list.class;

Parameter Entity

The replacement text for this entity is empty.

%local.method.synop.class;

Parameter Entity

The replacement text for this entity is empty.

%local.nav.class;

Parameter Entity

The replacement text for this entity is empty.

%local.ndxterm.class;

Parameter Entity

The replacement text for this entity is empty.

%local.notation.class;

Parameter Entity

The replacement text for this entity is empty.

%local.other.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.para.class;

Parameter Entity

The replacement text for this entity is empty.

%local.refentry.class;

Parameter Entity

The replacement text for this entity is empty.

%local.section.class;

Parameter Entity

The replacement text for this entity is empty.

%local.synop.class;

Parameter Entity

The replacement text for this entity is empty.

%local.tech.char.class;

Parameter Entity

The replacement text for this entity is empty.

%local.xref.char.class;

Parameter Entity

The replacement text for this entity is empty.

■ %local.*.mix; Parameter Entities

Name

%local.*.mix; Parameter Entities – Allow mixture extension

Synopsis

These parameter entities offer a place where you can easily add new elements to the DocBook mixtures.

Description

The following sections identify all of the local mixture parameter entities in DocBook. Each of the entities is defined as empty.

%local.admon.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.bibliocomponent.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.component.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.cptr.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.divcomponent.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.docinfo.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.example.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.figure.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.footnote.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.glossdef.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.highlights.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.indexdivcomponent.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.legalnotice.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.listpreamble.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.mediaobject.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.ndxterm.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.para.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.para.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.partcontent.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.person.ident.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.qandaset.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.refclass.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.refcomponent.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.refinline.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.refname.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.revdescription.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.sidebar.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.smallcptr.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.tabentry.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.textobject.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.title.char.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.ubiq.mix;

Parameter Entity

The replacement text for this entity is empty.

%local.word.char.mix;

Parameter Entity

The replacement text for this entity is empty.

■ %*.mix; Parameter Entities

Name

%*.mix; Parameter Entities – Parameter entities which define the DocBook mixtures

Synopsis

Mixtures are collections of classes that appear in content models. For example, the content model of Example element includes the %example.mix;. Not every element's content model is a single mixture, but elements in the same class tend to have the same mixture in their content model.

If you want to change the content model of some class of elements (lists or admonitions, perhaps), you generally want to change the definition of the appropriate mixture.

Description

The following sections identify all of the mixture parameter entities in DocBook.

%admon.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |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|sidebar |anchor|bridgehead|remark |indexterm |beginpage

%admon.mix; appears in:

caution constraintdef important

note tip warning

%bibliocomponent.mix;

Parameter Entity

abbrev|abstract|address|artpagenums|author |authorgroup|authorinitials|bibliomisc|biblioset |collab|confgroup| contractnum|contractsponsor |copyright|corpauthor|corpname|date|edition |editor|invpartnumber|isbn| issn|issuenum|orgname |biblioid|citebiblioid|bibliosource|bibliorelation|bibliocoverage |othercredit|pagenums| printhistory|productname |productnumber|pubdate|publisher|publishername |pubsnumber|releaseinfo| revhistory|seriesvolnums |subtitle|title|titleabbrev|volumenum|citetitle |personname|honorific|firstname| surname|lineage|othername|affiliation |authorblurb|contrib |indexterm

%bibliocomponent.mix; appears in:

biblioentry bibliomixed bibliomset

biblioset

%component.mix;

Parameter Entity

calloutlist|glosslist|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 |msgset|procedure|sidebar| qandaset |productionset|constraintdef |anchor|bridgehead|remark|highlights |abstract|authorblurb|epigraph |indexterm |beginpage

%component.mix; appears in:

bibliodiv bibliography blockquote

callout glossary glossdiv

index listitem msgexplan

msgtext procedure setindex

step

%cptr.char.mix;

Parameter Entity

#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|constant|markup|medialabel |menuchoice| mousebutton|option|optional|parameter |prompt|property|replaceable|returnvalue|sgmltag|structfield | structname|symbol|systemitem|token|type|userinput|varname |nonterminal |anchor |remark|subscript| superscript |inlinegraphic|inlinemediaobject |indexterm |beginpage

%cptr.char.mix; appears in:

action classsynopsisinfo command

computeroutput database filename

funcparams funcsynopsisinfo function

hardware interfacename keycap

literal option optional

parameter property systemitem

userinput

%divcomponent.mix;

Parameter Entity

calloutlist|glosslist|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 |msgset|procedure|sidebar| qandaset |productionset|constraintdef |anchor|bridgehead|remark|highlights |abstract|authorblurb|epigraph |indexterm |beginpage

%divcomponent.mix; appears in:

`sect1` `sect2` `sect3`
`sect4` `sect5` `section`
`simplesect`

%docinfo.char.mix;

Parameter Entity

#PCDATA |link|olink|ulink |emphasis|trademark |replaceable |remark|subscript|superscript |inlinegraphic| inlinemediaobject |indexterm

%docinfo.char.mix; appears in:

`ackno` `artpagenums` `authorinitials`
`bibliocoverage` `biblioid` `bibliorelation`
`bibliosource` `citebiblioid` `city`
`collabname` `confdates` `confnum`
`confsponsor` `conftitle` `contractnum`
`contractsponsor` `contrib` `corpauthor`
`corpname` `country` `date`
`edition` `email` `fax`
`firstname` `holder` `honorific`
`invpartnumber` `isbn` `issn`
`issuenum` `jobtitle` `lineage`
`modespec` `orgdiv` `orgname`
`otheraddr` `othername` `pagenums`
`phone` `pob` `postcode`
`productnumber` `pubdate` `publishername`
`pubsnumber` `refmiscinfo` `releaseinfo`
`revnumber` `revremark` `seriesvolnums`
`shortaffil` `state` `street`
`surname` `volumenum` `year`

%example.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |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 |indexterm |beginpage

%example.mix; appears in:

`example` `informalexample`

%figure.mix;

Parameter Entity

literallayout|programlisting|programlistingco|screen |screenco|screenshot |synopsis|cmdsynopsis|funcsynopsis |classsynopsis|fieldsynopsis |constructorsynopsis |destructorsynopsis |methodsynopsis |address|blockquote |graphic|graphicco|mediaobject|mediaobjectco |informalequation |informalexample |informalfigure | informaltable |indexterm |beginpage

%figure.mix; appears in:

`figure` `informalfigure`

%footnote.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |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

%footnote.mix; appears in:

`footnote`

%glossdef.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |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 |remark |indexterm |beginpage

%glossdef.mix; appears in:

`glossdef`

%highlights.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |caution|important|note| tip|warning |formalpara|para|simpara |indexterm

%highlights.mix; appears in:

`highlights`

%indexdivcomponent.mix;

Parameter Entity

itemizedlist|orderedlist|variablelist|simplelist |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 |anchor|remark |link|olink|ulink | beginpage

%indexdivcomponent.mix; appears in:

`indexdiv`

%legalnotice.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |caution|important|note| tip|warning |literallayout|programlisting|programlistingco|screen |screenco|screenshot |formalpara|para| simpara |blockquote |indexterm |beginpage

%legalnotice.mix; appears in:

`dedication` `legalnotice`

%listpreamble.mix;

Parameter Entity

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 |anchor|bridgehead|remark|highlights |abstract|authorblurb|epigraph |indexterm |beginpage

%listpreamble.mix; appears in:

`itemizedlist` `orderedlist` `variablelist`

%mediaobject.mix;

Parameter Entity

videoobject|audioobject|imageobject|textobject

%mediaobject.mix; appears in:

`inlinemediaobject` `mediaobject`

%ndxterm.char.mix;

Parameter Entity

#PCDATA |footnoteref|xref |abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase| glossterm|footnote|phrase |quote|trademark|wordasword|personname |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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt| property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type|userinput| varname |nonterminal |anchor |author|authorinitials|corpauthor|modespec|othercredit |productname| productnumber|revhistory |remark|subscript|superscript |inlinegraphic|inlinemediaobject

%ndxterm.char.mix; appears in:

`primary` `primaryie` `secondary`

`secondaryie` `see` `seealso`

`seealsoie` `seeie` `tertiary`

`tertiaryie`

%para.char.mix;

Parameter Entity

#PCDATA |footnoteref|xref |abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase| glossterm|footnote|phrase |quote|trademark|wordasword|personname |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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt| property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type|userinput|

varname |nonterminal |anchor |author|authorinitials|corpauthor|modespec|othercredit |productname| productnumber|revhistory |remark|subscript|superscript |inlinegraphic|inlinemediaobject|inlineequation |synopsis|cmdsynopsis|funcsynopsis |classsynopsis|fieldsynopsis |constructorsynopsis |destructorsynopsis |methodsynopsis |indexterm |beginpage

%para.char.mix; appears in:

`application` `attribution` `bibliomisc`
`citation` `citetitle` `emphasis`
`foreignphrase` `glosssee` `glossseealso`
`glossterm` `lineannotation` `link`
`literallayout` `lotentry` `member`
`msgaud` `olink` `para`
`phrase` `productname` `programlisting`
`quote` `refentrytitle` `remark`
`screen` `screeninfo` `seg`
`simpara` `synopsis` `term`
`tocback` `tocentry` `tocfront`
`ulink`

%para.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |caution|important|note| tip|warning |literallayout|programlisting|programlistingco|screen |screenco|screenshot |address|blockquote |graphic|graphicco|mediaobject|mediaobjectco |informalequation |informalexample |informalfigure | informaltable |equation|example|figure|table

%para.mix; appears in:

`para`

%partcontent.mix;

Parameter Entity

appendix |chapter |toc|lot|index|glossary|bibliography |article |preface|refentry |reference

%partcontent.mix; appears in:

`part`

%person.ident.mix;

Parameter Entity

honorific|firstname|surname|lineage|othername|affiliation |authorblurb|contrib

%person.ident.mix; appears in:

`address` `author` `editor`
`othercredit`

%qandaset.mix;

Parameter Entity

calloutlist|glosslist|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

%qandaset.mix; appears in:

`answer` `qandadiv` `qandaset`

`question`

%refclass.char.mix;

Parameter Entity

#PCDATA |application

%refclass.char.mix; appears in:

`refclass`

%refcomponent.mix;

Parameter Entity

calloutlist|glosslist|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 |msgset|procedure|sidebar| qandaset |productionset|constraintdef |anchor|bridgehead|remark|highlights |abstract|authorblurb|epigraph |indexterm |beginpage

%refcomponent.mix; appears in:

`refsect1` `refsect2` `refsect3`

`refsection` `refsynopsisdiv`

%refinline.char.mix;

Parameter Entity

#PCDATA |footnoteref|xref |abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase| glossterm|footnote|phrase |quote|trademark|wordasword|personname |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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt| property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type|userinput| varname |nonterminal |anchor |author|authorinitials|corpauthor|modespec|othercredit |productname| productnumber|revhistory |remark|subscript|superscript |indexterm |beginpage

%refinline.char.mix; appears in:

`refpurpose`

%refname.char.mix;

Parameter Entity

#PCDATA |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|constant|markup|medialabel |menuchoice|mousebutton|option| optional|parameter |prompt|property|replaceable|returnvalue|sgmltag|structfield |structname|symbol| systemitem|token|type|userinput|varname |nonterminal

%refname.char.mix; appears in:

`refdescriptor` `refname`

%revdescription.mix;

Parameter Entity

calloutlist|glosslist|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

%revdescription.mix; appears in:

`revdescription`

%sidebar.mix;

Parameter Entity

calloutlist|glosslist|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 |beginpage

%sidebar.mix; appears in:

`sidebar`

%smallcptr.char.mix;

Parameter Entity

#PCDATA |replaceable |inlinegraphic|inlinemediaobject |indexterm |beginpage

%smallcptr.char.mix; appears in:

`accel` `classname` `constant`

`envar` `errorcode` `errorname`

`errortext` `errortype` `exceptionname`

`guibutton` `guiicon` `guilabel`

`guimenu` `guimenuitem` `guisubmenu`

`initializer` `interface` `keycode`

`keysym` `markup` `medialabel`

`methodname` `modifier` `mousebutton`

`msglevel` `msgorig` `prompt`

`returnvalue` `sgmltag` `structfield`

`structname` `symbol` `token`

`type` `varname`

%textobject.mix;

Parameter Entity

calloutlist|glosslist|itemizedlist|orderedlist|segmentedlist |simplelist|variablelist |caution|important|note| tip|warning |literallayout|programlisting|programlistingco|screen |screenco|screenshot |formalpara|para| simpara |blockquote

%textobject.mix; appears in:

`caption` `colophon` `textobject`

%title.char.mix;

Parameter Entity

#PCDATA |footnoteref|xref |abbrev|acronym|citation|citerefentry|citetitle|emphasis |firstterm|foreignphrase| glossterm|footnote|phrase |quote|trademark|wordasword|personname |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|constant|markup|medialabel |menuchoice|mousebutton|option|optional|parameter |prompt| property|replaceable|returnvalue|sgmltag|structfield |structname|symbol|systemitem|token|type|userinput| varname |nonterminal |anchor |author|authorinitials|corpauthor|modespec|othercredit |productname| productnumber|revhistory |remark|subscript|superscript |inlinegraphic|inlinemediaobject|inlineequation |indexterm

%title.char.mix; appears in:

`bridgehead` `segtitle` `subtitle`

`title` `titleabbrev`

%ubiq.mix;

Parameter Entity

The replacement text for this entity is empty.

%word.char.mix;

Parameter Entity

#PCDATA |acronym|emphasis|trademark |link|olink|ulink |anchor |remark|subscript|superscript |inlinegraphic| inlinemediaobject |indexterm |beginpage

%word.char.mix; appears in:

`abbrev` `acronym` `firstterm`

`label` `manvolnum` `wordasword`

■ %*.module; Parameter Entities

Name

`%*.module;` Parameter Entities – Control element definitions

Synopsis

The module parameter entities provide marked sections around a single pair of element and attribute declarations. You can selectively include or remove elements from DocBook by changing these parameter entities.

Description

The following sections identify all of the module parameter entities in DocBook.

%ISOamsa.module;

Parameter Entity

INCLUDE

%ISOamsb.module;

Parameter Entity

INCLUDE

%ISOamsc.module;

Parameter Entity

INCLUDE

%ISOamsn.module;

Parameter Entity

INCLUDE

%ISOamso.module;

Parameter Entity

INCLUDE

%ISOamsr.module;

Parameter Entity

INCLUDE

%ISObox.module;

Parameter Entity

INCLUDE

%ISOcyr1.module;

Parameter Entity

INCLUDE

%ISOcyr2.module;

Parameter Entity

INCLUDE

%ISOdia.module;

Parameter Entity

INCLUDE

%ISOgrk1.module;

Parameter Entity

INCLUDE

%ISOgrk2.module;

Parameter Entity

INCLUDE

%ISOgrk3.module;

Parameter Entity

INCLUDE

%ISOgrk4.module;

Parameter Entity

INCLUDE

%ISOlat1.module;

Parameter Entity

INCLUDE

%ISOlat2.module;

Parameter Entity

INCLUDE

%ISOnum.module;

Parameter Entity

INCLUDE

%ISOpub.module;

Parameter Entity

INCLUDE

%ISOtech.module;

Parameter Entity

INCLUDE

%abbrev.module;

Parameter Entity

INCLUDE

%abstract.module;

Parameter Entity

INCLUDE

%accel.module;

Parameter Entity

INCLUDE

%ackno.module;

Parameter Entity

INCLUDE

%acronym.module;

Parameter Entity

INCLUDE

%action.module;

Parameter Entity

INCLUDE

%address.module;

Parameter Entity

INCLUDE

%admon.module;

Parameter Entity

INCLUDE

%affiliation.module;

Parameter Entity

INCLUDE

%alt.module;

Parameter Entity

INCLUDE

%anchor.module;

Parameter Entity

INCLUDE

%answer.module;

Parameter Entity

INCLUDE

%appendix.module;

Parameter Entity

INCLUDE

%appendixinfo.module;

Parameter Entity

INCLUDE

%application.module;

Parameter Entity

INCLUDE

%area.module;

Parameter Entity

INCLUDE

%areaset.module;

Parameter Entity

INCLUDE

%areaspec.module;

Parameter Entity

INCLUDE

%arg.module;

Parameter Entity

INCLUDE

%article.module;

Parameter Entity

INCLUDE

%articleinfo.module;

Parameter Entity

INCLUDE

%artpagenums.module;

Parameter Entity

INCLUDE

%attribution.module;

Parameter Entity

INCLUDE

%audiodata.module;

Parameter Entity

INCLUDE

%audioobject.module;

Parameter Entity

INCLUDE

%author.module;

Parameter Entity

INCLUDE

%authorblurb.module;

Parameter Entity

INCLUDE

%authorgroup.module;

Parameter Entity

INCLUDE

%authorinitials.module;

Parameter Entity

INCLUDE

%beginpage.module;

Parameter Entity

INCLUDE

%bibliocoverage.module;

Parameter Entity

INCLUDE

%bibliodiv.module;

Parameter Entity

INCLUDE

%biblioentry.module;

Parameter Entity

INCLUDE

%bibliography.module;

Parameter Entity

INCLUDE

%bibliographyinfo.module;

Parameter Entity

INCLUDE

%biblioid.module;

Parameter Entity

INCLUDE

%bibliomisc.module;

Parameter Entity

INCLUDE

%bibliomixed.module;

Parameter Entity

INCLUDE

%bibliomset.module;

Parameter Entity

INCLUDE

%bibliorelation.module;

Parameter Entity

INCLUDE

%biblioset.module;

Parameter Entity

INCLUDE

%bibliosource.module;

Parameter Entity

INCLUDE

%blockinfo.module;

Parameter Entity

INCLUDE

%blockquote.module;

Parameter Entity

INCLUDE

%book.module;

Parameter Entity

INCLUDE

%bookinfo.module;

Parameter Entity

INCLUDE

%bridgehead.module;

Parameter Entity

INCLUDE

%callout.module;

Parameter Entity

INCLUDE

%calloutlist.module;

Parameter Entity

INCLUDE

%cals.table.module;

Parameter Entity

INCLUDE

%caption.module;

Parameter Entity

INCLUDE

%chapter.module;

Parameter Entity

INCLUDE

%chapterinfo.module;

Parameter Entity

INCLUDE

%citation.module;

Parameter Entity

INCLUDE

%citebiblioid.module;

Parameter Entity

INCLUDE

%citerefentry.module;

Parameter Entity

INCLUDE

%citetitle.module;

Parameter Entity

INCLUDE

%city.module;

Parameter Entity

INCLUDE

%classname.module;

Parameter Entity

INCLUDE

%classsynopsis.module;

Parameter Entity

INCLUDE

%classsynopsisinfo.module;

Parameter Entity

INCLUDE

%cmdsynopsis.module;

Parameter Entity

INCLUDE

%co.module;

Parameter Entity

INCLUDE

%collab.module;

Parameter Entity

INCLUDE

%collabname.module;

Parameter Entity

INCLUDE

%colophon.module;

Parameter Entity

INCLUDE

%command.module;

Parameter Entity

INCLUDE

%computeroutput.module;

Parameter Entity

INCLUDE

%confdates.module;

Parameter Entity

INCLUDE

%confgroup.module;

Parameter Entity

INCLUDE

%confnum.module;

Parameter Entity

INCLUDE

%confsponsor.module;

Parameter Entity

INCLUDE

%conftitle.module;

Parameter Entity

INCLUDE

%constant.module;

Parameter Entity

INCLUDE

%constructorsynopsis.module;

Parameter Entity

INCLUDE

%contractnum.module;

Parameter Entity

INCLUDE

%contractsponsor.module;

Parameter Entity

INCLUDE

%contrib.module;

Parameter Entity

INCLUDE

%copyright.module;

Parameter Entity

INCLUDE

%coref.module;

Parameter Entity

INCLUDE

%corpauthor.module;

Parameter Entity

INCLUDE

%corpname.module;

Parameter Entity

INCLUDE

%country.module;

Parameter Entity

INCLUDE

%database.module;

Parameter Entity

INCLUDE

%date.module;

Parameter Entity

INCLUDE

%dedication.module;

Parameter Entity

INCLUDE

%destructorsynopsis.module;

Parameter Entity

INCLUDE

%docbook;

External Entity

Public identifier: -//OASIS//DTD DocBook XML V4.2//EN **System identifier**: http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd

%edition.module;

Parameter Entity

INCLUDE

%editor.module;

Parameter Entity

INCLUDE

%email.module;

Parameter Entity

INCLUDE

%emphasis.module;

Parameter Entity

INCLUDE

%envar.module;

Parameter Entity

INCLUDE

%epigraph.module;

Parameter Entity

INCLUDE

%equation.module;

Parameter Entity

INCLUDE

%errorcode.module;

Parameter Entity

INCLUDE

%errorname.module;

Parameter Entity

INCLUDE

%errortext.module;

Parameter Entity

INCLUDE

%errortype.module;

Parameter Entity

INCLUDE

%example.module;

Parameter Entity

INCLUDE

%exceptionname.module;

Parameter Entity

INCLUDE

%exchange.table.module;

Parameter Entity

IGNORE

%fax.module;

Parameter Entity

INCLUDE

%fieldsynopsis.module;

Parameter Entity

INCLUDE

%figure.module;

Parameter Entity

INCLUDE

%filename.module;

Parameter Entity

INCLUDE

%firstname.module;

Parameter Entity

INCLUDE

%firstterm.module;

Parameter Entity

INCLUDE

%footnote.module;

Parameter Entity

INCLUDE

%footnoteref.module;

Parameter Entity

INCLUDE

%foreignphrase.module;

Parameter Entity

INCLUDE

%formalpara.module;

Parameter Entity

INCLUDE

%funcdef.module;

Parameter Entity

INCLUDE

%funcparams.module;

Parameter Entity

INCLUDE

%funcprototype.module;

Parameter Entity

INCLUDE

%funcsynopsis.module;

Parameter Entity

INCLUDE

%funcsynopsisinfo.module;

Parameter Entity

INCLUDE

%function.module;

Parameter Entity

INCLUDE

%glossary.module;

Parameter Entity

INCLUDE

%glossaryinfo.module;

Parameter Entity

INCLUDE

%glossdef.module;

Parameter Entity

INCLUDE

%glossdiv.module;

Parameter Entity

INCLUDE

%glossentry.module;

Parameter Entity

INCLUDE

%glosslist.module;

Parameter Entity

INCLUDE

%glosssee.module;

Parameter Entity

INCLUDE

%glossseealso.module;

Parameter Entity

INCLUDE

%glossterm.module;

Parameter Entity

INCLUDE

%graphic.module;

Parameter Entity

INCLUDE

%graphicco.module;

Parameter Entity

INCLUDE

%group.module;

Parameter Entity

INCLUDE

%guibutton.module;

Parameter Entity

INCLUDE

%guiicon.module;

Parameter Entity

INCLUDE

%guilabel.module;

Parameter Entity

INCLUDE

%guimenu.module;

Parameter Entity

INCLUDE

%guimenuitem.module;

Parameter Entity

INCLUDE

%guisubmenu.module;

Parameter Entity

INCLUDE

%hardware.module;

Parameter Entity

INCLUDE

%highlights.module;

Parameter Entity

INCLUDE

%holder.module;

Parameter Entity

INCLUDE

%honorific.module;

Parameter Entity

INCLUDE

%imagedata.module;

Parameter Entity

INCLUDE

%imageobject.module;

Parameter Entity

INCLUDE

%imageobjectco.module;

Parameter Entity

INCLUDE

%indexdiv.module;

Parameter Entity

INCLUDE

%indexentry.module;

Parameter Entity

INCLUDE

%indexes.module;

Parameter Entity

INCLUDE

%indexinfo.module;

Parameter Entity

INCLUDE

%indexterm.module;

Parameter Entity

INCLUDE

%informalequation.module;

Parameter Entity

INCLUDE

%informalexample.module;

Parameter Entity

INCLUDE

%informalfigure.module;

Parameter Entity

INCLUDE

%informaltable.module;

Parameter Entity

INCLUDE

%initializer.module;

Parameter Entity

INCLUDE

%inlineequation.module;

Parameter Entity

INCLUDE

%inlinegraphic.module;

Parameter Entity

INCLUDE

%inlinemediaobject.module;

Parameter Entity

INCLUDE

%interface.module;

Parameter Entity

INCLUDE

%interfacename.module;

Parameter Entity

INCLUDE

%intermod.redecl.module;

Parameter Entity

IGNORE

%invpartnumber.module;

Parameter Entity

INCLUDE

%isbn.module;

Parameter Entity

INCLUDE

%issn.module;

Parameter Entity

INCLUDE

%issuenum.module;

Parameter Entity

INCLUDE

%itemizedlist.module;

Parameter Entity

INCLUDE

%itermset.module;

Parameter Entity

INCLUDE

%jobtitle.module;

Parameter Entity

INCLUDE

%keycap.module;

Parameter Entity

INCLUDE

%keycode.module;

Parameter Entity

INCLUDE

%keycombo.module;

Parameter Entity

INCLUDE

%keysym.module;

Parameter Entity

INCLUDE

%keyword.module;

Parameter Entity

INCLUDE

%keywordset.module;

Parameter Entity

INCLUDE

%label.module;

Parameter Entity

INCLUDE

%legalnotice.module;

Parameter Entity

INCLUDE

%lineage.module;

Parameter Entity

INCLUDE

%lineannotation.module;

Parameter Entity

INCLUDE

%link.module;

Parameter Entity

INCLUDE

%listitem.module;

Parameter Entity

INCLUDE

%literal.module;

Parameter Entity

INCLUDE

%literallayout.module;

Parameter Entity

INCLUDE

%lot.module;

Parameter Entity

INCLUDE

%lotentry.module;

Parameter Entity

INCLUDE

%manvolnum.module;

Parameter Entity

INCLUDE

%markup.module;

Parameter Entity

INCLUDE

%medialabel.module;

Parameter Entity

INCLUDE

%mediaobject.module;

Parameter Entity

INCLUDE

%mediaobjectco.module;

Parameter Entity

INCLUDE

%member.module;

Parameter Entity

INCLUDE

%menuchoice.module;

Parameter Entity

INCLUDE

%methodname.module;

Parameter Entity

INCLUDE

%methodparam.module;

Parameter Entity

INCLUDE

%methodsynopsis.module;

Parameter Entity

INCLUDE

%modespec.module;

Parameter Entity

INCLUDE

%modifier.module;

Parameter Entity

INCLUDE

%mousebutton.module;

Parameter Entity

INCLUDE

%msg.module;

Parameter Entity

INCLUDE

%msgaud.module;

Parameter Entity

INCLUDE

%msgentry.module;

Parameter Entity

INCLUDE

%msgexplan.module;

Parameter Entity

INCLUDE

%msginfo.module;

Parameter Entity

INCLUDE

%msglevel.module;

Parameter Entity

INCLUDE

%msgmain.module;

Parameter Entity

INCLUDE

%msgorig.module;

Parameter Entity

INCLUDE

%msgrel.module;

Parameter Entity

INCLUDE

%msgset.module;

Parameter Entity

INCLUDE

%msgsub.module;

Parameter Entity

INCLUDE

%msgtext.module;

Parameter Entity

INCLUDE

%objectinfo.module;

Parameter Entity

INCLUDE

%olink.module;

Parameter Entity

INCLUDE

%ooclass.module;

Parameter Entity

INCLUDE

%ooexception.module;

Parameter Entity

INCLUDE

%oointerface.module;

Parameter Entity

INCLUDE

%option.module;

Parameter Entity

INCLUDE

%optional.module;

Parameter Entity

INCLUDE

%orderedlist.module;

Parameter Entity

INCLUDE

%orgdiv.module;

Parameter Entity

INCLUDE

%orgname.module;

Parameter Entity

INCLUDE

%otheraddr.module;

Parameter Entity

INCLUDE

%othercredit.module;

Parameter Entity

INCLUDE

%othername.module;

Parameter Entity

INCLUDE

%pagenums.module;

Parameter Entity

INCLUDE

%para.module;

Parameter Entity

INCLUDE

%paramdef.module;

Parameter Entity

INCLUDE

%parameter.module;

Parameter Entity

INCLUDE

%part.module;

Parameter Entity

INCLUDE

%partinfo.module;

Parameter Entity

INCLUDE

%partintro.module;

Parameter Entity

INCLUDE

%person.ident.module;

Parameter Entity

INCLUDE

%personblurb.module;

Parameter Entity

INCLUDE

%personname.module;

Parameter Entity

INCLUDE

%phone.module;

Parameter Entity

INCLUDE

%phrase.module;

Parameter Entity

INCLUDE

%pob.module;

Parameter Entity

INCLUDE

%postcode.module;

Parameter Entity

INCLUDE

%preface.module;

Parameter Entity

INCLUDE

%prefaceinfo.module;

Parameter Entity

INCLUDE

%primsecter.module;

Parameter Entity

INCLUDE

%primsecterie.module;

Parameter Entity

INCLUDE

%printhistory.module;

Parameter Entity

INCLUDE

%procedure.module;

Parameter Entity

INCLUDE

%productname.module;

Parameter Entity

INCLUDE

%productnumber.module;

Parameter Entity

INCLUDE

%programlisting.module;

Parameter Entity

INCLUDE

%programlistingco.module;

Parameter Entity

INCLUDE

%prompt.module;

Parameter Entity

INCLUDE

%property.module;

Parameter Entity

INCLUDE

%pubdate.module;

Parameter Entity

INCLUDE

%publisher.module;

Parameter Entity

INCLUDE

%publishername.module;

Parameter Entity

INCLUDE

%pubsnumber.module;

Parameter Entity

INCLUDE

%qandadiv.module;

Parameter Entity

INCLUDE

%qandaentry.module;

Parameter Entity

INCLUDE

%qandaset.module;

Parameter Entity

INCLUDE

%question.module;

Parameter Entity

INCLUDE

%quote.module;

Parameter Entity

INCLUDE

%refclass.module;

Parameter Entity

INCLUDE

%refdescriptor.module;

Parameter Entity

INCLUDE

%refentry.module;

Parameter Entity

INCLUDE

%refentryinfo.module;

Parameter Entity

INCLUDE

%refentrytitle.module;

Parameter Entity

INCLUDE

%reference.module;

Parameter Entity

INCLUDE

%referenceinfo.module;

Parameter Entity

INCLUDE

%refmeta.module;

Parameter Entity

INCLUDE

%refmiscinfo.module;

Parameter Entity

INCLUDE

%refname.module;

Parameter Entity

INCLUDE

%refnamediv.module;

Parameter Entity

INCLUDE

%refpurpose.module;

Parameter Entity

INCLUDE

%refsect1.module;

Parameter Entity

INCLUDE

%refsect1info.module;

Parameter Entity

INCLUDE

%refsect2.module;

Parameter Entity

INCLUDE

%refsect2info.module;

Parameter Entity

INCLUDE

%refsect3.module;

Parameter Entity

INCLUDE

%refsect3info.module;

Parameter Entity

INCLUDE

%refsection.module;

Parameter Entity

INCLUDE

%refsectioninfo.module;

Parameter Entity

INCLUDE

%refsynopsisdiv.module;

Parameter Entity

INCLUDE

%refsynopsisdivinfo.module;

Parameter Entity

INCLUDE

%releaseinfo.module;

Parameter Entity

INCLUDE

%remark.module;

Parameter Entity

INCLUDE

%replaceable.module;

Parameter Entity

INCLUDE

%returnvalue.module;

Parameter Entity

INCLUDE

%revdescription.module;

Parameter Entity

INCLUDE

%revhistory.module;

Parameter Entity

INCLUDE

%revision.module;

Parameter Entity

INCLUDE

%revnumber.module;

Parameter Entity

INCLUDE

%revremark.module;

Parameter Entity

INCLUDE

%sbr.module;

Parameter Entity

INCLUDE

%screen.module;

Parameter Entity

INCLUDE

%screenco.module;

Parameter Entity

INCLUDE

%screeninfo.module;

Parameter Entity

INCLUDE

%screenshot.module;

Parameter Entity

INCLUDE

%sect1.module;

Parameter Entity

INCLUDE

%sect2.module;

Parameter Entity

INCLUDE

%sect3.module;

Parameter Entity

INCLUDE

%sect4.module;

Parameter Entity

INCLUDE

%sect5.module;

Parameter Entity

INCLUDE

%section.module;

Parameter Entity

INCLUDE

%sectioninfo.module;

Parameter Entity

INCLUDE

%seealsoie.module;

Parameter Entity

INCLUDE

%seeie.module;

Parameter Entity

INCLUDE

%seeseealso.module;

Parameter Entity

INCLUDE

%seg.module;

Parameter Entity

INCLUDE

%seglistitem.module;

Parameter Entity

INCLUDE

%segmentedlist.module;

Parameter Entity

INCLUDE

%segtitle.module;

Parameter Entity

INCLUDE

%seriesvolnums.module;

Parameter Entity

INCLUDE

%set.module;

Parameter Entity

INCLUDE

%setindexinfo.module;

Parameter Entity

INCLUDE

%setinfo.module;

Parameter Entity

INCLUDE

%sgmltag.module;

Parameter Entity

INCLUDE

%shortaffil.module;

Parameter Entity

INCLUDE

%shortcut.module;

Parameter Entity

INCLUDE

%sidebar.module;

Parameter Entity

INCLUDE

%sidebarinfo.module;

Parameter Entity

INCLUDE

%simpara.module;

Parameter Entity

INCLUDE

%simplelist.module;

Parameter Entity

INCLUDE

%simplemsgentry.module;

Parameter Entity

INCLUDE

%simplesect.module;

Parameter Entity

INCLUDE

%ssscript.module;

Parameter Entity

INCLUDE

%state.module;

Parameter Entity

INCLUDE

%step.module;

Parameter Entity

INCLUDE

%street.module;

Parameter Entity

INCLUDE

%structfield.module;

Parameter Entity

INCLUDE

%structname.module;

Parameter Entity

INCLUDE

%subject.module;

Parameter Entity

INCLUDE

%subjectset.module;

Parameter Entity

INCLUDE

%subjectterm.module;

Parameter Entity

INCLUDE

%substeps.module;

Parameter Entity

INCLUDE

%subtitle.module;

Parameter Entity

INCLUDE

%surname.module;

Parameter Entity

INCLUDE

%symbol.module;

Parameter Entity

INCLUDE

%synopfragment.module;

Parameter Entity

INCLUDE

%synopfragmentref.module;

Parameter Entity

INCLUDE

%synopsis.module;

Parameter Entity

INCLUDE

%systemitem.module;

Parameter Entity

INCLUDE

%table.module;

Parameter Entity

INCLUDE

%term.module;

Parameter Entity

INCLUDE

%textdata.module;

Parameter Entity

INCLUDE

%textobject.module;

Parameter Entity

INCLUDE

%title.module;

Parameter Entity

INCLUDE

%titleabbrev.module;

Parameter Entity

INCLUDE

%toc.module;

Parameter Entity

INCLUDE

%tocback.module;

Parameter Entity

INCLUDE

%tocchap.module;

Parameter Entity

INCLUDE

%tocentry.module;

Parameter Entity

INCLUDE

%tocfront.module;

Parameter Entity

INCLUDE

%toclevel1.module;

Parameter Entity

INCLUDE

%toclevel2.module;

Parameter Entity

INCLUDE

%toclevel3.module;

Parameter Entity

INCLUDE

%toclevel4.module;

Parameter Entity

INCLUDE

%toclevel5.module;

Parameter Entity

INCLUDE

%tocpart.module;

Parameter Entity

INCLUDE

%token.module;

Parameter Entity

INCLUDE

%trademark.module;

Parameter Entity

INCLUDE

%type.module;

Parameter Entity

INCLUDE

%ulink.module;

Parameter Entity

INCLUDE

%userinput.module;

Parameter Entity

INCLUDE

%varargs.module;

Parameter Entity

INCLUDE

%variablelist.module;

Parameter Entity

INCLUDE

%varlistentry.module;

Parameter Entity

INCLUDE

%varname.module;

Parameter Entity

INCLUDE

%videodata.module;

Parameter Entity

INCLUDE

%videoobject.module;

Parameter Entity

INCLUDE

%void.module;

Parameter Entity

INCLUDE

%volumenum.module;

Parameter Entity

INCLUDE

%wordasword.module;

Parameter Entity

INCLUDE

%xref.module;

Parameter Entity

INCLUDE

%year.module;

Parameter Entity

INCLUDE

%*.role.attrib; Parameter Entities

Name

%*.role.attrib; Parameter Entities – Parameter entities which control definition of role attributes

Synopsis

Role is a string used to classify or subclassify an element.

The declaration for the role attribute is parameterized in such a way that every element's role can be independently redefined. This provides an easy mechanism for modifying the legal values of the role attribute for particular elements.

Description

The following sections identify all of the role parameter entities in DocBook.

%abbrev.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%abbrev.role.attrib; appears in:
abbrev

%abstract.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%abstract.role.attrib; appears in:
abstract

%accel.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%accel.role.attrib; appears in:
accel

%ackno.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%ackno.role.attrib; appears in:
ackno

%acronym.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%acronym.role.attrib; appears in:
acronym

%action.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%action.role.attrib; appears in:

`action`

%address.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%address.role.attrib; appears in:

`address`

%admon.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%admon.role.attrib; appears in:

`caution` `important` `note`
`tip` `warning`

%affiliation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%affiliation.role.attrib; appears in:

`affiliation`

%alt.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%alt.role.attrib; appears in:

`alt`

%anchor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%anchor.role.attrib; appears in:

`anchor`

%answer.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%answer.role.attrib; appears in:

`answer`

%appendix.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%appendix.role.attrib; appears in:

`appendix`

%appendixinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%appendixinfo.role.attrib; appears in:

`appendixinfo`

%application.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%application.role.attrib; appears in:

`application`

%area.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%area.role.attrib; appears in:

`area`

%areaset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%areaset.role.attrib; appears in:

`areaset`

%areaspec.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%areaspec.role.attrib; appears in:

`areaspec`

%arg.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%arg.role.attrib; appears in:

`arg`

%article.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%article.role.attrib; appears in:

`article`

%articleinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%articleinfo.role.attrib; appears in:

`articleinfo`

%artpagenums.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%artpagenums.role.attrib; appears in:

`artpagenums`

%attribution.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%attribution.role.attrib; appears in:

`attribution`

%audiodata.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%audiodata.role.attrib; appears in:

`audiodata`

%audioobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%audioobject.role.attrib; appears in:

`audioobject`

%author.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%author.role.attrib; appears in:

`author`

%authorblurb.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%authorblurb.role.attrib; appears in:

`authorblurb`

%authorgroup.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%authorgroup.role.attrib; appears in:

`authorgroup`

%authorinitials.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%authorinitials.role.attrib; appears in:

`authorinitials`

%beginpage.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%beginpage.role.attrib; appears in:

`beginpage`

%bibliocoverage.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliocoverage.role.attrib; appears in:

`bibliocoverage`

%bibliodiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliodiv.role.attrib; appears in:

`bibliodiv`

%biblioentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%biblioentry.role.attrib; appears in:

`biblioentry`

%bibliography.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliography.role.attrib; appears in:

`bibliography`

%bibliographyinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliographyinfo.role.attrib; appears in:

`bibliographyinfo`

%biblioid.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%biblioid.role.attrib; appears in:

`biblioid`

%bibliomisc.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliomisc.role.attrib; appears in:

`bibliomisc`

%bibliomixed.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliomixed.role.attrib; appears in:

`bibliomixed`

%bibliomset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliomset.role.attrib; appears in:

`bibliomset`

%bibliorelation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliorelation.role.attrib; appears in:

`bibliorelation`

%biblioset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%biblioset.role.attrib; appears in:

`biblioset`

%bibliosource.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bibliosource.role.attrib; appears in:

`bibliosource`

%blockinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%blockinfo.role.attrib; appears in:

`blockinfo`

%blockquote.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%blockquote.role.attrib; appears in:

`blockquote`

%book.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%book.role.attrib; appears in:

`book`

%bookinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bookinfo.role.attrib; appears in:

`bookinfo`

%bridgehead.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%bridgehead.role.attrib; appears in:

`bridgehead`

%callout.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%callout.role.attrib; appears in:

`callout`

%calloutlist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%calloutlist.role.attrib; appears in:

`calloutlist`

%caption.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%caption.role.attrib; appears in:

`caption`

%chapter.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%chapter.role.attrib; appears in:

`chapter`

%chapterinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%chapterinfo.role.attrib; appears in:

`chapterinfo`

%citation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%citation.role.attrib; appears in:

`citation`

%citebiblioid.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%citebiblioid.role.attrib; appears in:

`citebiblioid`

%citerefentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%citerefentry.role.attrib; appears in:

`citerefentry`

%citetitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%citetitle.role.attrib; appears in:

`citetitle`

%city.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%city.role.attrib; appears in:

`city`

%classname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%classname.role.attrib; appears in:

`classname`

%classsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%classsynopsis.role.attrib; appears in:

`classsynopsis`

%classsynopsisinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%classsynopsisinfo.role.attrib; appears in:

`classsynopsisinfo`

%cmdsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%cmdsynopsis.role.attrib; appears in:

`cmdsynopsis`

%co.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%co.role.attrib; appears in:

`co`

%collab.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%collab.role.attrib; appears in:

`collab`

%collabname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%collabname.role.attrib; appears in:

`collabname`

%colophon.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%colophon.role.attrib; appears in:

`colophon`

%command.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%command.role.attrib; appears in:

`command`

%computeroutput.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%computeroutput.role.attrib; appears in:

`computeroutput`

%confdates.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%confdates.role.attrib; appears in:

`confdates`

%confgroup.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%confgroup.role.attrib; appears in:

`confgroup`

%confnum.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%confnum.role.attrib; appears in:

`confnum`

%confsponsor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%confsponsor.role.attrib; appears in:

`confsponsor`

%conftitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%conftitle.role.attrib; appears in:

`conftitle`

%constant.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%constant.role.attrib; appears in:

`constant`

%constructorsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%constructorsynopsis.role.attrib; appears in:

`constructorsynopsis`

%contractnum.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%contractnum.role.attrib; appears in:

`contractnum`

%contractsponsor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%contractsponsor.role.attrib; appears in:

`contractsponsor`

%contrib.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%contrib.role.attrib; appears in:

`contrib`

%copyright.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%copyright.role.attrib; appears in:

`copyright`

%coref.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%coref.role.attrib; appears in:

`coref`

%corpauthor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%corpauthor.role.attrib; appears in:

`corpauthor`

%corpname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%corpname.role.attrib; appears in:

`corpname`

%country.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%country.role.attrib; appears in:

`country`

%database.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%database.role.attrib; appears in:

`database`

%date.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%date.role.attrib; appears in:

`date`

%dedication.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%dedication.role.attrib; appears in:

`dedication`

%destructorsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%destructorsynopsis.role.attrib; appears in:

`destructorsynopsis`

%edition.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%edition.role.attrib; appears in:

`edition`

%editor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%editor.role.attrib; appears in:

`editor`

%email.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%email.role.attrib; appears in:

`email`

%emphasis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%emphasis.role.attrib; appears in:

`emphasis`

%envar.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%envar.role.attrib; appears in:

`envar`

%epigraph.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%epigraph.role.attrib; appears in:

`epigraph`

%equation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%equation.role.attrib; appears in:

`equation`

%errorcode.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%errorcode.role.attrib; appears in:

`errorcode`

%errorname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%errorname.role.attrib; appears in:

`errorname`

%errortext.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%errortext.role.attrib; appears in:

`errortext`

%errortype.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%errortype.role.attrib; appears in:

`errortype`

%example.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%example.role.attrib; appears in:

`example`

%exceptionname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%exceptionname.role.attrib; appears in:

`exceptionname`

%fax.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%fax.role.attrib; appears in:

`fax`

%fieldsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%fieldsynopsis.role.attrib; appears in:

`fieldsynopsis`

%figure.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%figure.role.attrib; appears in:

`figure`

%filename.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%filename.role.attrib; appears in:

`filename`

%firstname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%firstname.role.attrib; appears in:

`firstname`

%firstterm.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%firstterm.role.attrib; appears in:

`firstterm`

%footnote.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%footnote.role.attrib; appears in:

`footnote`

%footnoteref.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%footnoteref.role.attrib; appears in:

`footnoteref`

%foreignphrase.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%foreignphrase.role.attrib; appears in:

`foreignphrase`

%formalpara.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%formalpara.role.attrib; appears in:

`formalpara`

%funcdef.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%funcdef.role.attrib; appears in:

`funcdef`

%funcparams.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%funcparams.role.attrib; appears in:

`funcparams`

%funcprototype.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%funcprototype.role.attrib; appears in:

`funcprototype`

%funcsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%funcsynopsis.role.attrib; appears in:

`funcsynopsis`

%funcsynopsisinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%funcsynopsisinfo.role.attrib; appears in:

`funcsynopsisinfo`

%function.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%function.role.attrib; appears in:

`function`

%glossary.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossary.role.attrib; appears in:

`glossary`

%glossaryinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossaryinfo.role.attrib; appears in:

`glossaryinfo`

%glossdef.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossdef.role.attrib; appears in:

`glossdef`

%glossdiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossdiv.role.attrib; appears in:

`glossdiv`

%glossentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossentry.role.attrib; appears in:

`glossentry`

%glosslist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glosslist.role.attrib; appears in:

`glosslist`

%glosssee.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glosssee.role.attrib; appears in:

`glosssee`

%glossseealso.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossseealso.role.attrib; appears in:

`glossseealso`

%glossterm.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%glossterm.role.attrib; appears in:

`glossterm`

%graphic.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%graphic.role.attrib; appears in:

`graphic`

%graphicco.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%graphicco.role.attrib; appears in:

`graphicco`

%group.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%group.role.attrib; appears in:

`group`

%guibutton.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guibutton.role.attrib; appears in:

`guibutton`

%guiicon.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guiicon.role.attrib; appears in:

`guiicon`

%guilabel.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guilabel.role.attrib; appears in:

`guilabel`

%guimenu.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guimenu.role.attrib; appears in:

`guimenu`

%guimenuitem.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guimenuitem.role.attrib; appears in:

`guimenuitem`

%guisubmenu.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%guisubmenu.role.attrib; appears in:

`guisubmenu`

%hardware.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%hardware.role.attrib; appears in:

`hardware`

%highlights.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%highlights.role.attrib; appears in:

`highlights`

%holder.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%holder.role.attrib; appears in:

`holder`

%honorific.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%honorific.role.attrib; appears in:

`honorific`

%imagedata.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%imagedata.role.attrib; appears in:

`imagedata`

%imageobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%imageobject.role.attrib; appears in:

`imageobject`

%imageobjectco.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%imageobjectco.role.attrib; appears in:

`imageobjectco`

%indexdiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%indexdiv.role.attrib; appears in:

`indexdiv`

%indexentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%indexentry.role.attrib; appears in:

`indexentry`

%indexes.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%indexes.role.attrib; appears in:

`index` `setindex`

%indexinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%indexinfo.role.attrib; appears in:

`indexinfo`

%indexterm.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%indexterm.role.attrib; appears in:

`indexterm`

%informalequation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%informalequation.role.attrib; appears in:

`informalequation`

%informalexample.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%informalexample.role.attrib; appears in:

`informalexample`

%informalfigure.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%informalfigure.role.attrib; appears in:

`informalfigure`

%initializer.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%initializer.role.attrib; appears in:

`initializer`

%inlineequation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%inlineequation.role.attrib; appears in:

`inlineequation`

%inlinegraphic.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%inlinegraphic.role.attrib; appears in:

`inlinegraphic`

%inlinemediaobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%inlinemediaobject.role.attrib; appears in:

`inlinemediaobject`

%interface.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%interface.role.attrib; appears in:

`interface`

%interfacename.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%interfacename.role.attrib; appears in:

`interfacename`

%invpartnumber.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%invpartnumber.role.attrib; appears in:

`invpartnumber`

%isbn.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%isbn.role.attrib; appears in:

`isbn`

%issn.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%issn.role.attrib; appears in:

`issn`

%issuenum.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%issuenum.role.attrib; appears in:

`issuenum`

%itemizedlist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%itemizedlist.role.attrib; appears in:

`itemizedlist`

%itermset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%itermset.role.attrib; appears in:

`itermset`

%jobtitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%jobtitle.role.attrib; appears in:

`jobtitle`

%keycap.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keycap.role.attrib; appears in:

`keycap`

%keycode.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keycode.role.attrib; appears in:

`keycode`

%keycombo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keycombo.role.attrib; appears in:

`keycombo`

%keysysm.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keysysm.role.attrib; appears in:

`keysym`

%keyword.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keyword.role.attrib; appears in:

`keyword`

%keywordset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%keywordset.role.attrib; appears in:

`keywordset`

%label.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%label.role.attrib; appears in:

`label`

%legalnotice.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%legalnotice.role.attrib; appears in:

`legalnotice`

%lineage.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%lineage.role.attrib; appears in:

`lineage`

%lineannotation.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%lineannotation.role.attrib; appears in:

`lineannotation`

%link.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%link.role.attrib; appears in:

`link`

%listitem.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%listitem.role.attrib; appears in:

`listitem`

%literal.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%literal.role.attrib; appears in:

`literal`

%literallayout.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%literallayout.role.attrib; appears in:

`literallayout`

%lot.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%lot.role.attrib; appears in:

`lot`

%lotentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%lotentry.role.attrib; appears in:

`lotentry`

%markup.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%markup.role.attrib; appears in:

`markup`

%medialabel.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%medialabel.role.attrib; appears in:

`medialabel`

%mediaobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%mediaobject.role.attrib; appears in:

`mediaobject`

%mediaobjectco.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%mediaobjectco.role.attrib; appears in:

`mediaobjectco`

%member.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%member.role.attrib; appears in:

`member`

%menuchoice.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%menuchoice.role.attrib; appears in:

`menuchoice`

%methodname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%methodname.role.attrib; appears in:

`methodname`

%methodparam.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%methodparam.role.attrib; appears in:

`methodparam`

%methodsynopsis.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%methodsynopsis.role.attrib; appears in:

`methodsynopsis`

%modespec.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%modespec.role.attrib; appears in:

`modespec`

%modifier.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%modifier.role.attrib; appears in:

`modifier`

%mousebutton.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%mousebutton.role.attrib; appears in:

`mousebutton`

%msg.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msg.role.attrib; appears in:

`msg`

%msgaud.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgaud.role.attrib; appears in:

`msgaud`

%msgentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgentry.role.attrib; appears in:

`msgentry`

%msgexplan.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgexplan.role.attrib; appears in:

`msgexplan`

%msginfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msginfo.role.attrib; appears in:

`msginfo`

%msglevel.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msglevel.role.attrib; appears in:

`msglevel`

%msgmain.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgmain.role.attrib; appears in:

`msgmain`

%msgorig.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgorig.role.attrib; appears in:

`msgorig`

%msgrel.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgrel.role.attrib; appears in:

`msgrel`

%msgset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgset.role.attrib; appears in:

`msgset`

%msgsub.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgsub.role.attrib; appears in:

`msgsub`

%msgtext.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%msgtext.role.attrib; appears in:

`msgtext`

%namvolnum.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%namvolnum.role.attrib; appears in:

`manvolnum`

%objectinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%objectinfo.role.attrib; appears in:

`objectinfo`

%olink.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%olink.role.attrib; appears in:

`olink`

%ooclass.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%ooclass.role.attrib; appears in:

`ooclass`

%ooexception.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%ooexception.role.attrib; appears in:

`ooexception`

%oointerface.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%oointerface.role.attrib; appears in:

`oointerface`

%option.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%option.role.attrib; appears in:

`option`

%optional.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%optional.role.attrib; appears in:

`optional`

%orderedlist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%orderedlist.role.attrib; appears in:

`orderedlist`

%orgdiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%orgdiv.role.attrib; appears in:

`orgdiv`

%orgname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%orgname.role.attrib; appears in:

`orgname`

%otheraddr.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%otheraddr.role.attrib; appears in:

`otheraddr`

%othercredit.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%othercredit.role.attrib; appears in:

`othercredit`

%othername.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%othername.role.attrib; appears in:

`othername`

%pagenums.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%pagenums.role.attrib; appears in:

`pagenums`

%para.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%para.role.attrib; appears in:

`para`

%paramdef.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%paramdef.role.attrib; appears in:

`paramdef`

%parameter.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%parameter.role.attrib; appears in:

`parameter`

%part.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%part.role.attrib; appears in:

`part`

%partinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%partinfo.role.attrib; appears in:

`partinfo`

%partintro.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%partintro.role.attrib; appears in:

`partintro`

%personblurb.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%personblurb.role.attrib; appears in:

`personblurb`

%personname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%personname.role.attrib; appears in:

`personname`

%phone.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%phone.role.attrib; appears in:

`phone`

%phrase.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%phrase.role.attrib; appears in:

`phrase`

%pob.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%pob.role.attrib; appears in:

`pob`

%postcode.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%postcode.role.attrib; appears in:

`postcode`

%preface.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%preface.role.attrib; appears in:

`preface`

%prefaceinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%prefaceinfo.role.attrib; appears in:

`prefaceinfo`

%primsecter.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%primsecter.role.attrib; appears in:

`primary` `secondary` `tertiary`

%primsecterie.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%primsecterie.role.attrib; appears in:

`primaryie` `secondaryie` `tertiaryie`

%printhistory.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%printhistory.role.attrib; appears in:

`printhistory`

%procedure.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%procedure.role.attrib; appears in:

`procedure`

%productname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%productname.role.attrib; appears in:

`productname`

%productnumber.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%productnumber.role.attrib; appears in:

`productnumber`

%programlisting.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%programlisting.role.attrib; appears in:

`programlisting`

%programlistingco.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%programlistingco.role.attrib; appears in:

`programlistingco`

%prompt.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%prompt.role.attrib; appears in:

`prompt`

%property.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%property.role.attrib; appears in:

`property`

%pubdate.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%pubdate.role.attrib; appears in:

`pubdate`

%publisher.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%publisher.role.attrib; appears in:

`publisher`

%publishername.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%publishername.role.attrib; appears in:

`publishername`

%pubsnumber.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%pubsnumber.role.attrib; appears in:

`pubsnumber`

%qandadiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%qandadiv.role.attrib; appears in:

`qandadiv`

%qandaentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%qandaentry.role.attrib; appears in:

`qandaentry`

%qandaset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%qandaset.role.attrib; appears in:

`qandaset`

%question.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%question.role.attrib; appears in:

`question`

%quote.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%quote.role.attrib; appears in:

`quote`

%refclass.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refclass.role.attrib; appears in:

`refclass`

%refdescriptor.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refdescriptor.role.attrib; appears in:

`refdescriptor`

%refentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refentry.role.attrib; appears in:

`refentry`

%refentryinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refentryinfo.role.attrib; appears in:

`refentryinfo`

%refentrytitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refentrytitle.role.attrib; appears in:

`refentrytitle`

%reference.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%reference.role.attrib; appears in:

`reference`

%referenceinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%referenceinfo.role.attrib; appears in:

`referenceinfo`

%refmeta.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refmeta.role.attrib; appears in:

`refmeta`

%refmiscinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refmiscinfo.role.attrib; appears in:

`refmiscinfo`

%refname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refname.role.attrib; appears in:

`refname`

%refnamediv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refnamediv.role.attrib; appears in:

`refnamediv`

%refpurpose.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refpurpose.role.attrib; appears in:

`refpurpose`

%refsect1.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect1.role.attrib; appears in:

`refsect1`

%refsect1info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect1info.role.attrib; appears in:

`refsect1info`

%refsect2.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect2.role.attrib; appears in:

`refsect2`

%refsect2info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect2info.role.attrib; appears in:

`refsect2info`

%refsect3.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect3.role.attrib; appears in:

`refsect3`

%refsect3info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsect3info.role.attrib; appears in:

`refsect3info`

%refsection.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsection.role.attrib; appears in:

`refsection`

%refsectioninfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsectioninfo.role.attrib; appears in:

`refsectioninfo`

%refsynopsisdiv.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsynopsisdiv.role.attrib; appears in:

`refsynopsisdiv`

%refsynopsisdivinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%refsynopsisdivinfo.role.attrib; appears in:

`refsynopsisdivinfo`

%releaseinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%releaseinfo.role.attrib; appears in:

`releaseinfo`

%remark.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%remark.role.attrib; appears in:

`remark`

%replaceable.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%replaceable.role.attrib; appears in:

`replaceable`

%returnvalue.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%returnvalue.role.attrib; appears in:

`returnvalue`

%revdescription.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%revdescription.role.attrib; appears in:

`revdescription`

%revhistory.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%revhistory.role.attrib; appears in:

`revhistory`

%revision.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%revision.role.attrib; appears in:

`revision`

%revnumber.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%revnumber.role.attrib; appears in:

`revnumber`

%revremark.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%revremark.role.attrib; appears in:

`revremark`

%sbr.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sbr.role.attrib; appears in:

`sbr`

%screen.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%screen.role.attrib; appears in:

`screen`

%screenco.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%screenco.role.attrib; appears in:

`screenco`

%screeninfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%screeninfo.role.attrib; appears in:

`screeninfo`

%screenshot.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%screenshot.role.attrib; appears in:

`screenshot`

%sect1.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect1.role.attrib; appears in:

`sect1`

%sect1info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect1info.role.attrib; appears in:

`sect1info`

%sect2.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect2.role.attrib; appears in:

`sect2`

%sect2info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect2info.role.attrib; appears in:

`sect2info`

%sect3.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect3.role.attrib; appears in:

`sect3`

%sect3info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect3info.role.attrib; appears in:

`sect3info`

%sect4.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect4.role.attrib; appears in:

`sect4`

%sect4info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect4info.role.attrib; appears in:

`sect4info`

%sect5.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect5.role.attrib; appears in:

`sect5`

%sect5info.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sect5info.role.attrib; appears in:

`sect5info`

%section.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%section.role.attrib; appears in:

`section`

%sectioninfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sectioninfo.role.attrib; appears in:

`sectioninfo`

%seealsoie.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seealsoie.role.attrib; appears in:

`seealsoie`

%seeie.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seeie.role.attrib; appears in:

`seeie`

%seeseealso.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seeseealso.role.attrib; appears in:

`see` `seealso`

%seg.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seg.role.attrib; appears in:

`seg`

%seglistitem.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seglistitem.role.attrib; appears in:

`seglistitem`

%segmentedlist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%segmentedlist.role.attrib; appears in:

`segmentedlist`

%segtitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%segtitle.role.attrib; appears in:

`segtitle`

%seriesvolnums.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%seriesvolnums.role.attrib; appears in:

`seriesvolnums`

%set.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%set.role.attrib; appears in:

`set`

%setindexinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%setindexinfo.role.attrib; appears in:

`setindexinfo`

%setinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%setinfo.role.attrib; appears in:

`setinfo`

%sgmltag.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sgmltag.role.attrib; appears in:

`sgmltag`

%shortaffil.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%shortaffil.role.attrib; appears in:

`shortaffil`

%shortcut.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%shortcut.role.attrib; appears in:

`shortcut`

%sidebar.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sidebar.role.attrib; appears in:

`sidebar`

%sidebarinfo.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%sidebarinfo.role.attrib; appears in:

`sidebarinfo`

%simpara.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%simpara.role.attrib; appears in:

`simpara`

%simplelist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%simplelist.role.attrib; appears in:

`simplelist`

%simplemsgentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%simplemsgentry.role.attrib; appears in:

`simplemsgentry`

%simplesect.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%simplesect.role.attrib; appears in:

`simplesect`

%ssscript.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%ssscript.role.attrib; appears in:

`subscript` `superscript`

%state.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%state.role.attrib; appears in:

`state`

%step.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%step.role.attrib; appears in:

`step`

%street.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%street.role.attrib; appears in:

`street`

%structfield.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%structfield.role.attrib; appears in:

`structfield`

%structname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%structname.role.attrib; appears in:

`structname`

%subject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%subject.role.attrib; appears in:

`subject`

%subjectset.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%subjectset.role.attrib; appears in:

`subjectset`

%subjectterm.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%subjectterm.role.attrib; appears in:

`subjectterm`

%substeps.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%substeps.role.attrib; appears in:

`substeps`

%subtitle.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%subtitle.role.attrib; appears in:

`subtitle`

%surname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%surname.role.attrib; appears in:

`surname`

%symbol.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%symbol.role.attrib; appears in:
`symbol`

%synopfragment.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%synopfragment.role.attrib; appears in:
`synopfragment`

%synopfragmentref.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%synopfragmentref.role.attrib; appears in:
`synopfragmentref`

%synopsis.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%synopsis.role.attrib; appears in:
`synopsis`

%systemitem.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%systemitem.role.attrib; appears in:
`systemitem`

%tables.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*

%term.role.attrib;

Parameter Entity
Name Type Default
role CDATA *None*
%term.role.attrib; appears in:
`term`

%textdata.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%textdata.role.attrib; appears in:

`textdata`

%textobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%textobject.role.attrib; appears in:

`textobject`

%title.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%title.role.attrib; appears in:

`title`

%titleabbrev.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%titleabbrev.role.attrib; appears in:

`titleabbrev`

%toc.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toc.role.attrib; appears in:

`toc`

%tocback.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%tocback.role.attrib; appears in:

`tocback`

%tocchap.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%tocchap.role.attrib; appears in:

`tocchap`

%tocentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%tocentry.role.attrib; appears in:

`tocentry`

%tocfront.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%tocfront.role.attrib; appears in:

`tocfront`

%toclevel1.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toclevel1.role.attrib; appears in:

`toclevel1`

%toclevel2.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toclevel2.role.attrib; appears in:

`toclevel2`

%toclevel3.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toclevel3.role.attrib; appears in:

`toclevel3`

%toclevel4.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toclevel4.role.attrib; appears in:

`toclevel4`

%toclevel5.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%toclevel5.role.attrib; appears in:

`toclevel5`

%tocpart.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%tocpart.role.attrib; appears in:

`tocpart`

%token.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%token.role.attrib; appears in:

`token`

%trademark.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%trademark.role.attrib; appears in:

`trademark`

%type.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%type.role.attrib; appears in:

`type`

%ulink.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%ulink.role.attrib; appears in:

`ulink`

%userinput.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%userinput.role.attrib; appears in:

`userinput`

%varargs.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%varargs.role.attrib; appears in:

`varargs`

%variablelist.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%variablelist.role.attrib; appears in:

`variablelist`

%varlistentry.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%varlistentry.role.attrib; appears in:

`varlistentry`

%varname.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%varname.role.attrib; appears in:

`varname`

%videodata.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%videodata.role.attrib; appears in:

`videodata`

%videoobject.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%videoobject.role.attrib; appears in:

`videoobject`

%void.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%void.role.attrib; appears in:

`void`

%volumenum.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%volumenum.role.attrib; appears in:

`volumenum`

%wordasword.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%wordasword.role.attrib; appears in:

`wordasword`

%xref.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%xref.role.attrib; appears in:

`xref`

%year.role.attrib;

Parameter Entity

Name Type Default

role CDATA *None*

%year.role.attrib; appears in:

`year`

Chapter 8

DocBook Character Entity Reference

■ Added Math Symbols: Arrow Relations Character Entities (%isoamsa;)

Name

%isoamsa; – Added Math Symbols: Arrow Relations Character Entities

Synopsis

The `%isoamsa;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Arrow Relations//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
cularr	21B6	↶	Anticlockwise top semicircle arrow
curarr	21B7	↷	Clockwise top semicircle arrow
dArr	21D3	⇓	Downwards double arrow
darr2	21CA	⇊	Downwards paired arrows
dharl	21C3	⇃	Downwards harpoon with barb leftwards
dharr	21C2	⇂	Downwards harpoon with barb rightwards

EntityName	UnicodeCode point	SampleGlyph	Description
lAarr	21DA		Leftwards triple arrow
Larr	219E		Leftwards two headed arrow
larr2	21C7		Leftwards paired arrows
larrhk	21A9		Leftwards arrow with hook
larrlp	21AB		Leftwards arrow with loop
larrtl	21A2		Leftwards arrow with tail
lhard	21BD		Leftwards harpoon with barb downwards
lharu	21BC		Leftwards harpoon with barb upwards
hArr	21D4		Left right double arrow
harr	2194		Left right arrow
lrarr2	21C6		Leftwards arrow over rightwards arrow
rlarr2	21C4		Rightwards arrow over leftwards arrow
harrw	21AD		Left right wave arrow
rlhar2	21CC		Rightwards harpoon over leftwards harpoon

EntityName	UnicodeCode point	SampleGlyph	Description
lrhar2	21CB		Leftwards harpoon over rightwards harpoon
lsh	21B0		Upwards arrow with tip leftwards
map	21A6		Rightwards arrow from bar
mumap	22B8		Multimap
nearr	2197		North east arrow
nlArr	21CD		Leftwards double arrow with stroke
nlarr	219A		Leftwards arrow with stroke
nhArr	21CE		Left right double arrow with stroke
nharr	21AE		Left right arrow with stroke
nrarr	219B		Rightwards arrow with stroke
nrArr	21CF		Rightwards double arrow with stroke
nwarr	2196		North west arrow
olarr	21BA		Anticlockwise open circle arrow
orarr	21BB		Clockwise open circle arrow

EntityName	UnicodeCode point	SampleGlyph	Description
rAarr	21DB		Rightwards triple arrow
Rarr	21A0		Rightwards two headed arrow
rarr2	21C9		Rightwards paired arrows
rarrhk	21AA		Rightwards arrow with hook
rarrlp	21AC		Rightwards arrow with loop
rarrtl	21A3		Rightwards arrow with tail
rarrw	21DD		Rightwards squiggle arrow
rhard	21C1		Rightwards harpoon with barb downwards
rharu	21C0		Rightwards harpoon with barb upwards
rsh	21B1		Upwards arrow with tip rightwards
drarr	2198		South east arrow
dlarr	2199		South west arrow
uArr	21D1		Upwards double arrow
uarr2	21C8		Upwards paired arrows

EntityName	UnicodeCode point	SampleGlyph	Description
vArr	21D5		Up down double arrow
varr	2195		Up down arrow
uharl	21BF		Upwards harpoon with barb leftwards
uharr	21BE		Upwards harpoon with barb rightwards
xlArr			LEFTWARDS DOUBLE ARROW
xhArr			LEFT RIGHT ARROW
xharr			LEFT RIGHT ARROW
xrArr			RIGHTWARDS DOUBLE ARROW

■ Added Math Symbols: Binary Operators Character Entities (%isoamsb;)

Name

%isoamsb; – Added Math Symbols: Binary Operators Character Entities

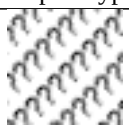
Synopsis

The `%isoamsb;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Binary Operators//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
amalg			N-ARY COPRODUCT

EntityName	UnicodeCode point	SampleGlyph	Description
Barwed	22BC		Nand
barwed	22BC		Nand
Cap	22D2		Double intersection
Cup	22D3		Double union
cuvee	22CE		Curly logical or
cuwed	22CF		Curly logical and
diam	22C4		Diamond operator
divonx	22C7		Division times
intcal	22BA		Intercalate
lthree	22CB		Left semidirect product
ltimes	22C9		Left normal factor semidirect product
minusb	229F		Squared minus
oast	229B		Circled asterisk operator
ocir	229A		Circled ring operator

EntityName	UnicodeCode point	SampleGlyph	Description
odash	229D		Circled dash
odot	2299		Circled dot operator
ominus	2296		Circled minus
oplus	2295		Circled plus
osol	2298		Circled division slash
otimes	2297		Circled times
plusb	229E		Squared plus
plusdo	2214		Dot plus
rthree	22CC		Right semidirect product
rtimes	22CA		Right normal factor semidirect product
sdot	22C5		Dot operator
sdotb	22A1		Squared dot operator
setmn	2216		Set minus
sqcap	2293		Square cap

EntityName	UnicodeCode point	SampleGlyph	Description
sqcup	2294	⊔	Square cup
ssetmn		∖	SET MINUS
sstarf	22C6	⋆	Star operator
timesb	22A0	⊠	Squared times
top	22A4	⊤	Down tack
uplus	228E	⊎	Multiset union
wreath	2240	≀	Wreath product
xcirc	25EF	◯	Large circle
xdtri	25BD	▽	White down-pointing triangle
xutri	25B3	△	White up-pointing triangle
coprod	2210	∐	N-ary coproduct
prod	220F	∏	N-ary product
sum	2211	∑	N-ary summation

■ Added Math Symbols: Delimiters Character Entities (%isoamsc;)

Name

%isoamsc; – Added Math Symbols: Delimiters Character Entities

Synopsis

The `%isoamsc;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Delimiters//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
rceil	2309		Right ceiling
rfloor	230B		Right floor
rpargt			
urcorn	231D		Top right corner
drcorn	231F		Bottom right corner
lceil	2308		Left ceiling
lfloor	230A		Left floor
ulcorn	231C		Top left corner
dlcorn	231E		Bottom left corner

■ Added Math Symbols: Negated Relations Character Entities (%isoamsn;)

Name

%isoamsn; – Added Math Symbols: Negated Relations Character Entities

Synopsis

The `%isoamsn;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Negated Relations//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
gnap			
gne	2269		Greater-than but not equal to
gnE	2269		Greater-than but not equal to
gnsim	22E7		Greater-than but not equivalent to
gvnE			GREATER-THAN BUT NOT EQUAL TO
lnap			
lnE	2268		Less-than but not equal to
lne	2268		Less-than but not equal to
lnsim	22E6		Less-than but not equivalent to
lvnE			LESS-THAN BUT NOT EQUAL TO

EntityName	UnicodeCode point	SampleGlyph	Description
nap	2249	≉	Not almost equal to
ncong	2247	≇	Neither approximately nor actually equal to
nequiv	2262	≢	Not identical to
ngE	2271	≱	Neither greater-than nor equal to
nge			NEITHER GREATER-THAN NOR EQUAL TO
nges	2271	≱	Neither greater-than nor equal to
ngt	226F	≯	Not greater-than
nle			NEITHER LESS-THAN NOR EQUAL TO
nlE	2270	≰	Neither less-than nor equal to
nles	2270	≰	Neither less-than nor equal to
nlt	226E	≮	Not less-than
nltri	22EA	⋪	Not normal subgroup of
nltrie	22EC	⋬	Not normal subgroup of or equal to
nmid	2224	∤	Does not divide

EntityName	UnicodeCode point	SampleGlyph	Description
npar	2226		Not parallel to
npr	2280		Does not precede
npre			DOES NOT PRECEDE OR EQUAL
nrtri	22EB		Does not contain as normal subgroup of
nrtrie	22ED		Does not contain as normal subgroup or equal
nsc	2281		Does not succeed
nsce			DOES NOT SUCCEED OR EQUAL
nsim	2241		Not tilde
nsime	2244		Not asymptotically equal to
nsmid			
nspar			NOT PARALLEL TO
nsub	2284		Not a subset of
nsube	2288		Neither a subset of nor equal to
nsubE	2288		Neither a subset of nor equal to

EntityName	UnicodeCode point	SampleGlyph	Description
nsup	2285		Not a superset of
nsupE	2289		Neither a superset of nor equal to
nsupe	2289		Neither a superset of nor equal to
nvdash	22AC		Does not prove
nvDash	22AD		Not true
nVDash	22AF		Negated double vertical bar double right turnstile
nVdash	22AE		Does not force
prnap	22E8		Precedes but not equivalent to
prnE			
prnsim	22E8		Precedes but not equivalent to
scnap	22E9		Succeed but not equivalent to
scnE			
scnsim	22E9		Succeed but not equivalent to
subne	228A		Subset of or not equal to

EntityName	UnicodeCode point	SampleGlyph	Description
subnE	228A	⊊	Subset of or not equal to
supne	228B	⊋	Superset of or not equal to
supnE	228B	⊋	Superset of or not equal to
vsubnE		⫋	
vsubne		⊊	SUBSET OF WITH NOT EQUAL TO
vsupne		⊋	SUPERSET OF WITH NOT EQUAL TO
vsupnE		⫌	SUPERSET OF WITH NOT EQUAL TO

■ Added Math Symbols: Ordinary Character Entities (%isoamso;)

Name

%isoamso; – Added Math Symbols: Ordinary Character Entities

Synopsis

The `%isoamso;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Ordinary//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
ang	2220	∠	Angle
angmsd	2221	∡	Measured angle

EntityName	UnicodeCode point	SampleGlyph	Description
beth	2136		Bet symbol
bprime	2035		Reversed prime
comp	2201		Complement
daleth	2138		Dalet symbol
ell	2113		Script small l
empty			
gimel	2137		Gimel symbol
image	2111		Fraktur letter capital i
inodot	0131		Latin small letter dotless i
nexist	2204		There does not exist
oS	24C8		Circled latin capital letter S
planck	0127		Latin small letter h with stroke
real	211C		Fraktur letter capital r
sbsol			SMALL REVERSE SOLIDUS

EntityName	UnicodeCode point	SampleGlyph	Description
vprime	2032	′	Prime
weierp	2118	℘	Script capital p

■ Added Math Symbols: Relations Character Entities (%isoamsr;)

Name

%isoamsr; – Added Math Symbols: Relations Character Entities

Synopsis

The `%isoamsr;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Math Symbols: Relations//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
ape	224A	≊	Almost equal or equal to
asymp	224D	≍	Equivalent to
bcong	224C	≌	All equal to
bepsi		∍	SMALL CONTAINS AS MEMBER
bowtie	22C8	⋈	Bowtie
bsim	223D	∽	Reversed tilde
bsime	22CD	⋍	Reversed tilde equals

EntityName	UnicodeCode point	SampleGlyph	Description
bump	224E		Geometrically equivalent to
bumpe	224F		Difference between
cire	2257		Ring equal to
colone	2254		Colon equals
cuepr	22DE		Equal to or precedes
cuesc	22DF		Equal to or succeeds
cupre	227C		Precedes or equal to
dashv	22A3		Left tack
ecir	2256		Ring in equal to
ecolon	2255		Equals colon
eDot	2251		Geometrically equal to
esdot	2250		Approaches the limit
efDot	2252		Approximately equal to or the image of
egs	22DD		Equal to or greater-than

EntityName	UnicodeCode point	SampleGlyph	Description
els	22DC		Equal to or less-than
erDot	2253		Image of or approximately equal to
fork	22D4		Pitchfork
frown	2322		Frown
gap	2273		Greater-than or equivalent to
gsdot	22D7		Greater-than with dot
gE	2267		Greater-than over equal to
gel	22DB		Greater-than equal to or less-than
gEl	22DB		Greater-than equal to or less-than
ges			GREATER-THAN OR EQUAL TO
Gg	22D9		Very much greater-than
gl	2277		Greater-than or less-than
gsim	2273		Greater-than or equivalent to
Gt	226B		Much greater-than

EntityName	UnicodeCode point	SampleGlyph	Description
lap	2272		Less-than or equivalent to
ldot	22D6		Less-than with dot
lE	2266		Less-than over equal to
lEg	22DA		Less-than equal to or greater-than
leg	22DA		Less-than equal to or greater-than
les			LESS-THAN OR EQUAL TO
lg	2276		Less-than or greater-than
Ll	22D8		Very much less-than
lsim	2272		Less-than or equivalent to
Lt	226A		Much less-than
ltrie	22B4		Normal subgroup of or equal to
mid	2223		Divides
models	22A7		Models
pr	227A		Precedes

EntityName	UnicodeCode point	SampleGlyph	Description
prap	227E		Precedes or equivalent to
pre			
prsim	227E		Precedes or equivalent to
rtrie	22B5		Contains as normal subgroup or equal to
samalg	2210		N-ary coproduct
sc	227B		Succeeds
scap	227F		Succeeds or equivalent to
sccue	227D		Succeeds or equal to
sce	227D		Succeeds or equal to
scsim	227F		Succeeds or equivalent to
sfrown			FROWN
smid			
smile	2323		Smile
spar			PARALLEL TO

EntityName	UnicodeCode point	SampleGlyph	Description
sqsub	228F		Square image of
sqsube	2291		Square image of or equal to
sqsup	2290		Square original of
sqsupe	2292		Square original of or equal to
ssmile			SMILE
Sub	22D0		Double subset
subE	2286		Subset of or equal to
Sup	22D1		Double superset
supE	2287		Superset of or equal to
thkap			ALMOST EQUAL TO
thksim			TILDE OPERATOR
trie	225C		Delta equal to
twixt	226C		Between
vdash	22A2		Right tack

EntityName	UnicodeCode point	SampleGlyph	Description
Vdash	22A9	⊩	Forces
vDash	22A8	⊨	True
veebar	22BB	⊻	Xor
vltri	22B2	⊲	Normal subgroup of
vprop	221D	∝	Proportional to
vrtri	22B3	⊳	Contains as normal subgroup
Vvdash	22AA	⊪	Triple vertical bar right turnstile

■ Box and Line Drawing Character Entities (%isobox;)

Name

%isobox; – Box and Line Drawing Character Entities

Synopsis

The `%isobox;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Box and Line Drawing//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
boxh	2500	─	Box drawings light horizontal
boxv	2502	│	Box drawings light vertical

EntityName	UnicodeCode point	SampleGlyph	Description
boxur	2514		Box drawings light up and right
boxul	2518		Box drawings light up and left
boxdl	2510		Box drawings light down and left
boxdr	250C		Box drawings light down and right
boxvr	251C		Box drawings light vertical and right
boxhu	2534		Box drawings light up and horizontal
boxvl	2524		Box drawings light vertical and left
boxhd	252C		Box drawings light down and horizontal
boxvh	253C		Box drawings light vertical and horizontal
boxvR	255E		Box drawings vertical single and right double
boxhU	2568		Box drawings up double and horizontal single
boxvL	2561		Box drawings vertical single and left double
boxhD	2565		Box drawings down double and horizontal single
boxvH	256A		Box drawings vertical single and horizontal double

EntityName	UnicodeCode point	SampleGlyph	Description
boxH	2550		Box drawings double horizontal
boxV	2551		Box drawings double vertical
boxUR	255A		Box drawings double up and right
boxUL	255D		Box drawings double up and left
boxDL	2557		Box drawings double down and left
boxDR	2554		Box drawings double down and right
boxVR	2560		Box drawings double vertical and right
boxHU	2569		Box drawings double up and horizontal
boxVL	2563		Box drawings double vertical and left
boxHD	2566		Box drawings double down and horizontal
boxVH	256C		Box drawings double vertical and horizontal
boxVr	255F		Box drawings vertical double and right single
boxHu	2567		Box drawings up single and horizontal double
boxVl	2562		Box drawings vertical double and left single

EntityName	UnicodeCode point	SampleGlyph	Description
boxHd	2564		Box drawings down single and horizontal double
boxVh	256B		Box drawings vertical double and horizontal single
boxuR	2558		Box drawings up single and right double
boxUl	255C		Box drawings up double and left single
boxdL	2555		Box drawings down single and left double
boxDr	2553		Box drawings down double and right single
boxUr	2559		Box drawings up double and right single
boxuL	255B		Box drawings up single and left double
boxDl	2556		Box drawings down double and left single
boxdR	2552		Box drawings down single and right double

■ Russian Cyrillic Character Entities (%isocyr1;)

Name

%isocyr1; – Russian Cyrillic Character Entities

Synopsis

The `%isocyr1;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Russian Cyrillic//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
acy	0430	а	Cyrillic small letter a
Acy	0410	А	Cyrillic capital letter A
bcy	0431	б	Cyrillic small letter be
Bcy	0411	Б	Cyrillic capital letter BE
vcy	0432	в	Cyrillic small letter ve
Vcy	0412	В	Cyrillic capital letter VE
gcy	0433	г	Cyrillic small letter ghe
Gcy	0413	Г	Cyrillic capital letter GHE
dcy	0434	д	Cyrillic small letter de
Dcy	0414	Д	Cyrillic capital letter DE
iecy	0435	е	Cyrillic small letter ie
IEcy	0415	Е	Cyrillic capital letter IE
iocy	0451	ё	Cyrillic small letter io
IOcy	0401	Ё	Cyrillic capital letter IO

EntityName	UnicodeCode point	SampleGlyph	Description
zhcy	0436	ж	Cyrillic small letter zhe
ZHcy	0416	Ж	Cyrillic capital letter ZHE
zcy	0437	з	Cyrillic small letter ze
Zcy	0417	З	Cyrillic capital letter ZE
icy	0438	и	Cyrillic small letter i
Icy	0418	И	Cyrillic capital letter I
jcy	0439	й	Cyrillic small letter short i
Jcy	0419	Й	Cyrillic capital letter SHORT i
kcy	043A	к	Cyrillic small letter ka
Kcy	041A	К	Cyrillic capital letter KA
lcy	043B	л	Cyrillic small letter el
Lcy	041B	Л	Cyrillic capital letter EL
mcy	043C	м	Cyrillic small letter em
Mcy	041C	М	Cyrillic capital letter EM

EntityName	UnicodeCode point	SampleGlyph	Description
ncy	043D		Cyrillic small letter en
Ncy	041D		Cyrillic capital letter EN
ocy	043E		Cyrillic small letter o
Ocy	041E		Cyrillic capital letter O
pcy	043F		Cyrillic small letter pe
Pcy	041F		Cyrillic capital letter PE
rcy	0440		Cyrillic small letter er
Rcy	0420		Cyrillic capital letter ER
scy	0441		Cyrillic small letter es
Scy	0421		Cyrillic capital letter ES
tcy	0442		Cyrillic small letter te
Tcy	0422		Cyrillic capital letter TE
ucy	0443		Cyrillic small letter u
Ucy	0423		Cyrillic capital letter U

EntityName	UnicodeCode point	SampleGlyph	Description
fcy	0444	ф	Cyrillic small letter ef
Fcy	0424	Ф	Cyrillic capital letter EF
khcy	0445	х	Cyrillic small letter ha
KHcy	0425	Х	Cyrillic capital letter HA
tscy	0446	ц	Cyrillic small letter tse
TScy	0426	Ц	Cyrillic capital letter TSE
chcy	0447	ч	Cyrillic small letter che
CHcy	0427	Ч	Cyrillic capital letter CHE
shcy	0448	ш	Cyrillic small letter sha
SHcy	0428	Ш	Cyrillic capital letter SHA
shchcy	0449	щ	Cyrillic small letter shcha
SHCHcy	0429	Щ	Cyrillic capital letter SHCHA
hardcy	044A	ъ	Cyrillic small letter hard sign
HARDcy	042A	Ъ	Cyrillic capital letter HARD sign

EntityName	UnicodeCode point	SampleGlyph	Description
ycy	044B	ы	Cyrillic small letter yeru
Ycy	042B	Ы	Cyrillic capital letter YERU
softcy	044C	ь	Cyrillic small letter soft sign
SOFTcy	042C	Ь	Cyrillic capital letter SOFT sign
ecy	044D	э	Cyrillic small letter e
Ecy	042D	Э	Cyrillic capital letter E
yucy	044E	ю	Cyrillic small letter yu
YUcy	042E	Ю	Cyrillic capital letter YU
yacy	044F	я	Cyrillic small letter ya
YAcy	042F	Я	Cyrillic capital letter YA
numero	2116	№	Numero sign

■ Non-Russian Cyrillic Character Entities (%isocyr2;)

Name

%isocyr2; – Non-Russian Cyrillic Character Entities

Synopsis

The `%isocyr2;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Non-Russian Cyrillic//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
djcy	0452		Cyrillic small letter dje
DJcy	0402		Cyrillic capital letter DJE
gjcy	0453		Cyrillic small letter gje
GJcy	0403		Cyrillic capital letter GJE
jukcy	0454		Cyrillic small letter ukrainian ie
Jukcy	0404		Cyrillic capital letter UKRAINIAN ie
dscy	0455		Cyrillic small letter dze
DScy	0405		Cyrillic capital letter DZE
iukcy	0456		Cyrillic small letter byelorussian-ukrainian i
Iukcy	0406		Cyrillic capital letter BYELORUSSIAN-UKRAINIAN i
yicy	0457		Cyrillic small letter yi
YIcy	0407		Cyrillic capital letter YI
jsercy	0458		Cyrillic small letter je

EntityName	UnicodeCode point	SampleGlyph	Description
Jsercy	0408		Cyrillic capital letter JE
ljcy	0459		Cyrillic small letter lje
LJcy	0409		Cyrillic capital letter LJE
njcy	045A		Cyrillic small letter nje
NJcy	040A		Cyrillic capital letter NJE
tshcy	045B		Cyrillic small letter tshe
TSHcy	040B		Cyrillic capital letter TSHE
kjcy	045C		Cyrillic small letter kje
KJcy	040C		Cyrillic capital letter KJE
ubrcy	045E		Cyrillic small letter short u
Ubrcy	040E		Cyrillic capital letter SHORT u
dzcy	045F		Cyrillic small letter dzhe
DZcy	040F		Cyrillic capital letter DZHE

■ Diacritical Marks Character Entities (%isodia;)

Name

%isodia; – Diacritical Marks Character Entities

Synopsis

The `%isodia;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Diacritical Marks//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
acute	00B4		Acute accent
breve	02D8		Breve
caron	02C7		Caron
cedil	00B8		Cedilla
circ	005E		Circumflex accent
dblac	02DD		Double acute accent
die	00A8		Diaeresis
dot	02D9		Dot above
grave	0060		Grave accent
macr	00AF		Macron

EntityName	UnicodeCode point	SampleGlyph	Description
ogon	02DB		Ogonek
ring	02DA		Ring above
tilde	02DC		Small tilde
uml	00A8		Diaeresis

■ Greek Letters Character Entities (%isogrk1;)

Name

%isogrk1; – Greek Letters Character Entities

Synopsis

The `%isogrk1;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Greek Letters//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
agr	03B1	α	Greek small letter alpha
Agr	0391	Α	Greek capital letter ALPHA
bgr	03B2	β	Greek small letter beta
Bgr	0392	Β	Greek capital letter BETA
ggr	03B3	γ	Greek small letter gamma

EntityName	UnicodeCode point	SampleGlyph	Description
Ggr	0393	Γ	Greek capital letter GAMMA
dgr	03B4	δ	Greek small letter delta
Dgr	0394	Δ	Greek capital letter DELTA
egr	03B5	ε	Greek small letter epsilon
Egr	0395	Ε	Greek capital letter EPSILON
zgr	03B6	ζ	Greek small letter zeta
Zgr	0396	Ζ	Greek capital letter ZETA
eegr	03B7	η	Greek small letter eta
EEgr	0397	Η	Greek capital letter ETA
thgr	03B8	θ	Greek small letter theta
THgr	0398	Θ	Greek capital letter THETA
igr	03B9	ι	Greek small letter iota
Igr	0399	Ι	Greek capital letter IOTA
kgr	03BA	κ	Greek small letter kappa

EntityName	UnicodeCode point	SampleGlyph	Description
Kgr	039A	Κ	Greek capital letter KAPPA
lgr	03BB	λ	Greek small letter lamda
Lgr	039B	Λ	Greek capital letter LAMDA
mgr	03BC	μ	Greek small letter mu
Mgr	039C	Μ	Greek capital letter MU
ngr	03BD	ν	Greek small letter nu
Ngr	039D	Ν	Greek capital letter NU
xgr	03BE	ξ	Greek small letter xi
Xgr	039E	Ξ	Greek capital letter XI
ogr	03BF	ο	Greek small letter omicron
Ogr	039F	Ο	Greek capital letter OMICRON
pgr	03C0	π	Greek small letter pi
Pgr	03A0	Π	Greek capital letter PI
rgr	03C1	ρ	Greek small letter rho

EntityName	UnicodeCode point	SampleGlyph	Description
Rgr	03A1	Ρ	Greek capital letter RHO
sgr	03C3	σ	Greek small letter sigma
Sgr	03A3	Σ	Greek capital letter SIGMA
sfgr	03C2	ς	Greek small letter final sigma
tgr	03C4	τ	Greek small letter tau
Tgr	03A4	Τ	Greek capital letter TAU
ugr	03C5	υ	Greek small letter upsilon
Ugr	03A5	Υ	Greek capital letter UPSILON
phgr	03C6	ϕ	Greek small letter phi
PHgr	03A6	Φ	Greek capital letter PHI
khgr	03C7	χ	Greek small letter chi
KHgr	03A7	Χ	Greek capital letter CHI
psgr	03C8	ψ	Greek small letter psi
PSgr	03A8	Ψ	Greek capital letter PSI

EntityName	UnicodeCode point	SampleGlyph	Description
ohgr	03C9	ω	Greek small letter omega
OHgr	03A9	Ω	Greek capital letter OMEGA

■ Monotoniko Greek Character Entities (%isogrk2;)

Name

%isogrk2; – Monotoniko Greek Character Entities

Synopsis

The `%isogrk2;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Monotoniko Greek//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
aacgr	03AC	ά	Greek small letter alpha with tonos
Aacgr	0386	Ά	Greek capital letter ALPHA with tonos
eacgr	03AD	έ	Greek small letter epsilon with tonos
Eacgr	0388	Έ	Greek capital letter EPSILON with tonos
eeacgr	03AE	ή	Greek small letter eta with tonos
EEacgr	0389	Ή	Greek capital letter ETA with tonos
idigr	03CA	ϊ	Greek small letter iota with dialytika

EntityName	UnicodeCode point	SampleGlyph	Description
Idigr	03AA	Ϊ	Greek capital letter IOTA with dialytika
iacgr	03AF	ί	Greek small letter iota with tonos
Iacgr	038A	Ί	Greek capital letter IOTA with tonos
idiagr	0390	ΐ	Greek small letter iota with dialytika and tonos
oacgr	03CC	ό	Greek small letter omicron with tonos
Oacgr	038C	Ό	Greek capital letter OMICRON with tonos
udigr	03CB	ϋ	Greek small letter upsilon with dialytika
Udigr	03AB	Ϋ	Greek capital letter UPSILON with dialytika
uacgr	03CD	ύ	Greek small letter upsilon with tonos
Uacgr	038E	Ύ	Greek capital letter UPSILON with tonos
udiagr	03B0	ΰ	Greek small letter upsilon with tonos and dialytika
ohacgr	03CE	ώ	Greek small letter omega with tonos
OHacgr	038F	Ώ	Greek capital letter OMEGA with tonos

■ Greek Symbols Character Entities (%isogrk3;)

Name

%isogrk3; – Greek Symbols Character Entities

Synopsis

The `%isogrk3;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Greek Symbols//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
alpha	03B1	α	Greek small letter alpha
beta	03B2	β	Greek small letter beta
gamma	03B3	γ	Greek small letter gamma
Gamma	0393	Γ	Greek capital letter GAMMA
gammad	03DC	Ϝ	Greek letter digamma
delta	03B4	δ	Greek small letter delta
Delta	0394	Δ	Greek capital letter DELTA
epsi	03B5	ε	Greek small letter epsilon
epsiv	025B	ɛ	Latin small letter open e
epsis	03B5	ε	Greek small letter epsilon

EntityName	UnicodeCode point	SampleGlyph	Description
zeta	03B6	ζ	Greek small letter zeta
eta	03B7	η	Greek small letter eta
thetas	03B8	θ	Greek small letter theta
Theta	0398	Θ	Greek capital letter THETA
thetav	03D1	ϑ	Greek theta symbol
iota	03B9	ι	Greek small letter iota
kappa	03BA	κ	Greek small letter kappa
kappav	03F0	ϰ	Greek kappa symbol
lambda	03BB	λ	Greek small letter lamda
Lambda	039B	Λ	Greek capital letter LAMDA
mu	03BC	μ	Greek small letter mu
nu	03BD	ν	Greek small letter nu
xi	03BE	ξ	Greek small letter xi
Xi	039E	Ξ	Greek capital letter XI

EntityName	UnicodeCode point	SampleGlyph	Description
pi	03C0	π	Greek small letter pi
piv	03D6	ϖ	Greek omega symbol
Pi	03A0	Π	Greek capital letter PI
rho	03C1	ρ	Greek small letter rho
rhov	03F1	ϱ	Greek rho symbol
sigma	03C3	σ	Greek small letter sigma
Sigma	03A3	Σ	Greek capital letter SIGMA
sigmav	03C2	ς	Greek small letter final sigma
tau	03C4	τ	Greek small letter tau
upsi	03C5	υ	Greek small letter upsilon
Upsi	03D2	ϒ	Greek upsilon with hook symbol
phis	03C6	ϕ	Greek small letter phi
Phi	03A6	Φ	Greek capital letter PHI
phiv	03D5	φ	Greek phi symbol

EntityName	UnicodeCode point	SampleGlyph	Description
chi	03C7	χ	Greek small letter chi
psi	03C8	ψ	Greek small letter psi
Psi	03A8	Ψ	Greek capital letter PSI
omega	03C9	ω	Greek small letter omega
Omega	03A9	Ω	Greek capital letter OMEGA

■ Alternative Greek Symbols Character Entities (%isogrk4;)

Name

%isogrk4; – Alternative Greek Symbols Character Entities

Synopsis

The `%isogrk4;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Alternative Greek Symbols//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
b.alpha	03B1	α	Greek small letter alpha
b.beta	03B2	β	Greek small letter beta
b.gamma	03B3	γ	Greek small letter gamma
b.Gamma	0393	Γ	Greek capital letter GAMMA

EntityName	UnicodeCode point	SampleGlyph	Description
b.gammad	03DC		Greek letter digamma
b.delta	03B4		Greek small letter delta
b.Delta	0394		Greek capital letter DELTA
b.epsi	03B5		Greek small letter epsilon
b.epsiv	025B		Latin small letter open e
b.epsis	03B5		Greek small letter epsilon
b.zeta	03B6		Greek small letter zeta
b.eta	03B7		Greek small letter eta
b.thetas	03B8		Greek small letter theta
b.Theta	0398		Greek capital letter THETA
b.thetav	03D1		Greek theta symbol
b.iota	03B9		Greek small letter iota
b.kappa	03BA		Greek small letter kappa
b.kappav	03F0		Greek kappa symbol

EntityName	UnicodeCode point	SampleGlyph	Description
b.lambda	03BB	λ	Greek small letter lamda
b.Lambda	039B	Λ	Greek capital letter LAMDA
b.mu	03BC	μ	Greek small letter mu
b.nu	03BD	ν	Greek small letter nu
b.xi	03BE	ξ	Greek small letter xi
b.Xi	039E	Ξ	Greek capital letter XI
b.pi	03C0	π	Greek small letter pi
b.Pi	03A0	Π	Greek capital letter PI
b.piv	03D6	ϖ	Greek omega symbol
b.rho	03C1	ρ	Greek small letter rho
b.rhov	03F1	ϱ	Greek rho symbol
b.sigma	03C3	σ	Greek small letter sigma
b.Sigma	03A3	Σ	Greek capital letter SIGMA
b.sigmav	03C2	ς	Greek small letter final sigma

EntityName	UnicodeCode point	SampleGlyph	Description
b.tau	03C4	τ	Greek small letter tau
b.upsi	03C5	υ	Greek small letter upsilon
b.Upsi	03D2	ϒ	Greek upsilon with hook symbol
b.phis	03C6	ϕ	Greek small letter phi
b.Phi	03A6	Φ	Greek capital letter PHI
b.phiv	03D5	φ	Greek phi symbol
b.chi	03C7	χ	Greek small letter chi
b.psi	03C8	ψ	Greek small letter psi
b.Psi	03A8	Ψ	Greek capital letter PSI
b.omega	03C9	ω	Greek small letter omega
b.Omega	03A9	Ω	Greek capital letter OMEGA

■ ISO Latin 1 Character Entities (%isolat1;)

Name

%isolat1; – ISO Latin 1 Character Entities

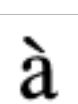
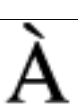
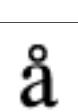
Synopsis

The `%isolat1;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Latin 1//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
aacute	00E1		Latin small letter a with acute
Aacute	00C1		Latin capital letter A with acute
acirc	00E2		Latin small letter a with circumflex
Acirc	00C2		Latin capital letter A with circumflex
agrave	00E0		Latin small letter a with grave
Agrave	00C0		Latin capital letter A with grave
aring	00E5		Latin small letter a with ring above
Aring	00C5		Latin capital letter A with ring above
atilde	00E3		Latin small letter a with tilde
Atilde	00C3		Latin capital letter A with tilde
auml	00E4		Latin small letter a with diaeresis
Auml	00C4		Latin capital letter A with diaeresis
aelig	00E6		Latin small letter ae

EntityName	UnicodeCode point	SampleGlyph	Description
AElig	00C6	Æ	Latin capital letter AE
ccedil	00E7	ç	Latin small letter c with cedilla
Ccedil	00C7	Ç	Latin capital letter C with cedilla
eth	00F0	ð	Latin small letter eth
ETH	00D0	Ð	Latin capital letter ETH
eacute	00E9	é	Latin small letter e with acute
Eacute	00C9	É	Latin capital letter E with acute
ecirc	00EA	ê	Latin small letter e with circumflex
Ecirc	00CA	Ê	Latin capital letter E with circumflex
egrave	00E8	è	Latin small letter e with grave
Egrave	00C8	È	Latin capital letter E with grave
euml	00EB	ë	Latin small letter e with diaeresis
Euml	00CB	Ë	Latin capital letter E with diaeresis
iacute	00ED	í	Latin small letter i with acute

EntityName	UnicodeCode point	SampleGlyph	Description
Iacute	00CD	Í	Latin capital letter I with acute
icirc	00EE	î	Latin small letter i with circumflex
Icirc	00CE	Î	Latin capital letter I with circumflex
igrave	00EC	ì	Latin small letter i with grave
Igrave	00CC	Ì	Latin capital letter I with grave
iuml	00EF	ï	Latin small letter i with diaeresis
Iuml	00CF	Ï	Latin capital letter I with diaeresis
ntilde	00F1	ñ	Latin small letter n with tilde
Ntilde	00D1	Ñ	Latin capital letter N with tilde
oacute	00F3	ó	Latin small letter o with acute
Oacute	00D3	Ó	Latin capital letter O with acute
ocirc	00F4	ô	Latin small letter o with circumflex
Ocirc	00D4	Ô	Latin capital letter O with circumflex
ograve	00F2	ò	Latin small letter o with grave

EntityName	UnicodeCode point	SampleGlyph	Description
Ograve	00D2	Ò	Latin capital letter O with grave
oslash	00F8	ø	Latin small letter o with stroke
Oslash	00D8	Ø	Latin capital letter O with stroke
otilde	00F5	õ	Latin small letter o with tilde
Otilde	00D5	Õ	Latin capital letter O with tilde
ouml	00F6	ö	Latin small letter o with diaeresis
Ouml	00D6	Ö	Latin capital letter O with diaeresis
szlig	00DF	ß	Latin small letter sharp s
thorn	00FE	þ	Latin small letter thorn
THORN	00DE	Þ	Latin capital letter THORN
uacute	00FA	ú	Latin small letter u with acute
Uacute	00DA	Ú	Latin capital letter U with acute
ucirc	00FB	û	Latin small letter u with circumflex
Ucirc	00DB	Û	Latin capital letter U with circumflex

EntityName	UnicodeCode point	SampleGlyph	Description
ugrave	00F9	ù	Latin small letter u with grave
Ugrave	00D9	Ù	Latin capital letter U with grave
uuml	00FC	ü	Latin small letter u with diaeresis
Uuml	00DC	Ü	Latin capital letter U with diaeresis
yacute	00FD	ý	Latin small letter y with acute
Yacute	00DD	Ý	Latin capital letter Y with acute
yuml	00FF	ÿ	Latin small letter y with diaeresis

■ Added Latin 2 Character Entities (%isolat2;)

Name

%isolat2; – Added Latin 2 Character Entities

Synopsis

The `%isolat2;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Added Latin 2//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
abreve	0103	ă	Latin small letter a with breve
Abreve	0102	Ă	Latin capital letter A with breve

EntityName	UnicodeCode point	SampleGlyph	Description
amacr	0101	ā	Latin small letter a with macron
Amacr	0100	Ā	Latin capital letter A with macron
aogon	0105	ą	Latin small letter a with ogonek
Aogon	0104	Ą	Latin capital letter A with ogonek
cacute	0107	ć	Latin small letter c with acute
Cacute	0106	Ć	Latin capital letter C with acute
ccaron	010D	č	Latin small letter c with caron
Ccaron	010C	Č	Latin capital letter C with caron
ccirc	0109	ĉ	Latin small letter c with circumflex
Ccirc	0108	Ĉ	Latin capital letter C with circumflex
cdot	010B	ċ	Latin small letter c with dot above
Cdot	010A	Ċ	Latin capital letter C with dot above
dcaron	010F	ď	Latin small letter d with caron
Dcaron	010E	Ď	Latin capital letter D with caron

EntityName	UnicodeCode point	SampleGlyph	Description
dstrok	0111	đ	Latin small letter d with stroke
Dstrok	0110	Đ	Latin capital letter D with stroke
ecaron	011B	ě	Latin small letter e with caron
Ecaron	011A	Ě	Latin capital letter E with caron
edot	0117	ė	Latin small letter e with dot above
Edot	0116	Ė	Latin capital letter E with dot above
emacr	0113	ē	Latin small letter e with macron
Emacr	0112	Ē	Latin capital letter E with macron
eogon	0119	ę	Latin small letter e with ogonek
Eogon	0118	Ę	Latin capital letter E with ogonek
gacute	01F5	ǵ	Latin small letter g with acute
gbreve	011F	ğ	Latin small letter g with breve
Gbreve	011E	Ğ	Latin capital letter G with breve
Gcedil	0122	Ģ	Latin capital letter G with cedilla

EntityName	UnicodeCode point	SampleGlyph	Description
gcirc	011D		Latin small letter g with circumflex
Gcirc	011C		Latin capital letter G with circumflex
gdot	0121		Latin small letter g with dot above
Gdot	0120		Latin capital letter G with dot above
hcirc	0125		Latin small letter h with circumflex
Hcirc	0124		Latin capital letter H with circumflex
hstrok	0127		Latin small letter h with stroke
Hstrok	0126		Latin capital letter H with stroke
Idot	0130		Latin capital letter I with dot above
Imacr	012A		Latin capital letter I with macron
imacr	012B		Latin small letter i with macron
ijlig	0133		Latin small ligature ij
IJlig	0132		Latin capital ligature ij
inodot	0131		Latin small letter dotless i

EntityName	UnicodeCode point	SampleGlyph	Description
iogon	012F		Latin small letter i with ogonek
Iogon	012E		Latin capital letter I with ogonek
itilde	0129		Latin small letter i with tilde
Itilde	0128		Latin capital letter I with tilde
jcirc	0135		Latin small letter j with circumflex
Jcirc	0134		Latin capital letter J with circumflex
kcedil	0137		Latin small letter k with cedilla
Kcedil	0136		Latin capital letter K with cedilla
kgreen	0138		Latin small letter kra
lacute	013A		Latin small letter l with acute
Lacute	0139		Latin capital letter L with acute
lcaron	013E		Latin small letter l with caron
Lcaron	013D		Latin capital letter L with caron
lcedil	013C		Latin small letter l with cedilla

EntityName	UnicodeCode point	SampleGlyph	Description
Lcedil	013B		Latin capital letter L with cedilla
lmidot	0140		Latin small letter l with middle dot
Lmidot	013F		Latin capital letter L with middle dot
lstrok	0142		Latin small letter l with stroke
Lstrok	0141		Latin capital letter L with stroke
nacute	0144		Latin small letter n with acute
Nacute	0143		Latin capital letter N with acute
eng	014B		Latin small letter eng
ENG	014A		Latin capital letter ENG
napos	0149		Latin small letter n preceded by apostrophe
ncaron	0148		Latin small letter n with caron
Ncaron	0147		Latin capital letter N with caron
ncedil	0146		Latin small letter n with cedilla
Ncedil	0145		Latin capital letter N with cedilla

EntityName	UnicodeCode point	SampleGlyph	Description
odblac	0151	ő	Latin small letter o with double acute
Odblac	0150	Ő	Latin capital letter O with double acute
Omacr	014C	Ō	Latin capital letter O with macron
omacr	014D	ō	Latin small letter o with macron
oelig	0153	œ	Latin small ligature oe
OElig	0152	Œ	Latin capital ligature oe
racute	0155	ŕ	Latin small letter r with acute
Racute	0154	Ŕ	Latin capital letter R with acute
rcaron	0159	ř	Latin small letter r with caron
Rcaron	0158	Ř	Latin capital letter R with caron
rcedil	0157	ŗ	Latin small letter r with cedilla
Rcedil	0156	Ŗ	Latin capital letter R with cedilla
sacute	015B	ś	Latin small letter s with acute
Sacute	015A	Ś	Latin capital letter S with acute

EntityName	UnicodeCode point	SampleGlyph	Description
scaron	0161		Latin small letter s with caron
Scaron	0160		Latin capital letter S with caron
scedil	015F		Latin small letter s with cedilla
Scedil	015E		Latin capital letter S with cedilla
scirc	015D		Latin small letter s with circumflex
Scirc	015C		Latin capital letter S with circumflex
tcaron	0165		Latin small letter t with caron
Tcaron	0164		Latin capital letter T with caron
tcedil	0163		Latin small letter t with cedilla
Tcedil	0162		Latin capital letter T with cedilla
tstrok	0167		Latin small letter t with stroke
Tstrok	0166		Latin capital letter T with stroke
ubreve	016D		Latin small letter u with breve
Ubreve	016C		Latin capital letter U with breve

EntityName	UnicodeCode point	SampleGlyph	Description
udblac	0171	ű	Latin small letter u with double acute
Udblac	0170	Ű	Latin capital letter U with double acute
umacr	016B	ū	Latin small letter u with macron
Umacr	016A	Ū	Latin capital letter U with macron
uogon	0173	ų	Latin small letter u with ogonek
Uogon	0172	Ų	Latin capital letter U with ogonek
uring	016F	ů	Latin small letter u with ring above
Uring	016E	Ů	Latin capital letter U with ring above
utilde	0169	ũ	Latin small letter u with tilde
Utilde	0168	Ũ	Latin capital letter U with tilde
wcirc	0175	ŵ	Latin small letter w with circumflex
Wcirc	0174	Ŵ	Latin capital letter W with circumflex
ycirc	0177	ŷ	Latin small letter y with circumflex
Ycirc	0176	Ŷ	Latin capital letter Y with circumflex

EntityName	UnicodeCode point	SampleGlyph	Description
Yuml	0178	Ÿ	Latin capital letter Y with diaeresis
zacute	017A	ź	Latin small letter z with acute
Zacute	0179	Ź	Latin capital letter Z with acute
zcaron	017E	ž	Latin small letter z with caron
Zcaron	017D	Ž	Latin capital letter Z with caron
zdot	017C	ż	Latin small letter z with dot above
Zdot	017B	Ż	Latin capital letter Z with dot above

■ Numeric and Special Graphic Character Entities (%isonum;)

Name

%isonum; – Numeric and Special Graphic Character Entities

Synopsis

The `%isonum;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES Numeric and Special Graphic//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
half	00BD	½	Vulgar fraction one half
frac12	00BD	½	Vulgar fraction one half

EntityName	UnicodeCode point	SampleGlyph	Description
frac14	00BC	¼	Vulgar fraction one quarter
frac34	00BE	¾	Vulgar fraction three quarters
frac18	215B	⅛	Vulgar fraction one eighth
frac38	215C	⅜	Vulgar fraction three eighths
frac58	215D	⅝	Vulgar fraction five eighths
frac78	215E	⅞	Vulgar fraction seven eighths
sup1	00B9	¹	Superscript one
sup2	00B2	²	Superscript two
sup3	00B3	³	Superscript three
plus	002B	+	Plus sign
plusmn	00B1	±	Plus-minus sign
lt	003C	<	Less-than sign
equals	003D	=	Equals sign
gt	003E	>	Greater-than sign

EntityName	UnicodeCode point	SampleGlyph	Description
divide	00F7	÷	Division sign
times	00D7	×	Multiplication sign
curren	00A4	¤	Currency sign
pound	00A3	£	Pound sign
dollar	0024	$	Dollar sign
cent	00A2	¢	Cent sign
yen	00A5	¥	Yen sign
num	0023	#	Number sign
percnt	0025	%	Percent sign
amp	0026	&	Ampersand
ast	002A	*	Asterisk
commat	0040	@	Commercial at
lsqb	005B	[	Left square bracket
bsol	005C	\	Reverse solidus

EntityName	UnicodeCode point	SampleGlyph	Description
rsqb	005D	]	Right square bracket
lcub	007B	{	Left curly bracket
horbar	2015	―	Horizontal bar
verbar	007C	|	Vertical line
rcub	007D	}	Right curly bracket
micro	00B5	µ	Micro sign
ohm	2126	Ω	Ohm sign
deg	00B0	°	Degree sign
ordm	00BA	º	Masculine ordinal indicator
ordf	00AA	ª	Feminine ordinal indicator
sect	00A7	§	Section sign
para	00B6	¶	Pilcrow sign
middot	00B7	·	Middle dot
larr	2190	←	Leftwards arrow

EntityName	UnicodeCode point	SampleGlyph	Description
rarr	2192	→	Rightwards arrow
uarr	2191	↑	Upwards arrow
darr	2193	↓	Downwards arrow
copy	00A9	©	Copyright sign
reg	00AE	®	Registered sign
trade	2122	TM	Trade mark sign
brvbar	00A6	¦	Broken bar
not	00AC	¬	Not sign
sung			Eighth note
excl	0021	!	Exclamation mark
iexcl	00A1	¡	Inverted exclamation mark
quot	0022	"	Quotation mark
apos	0027	'	Apostrophe
lpar	0028	(	Left parenthesis

EntityName	UnicodeCode point	SampleGlyph	Description
rpar	0029	)	Right parenthesis
comma	002C	,	Comma
lowbar	005F	_	Low line
hyphen	002D	-	Hyphen
period	002E	.	Period
sol	002F	/	Solidus
colon	003A	:	Colon
semi	003B	;	Semicolon
quest	003F	?	Question mark
iquest	00BF	¿	Inverted question mark
laquo	00AB	«	Left-pointing double angle quotation mark
raquo	00BB	»	Right-pointing double angle quotation mark
lsquo	2018	‘	Left single quotation mark
rsquo	2019	’	Right single quotation mark

EntityName	UnicodeCode point	SampleGlyph	Description
ldquo	201C	“	Left double quotation mark
rdquo	201D	”	Right double quotation mark
nbsp	00A0		No-break space
shy	00AD	‐	Soft hyphen

■ Publishing Character Entities (%isopub;)

Name

%isopub; – Publishing Character Entities

Synopsis

The `%isopub;` parameter entity includes the ISO character entities with the public identifier:

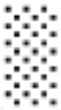
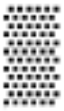
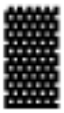
```
ISO 8879:1986//ENTITIES Publishing//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
emsp	2003		Em space
ensp	2002		En space
emsp13	2004		Three-per-em space
emsp14	2005		Four-per-em space
numsp	2007		Figure space

EntityName	UnicodeCode point	SampleGlyph	Description
puncsp	2008		Punctuation space
thinsp	2009		Thin space
hairsp	200A		Hair space
mdash	2014	—	Em dash
ndash	2013	–	En dash
dash	2010	‐	Dash
blank	2423	␣	Open box
hellip	2026	…	Horizontal ellipsis
nldr	2025	‥	Two dot leader
frac13	2153	⅓	Vulgar fraction one third
frac23	2154	⅔	Vulgar fraction two thirds
frac15	2155	⅕	Vulgar fraction one fifth
frac25	2156	⅖	Vulgar fraction two fifths
frac35	2157	⅗	Vulgar fraction three fifths

EntityName	UnicodeCode point	SampleGlyph	Description
frac45	2158		Vulgar fraction four fifths
frac16	2159		Vulgar fraction one sixth
frac56	215A		Vulgar fraction five sixths
incare	2105		Care of
block	2588		Full block
uhblk	2580		Upper half block
lhblk	2584		Lower half block
blk14	2591		Light shade
blk12	2592		Medium shade
blk34	2593		Dark shade
marker	25AE		Black vertical rectangle
cir	25CB		White circle
squ	25A1		White square
rect	25AD		White rectangle

EntityName	UnicodeCode point	SampleGlyph	Description
utri	25B5		White up-pointing small triangle
dtri	25BF		White down-pointing small triangle
star			STAR OPERATOR
bull	2022		Bullet
squf	25AA		Black small square
utrif	25B4		Black up-pointing small triangle
dtrif	25BE		Black down-pointing small triangle
ltrif	25C2		Black left-pointing small triangle
rtrif	25B8		Black right-pointing small triangle
clubs	2663		Black club suit
diams	2666		Black diamond suit
hearts	2661		White heart suit
spades	2660		Black spade suit
malt	2720		Maltese cross

EntityName	UnicodeCode point	SampleGlyph	Description
dagger	2020	†	Dagger
Dagger	2021	‡	Double dagger
check	2713	✓	Check mark
cross	2717	✗	Ballot x
sharp	266F	♯	Music sharp sign
flat	266D	♭	Music flat sign
male	2642	♂	Male sign
female	2640	♀	Female sign
phone	260E	☎	Black telephone
telrec	2315	⌕	Telephone recorder
copysr	2117	℗	Sound recording copyright
caret	2041	⁁	Caret insertion point
lsquor	201A	‚	Single low-nine quotation mark
ldquor	201E	„	Double low-nine quotation mark

EntityName	UnicodeCode point	SampleGlyph	Description
fflig	FB00	ﬀ	Latin small ligature ff
filig	007F	ﬁ	Latin small ligature fi
ffilig	FB03	ﬃ	Latin small ligature ffi
ffllig	FB04	ﬄ	Latin small ligature ffl
fllig	0090	ﬂ	Latin small ligature fl
mldr	2026	…	Horizontal ellipsis
rdquor	201D	”	Right double quotation mark
rsquor	2019	’	Right single quotation mark
vellip	22EE	⋮	Vertical ellipsis
hybull	2043	⁃	Hyphen bullet
loz	25CA	◊	Lozenge
lozf		⧫	
ltri	25C3	◃	White left-pointing small triangle
rtri	25B9	▹	White right-pointing small triangle

EntityName	UnicodeCode point	SampleGlyph	Description
starf			BLACK STAR
natur	266E		Music natural sign
rx	211E		Prescription take
sext	2736		Six pointed black star
target	2316		Position indicator
dlcrop	230D		Bottom left crop
drcrop	230C		Bottom right crop
ulcrop	230F		Top left crop
urcrop	230E		Top right crop

■ General Technical Character Entities (%isotech;)

Name

%isotech; – General Technical Character Entities

Synopsis

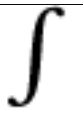
The `%isotech;` parameter entity includes the ISO character entities with the public identifier:

```
ISO 8879:1986//ENTITIES General Technical//EN
```

Description

The following character entities are defined in this entity set:

EntityName	UnicodeCode point	SampleGlyph	Description
aleph	2135	ℵ	Alef symbol
and	2227	∧	Logical and
ang90	221F	∟	Right angle
angsph	2222	∢	Spherical angle
ap	2248	≈	Almost equal to
becaus	2235	∵	Because
bottom	22A5	⊥	Up tack
cap	2229	∩	Intersection
cong	2245	≅	Approximately equal to
conint	222E	∮	Contour integral
cup	222A	∪	Union
equiv	2261	≡	Identical to
exist	2203	∃	There exists
forall	2200	∀	For all

EntityName	UnicodeCode point	SampleGlyph	Description
fnof	0192		Latin small letter f with hook
ge	2265		Greater-than or equal to
iff			LEFT RIGHT DOUBLE ARROW
infin	221E		Infinity
int	222B		Integral
isin	2208		Element of
lang	3008		Left angle bracket
lArr	21D0		Leftwards double arrow
le	2264		Less-than or equal to
minus	2212		Minus sign
mnplus	2213		Minus-or-plus sign
nabla	2207		Nabla
ne	2260		Not equal to
ni	220B		Contains as member

EntityName	UnicodeCode point	SampleGlyph	Description
or	2228	∨	Logical or
par	2225	∥	Parallel to
part	2202	∂	Partial differential
permil	2030	‰	Per mille sign
perp	22A5	⊥	Up tack
prime	2032	′	Prime
Prime	2033	″	Double prime
prop	221D	∝	Proportional to
radic	221A	√	Square root
rang	3009	〉	Right angle bracket
rArr	21D2	⇒	Rightwards double arrow
sim	223C	∼	Tilde operator
sime	2243	≃	Asymptotically equal to
square	25A1	□	White square

EntityName	UnicodeCode point	SampleGlyph	Description
sub	2282	⊂	Subset of
sube	2286	⊆	Subset of or equal to
sup	2283	⊃	Superset of
supe	2287	⊇	Superset of or equal to
there4	2234	∴	Therefore
Verbar	2016	‖	Double vertical line
angst	212B	Å	Angstrom sign
bernou	212C	ℬ	Script capital b
compfn	2218	∘	Ring operator
Dot	00A8	¨	Diaeresis
DotDot	20DC	⃜	Combining four dots above
hamilt	210B	ℋ	Script capital h
lagran	2112	ℒ	Script capital l
lowast	2217	∗	Asterisk operator

EntityName	UnicodeCode point	SampleGlyph	Description
notin	2209	∉	Not an element of
order	2134	ℴ	Script capital o
phmmat	2133	ℳ	Script capital m
tdot	20DB	⃛	Combining three dots above
tprime	2034	‴	Triple prime
wedgeq	2259	≙	Estimates

Part III

Appendixes

Appendix A

Installation

A.1 Installing the DocBook DTD

This appendix describes how to install the DocBook DTD on your system so that popular command-line tools like SP can use it. If you are installing DocBook for use with a commercial application, consult the documentation for your application as well.

This appendix describes the installation of DocBook V3.1. If you are using another distribution, the process should be about the same, but there may be minor differences. DocBook V3.1 is backward compatible with DocBook V3.0, and is the recommended distribution at the time of this writing.

A.1.1 Unpacking the DocBook V3.1 Distribution

DocBook is distributed on the DocBook web site `http://www.oasis-open.org/docbook/`. You will also find a copy of the distribution on the CD-ROM.

The distribution consists of 14 files:

`31chg.txt`	Describes the changes in DocBook V3.1 from the preceding version (3.0)
`40issues.txt`	Summarizes backwards-incompatible changes planned for DocBook V4.0
`50issues.txt`	Summarizes backwards-incompatible changes planned for DocBook V5.0
`cals-tbl.dtd`	The CALS Table Model DTD
`ChangeLog`	A GNU-style ChangeLog summarizing the individual edits made on each file in the distribution since V3.0.
`dbcent.mod`	The character entity module
`dbgenent.mod`	The general entity module
`dbhier.mod`	The document hierarchy module
`dbnotn.mod`	The notations module
`dbpool.mod`	The information pool module
`docbook.cat`	A sample OASIS catalog for DocBook
`docbook.dcl`	An SGML Declaration suitable for DocBook
`docbook.dtd`	The DocBook DTD
`readme.txt`	The DocBook V3.1 “readme” file

Unpack the distribution into a directory on your system. The exact location is irrelevant. On UNIX systems it’s common to put it somewhere under `/usr/local` or `/share` (for example, `/usr/local/sgml/docbook` or `/share/sgml/docbook`). On a PC, perhaps `c:\sgml\docbook`.

A.1.2 Getting the ISO Entity Sets

DocBook refers to a number of standard entity sets that are not distributed with DocBook. (They aren't distributed with DocBook because they aren't maintained by the DocBook TC. They're maintained by ISO.) If you've installed other SGML DTDs or tools, they may already be on your system.

If you are missing some of them, they are available from Robin Cover's pages at OASIS: http://www.oasis-open.org/cover/ISOEnts.zip `http://www.oasis-open.org/cover/ISOEnts.zip`.[1] See `http://www.oasis-open.org/cover/topics.html#entities` for more information.

A.1.3 The DocBook Catalog

DocBook uses public identifiers to refer to its constituent parts. In some sense, DocBook *is* DocBook because it has the formal public identifier "`-//OASIS//DTD DocBook V3.1//EN`". In order for tools on your system to find your locally installed copy of DocBook, you must map these public identifiers into system identifiers, i.e., filenames, on your system. For a complete discussion of catalog files, see Section 2.3.

The DocBook distribution includes a sample catalog, `docbook.cat`, which provides a mapping for all of the public identifiers referenced by DocBook. This mapping won't work "out of the box" for two reasons: first, your tools won't be able to find it, and second, the mappings for the ISO entity sets probably don't point to the right place on your system.

A.1.3.1 Finding the Catalog

If you've already got some other SGML DTDs installed, you probably already have a catalog file. In this case, the easiest thing to do is append the DocBook catalog entries to the end of your existing catalog and then change them to point to the files on your system.

If DocBook is the first DTD that you're installing, make a copy of `docbook.cat` and call it `catalog`. Put this file in a higher-level directory and edit the relative pathnames that it contains to point to the actual locations of the files on your system. For example, if you installed DocBook in `/share/sgml/docbk30/`, put the `catalog` in `/share/sgml/`.

In order for applications to find your catalog file(s), you may have to change the application preferences or set an environment variable. For SP and Jade, set the environment variable `SGML_CATALOG_FILES` to the delimited list of catalog filenames. On my system, this looks like:

```
SGML_CATALOG_FILES=./catalog;n:/share/sgml/catalog;n:/adept80/doctypes/catalog.jade;j:/jad
```

(On a UNIX machine, use colons instead of semicolons to delimit the filenames.)

If you don't wish to set the environment variable, you can explicitly pass the name of each catalog to the SP application with the `-c` option, like this:

```
nsgmls -c ./catalog -c n:/share/sgml/catalog -c othercatalogs
 ...
```

A.1.3.2 Fixing the Catalog

The basic format of each entry in the DocBook catalog is:

```
PUBLIC "some public id" "some filename"
```

What you have to do is change each of the "some filenames" to point to the actual name of the file on your system.

> NOTE
>
> 
>
> Filenames should be supplied using absolute filenames, or paths relative to the location of the *catalog* file.

To continue with the example above, let's say that you've got:

- DocBook in `/share/sgml/docbk30/`,

[1] The names of the entity files in this distribution do not exactly match the names of the files used in the catalog file distributed with DocBook (`docbook.cat`). Make sure your catalog file points to the right files.

- The ISO entities in `/share/sgml/entities/8879/`, and
- Your catalog in `/share/sgml/catalog`

Then you would change the catalog entry for the DTD to be:

```
PUBLIC "-//OASIS//DTD DocBook V3.1//EN" "docbk30/docbook.dtd"
```

You would change the catalog entry for the general technical character entities to:

```
PUBLIC "ISO 8879:1986//ENTITIES General Technical//EN" "entities/8879/iso-tech.gml"
```

And similarly for the other public identifiers used by DocBook. In each case, the filename specified for the public identifier should be the name of the file on your system, specified as an absolute filename, or relative to the location of the `catalog` in which it occurs.

A.1.3.3 Mapping System Identifiers for XML

Since XML documents are required to have system identifiers, but are not required to have public identifiers, it's likely that some of the documents you want to process will only have system identifiers.

It turns out that you can still take advantage of the catalog in this case. The `SYSTEM` directive allows you to map the system identifier used in the document to the actual location on your system.

Suppose that you work with a colleague who uses the system identifier "file:///c:/sgml/db3xml/db3xml.dtd" to identify the XML version of DocBook on her system. On your system, you want to map that to "/share/sgml/db3xml/db3xml.dtd". The following entry in your catalog will do the trick:

```
SYSTEM "http://docbook.org/docbook/xml/1.4/db3xml.dtd" "/share/sgml/db3xml/db3xml.dtd
```

Unfortunately, this technique only works with applications that read and understand catalog files.

A.1.4 Testing Your Installation

The best way to test your installation is with a simple command-line parser like **nsgmls** from SP. Create a small test document, like this:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>Test Chapter</title>
<para>
This is a test document.
</para>
</chapter>
```

and run the following command:

```
nsgmls -sv test.sgm
```

If the **nsgmls** command produces errors, review your catalog and resolve the errors. You can ignore warnings about `DTDDECL` being unsupported. It is unsupported, and there's no way to disable the warning message. Note, however, that this may also affect which SGML declaration gets used. When in doubt, pass the correct declaration explicitly and see if that corrects any problems. (To parse `test.sgm` with the declaration `docbook.dcl` explicitly, run **nsgmls -sv docbook.dcl test.sgm**.)

For some suggestions about how to work around these problems in SP and Jade, see the next section, Section A.2.

A.2 Installing Jade

For simplicity, the instructions assume that you are working with Jade on a Microsoft Windows machine. These instructions should be just as useful if you are working on another platform, except for the normal cross-platform idiosyncracies (path and filename separator characters, use of drive letters, etc.).

Download and unpack the Jade `http://www.jclark.com/jade/` distribution. Binary distributions are available for some platforms, which makes installation a simple matter of unpacking the distribution. Or you can build Jade from the source (consult the documentation that comes with Jade for more detail about building it from source).

You may wish to add the directory where you installed Jade to your `PATH`. If not, make sure that you use the fully qualified name of the executable when you run the commands below.

A.2.1 Setting Up the Catalog

First, the catalog needs to be set up as described in Section A.1.3 in order for Jade to be able to parse your DocBook documents. In addition, Jade comes with its own `catalog` file that you must add to the `SGML_CATALOG_FILES` environment variable or otherwise make available to Jade.

A.2.2 Testing Jade

Download `jtest.sgm` http://nwalsh.com/docbook/dsssl/doc/testdata/jtest.sgm and `jtest.dsl` http://nwalsh.com/docbook/dsssl/doc/testdata/jtest.dsl. (Or get them off the CD-ROM in FIXME.) These are self-contained test documents. Test Jade by running:

```
jade -t rtf -d jtest.dsl jtest.sgm
```

This command should silently produce jtest.rtf http://nwalsh.com/docbook/dsssl/doc/testdata/jtest.rtf. If you encounter warnings or errors here, Jade is not installed correctly. One possible culprit is your catalog setup. See Section 4.3.9.

A.2.2.1 DTDDECL Warnings

One annoying shortcoming in Jade is that it does not support the `DTDDECL` catalog directive and it complains loudly if it encounters one. In Jade, it's almost always possible to work around the problems that `DTDDECL` would solve, so you can generally ignore the warnings.

If you also use applications that do understand `DTDDECL`, and find the warnings too distracting to bear, setup alternate catalogs for SP applications, `catalog.jade`, that are identical to your normal catalogs but do not contain any `DTDDECL` entries. You can then avoid the warnings by putting `catalog.jade` in your `SGML_CATALOG_FILES` path, instead of `catalog`.

A.3 Installing the Modular DocBook Stylesheets

Norman Walsh http://nwalsh.com/~ndw/ (one of your intrepid authors ;-) maintains two DSSSL stylesheets for DocBook, one for print and one for online (HTML) output. You can obtain both of these stylesheets from http://nwalsh.com/docbook/dsssl/ http://nwalsh.com/docbook/dsssl/. (A recent version is also on the CD-ROM.)

1. If you have not already done so, download and install the DocBook DTD as described in Section A.1.
2. Likewise, if Jade is not installed on your system, download and install it as described in Section A.2.
3. Download and unpack the stylesheet distribution http://nwalsh.com/docbook/dsssl/.
4. Test the installation by processing `test.sgm` (from the previous section) with Jade:

   ```
   jade -t rtf -d d:\where-you-unpacked-the-stylesheets\docbook\print\docbook.dsl test.sg
   ```

   This command should silently produce `test.rtf` http://nwalsh.com/docbook/dsssl/doc/testdata/test.rtf. If not, and the preceding test succeeded, something has gone wrong—contact the maintainer http://nwalsh.com/~ndw/contact.html. To test the HTML stylesheet, run:

   ```
   jade -t sgml -d d:\where-you-unpacked-the-stylesheets\docbook\html\docbook.dsl test.sg
   ```

   This command should silently produce `c01.htm` http://nwalsh.com/docbook/dsssl/doc/testdata/c01.htm. If not, and the preceding test succeeded, something has gone wrong—contact the maintainer http://nwalsh.com/~ndw/contact.html.

Appendix B

DocBook and XML

XML, the Extensible Markup Language http://www.w3.org/TR/REC-xml, is a simple dialect of SGML. In the words of the XML specification, "the goal [of XML] is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML."

XML raises two issues with respect to DocBook:

- Are DocBook SGML instances valid XML instances?
- Can the DocBook DTD be made into a valid XML DTD?

If you have an existing SGML system, and your primary goal is to serve DocBook documents over the Web as XML, only the first of these issues is relevant. As the popularity of XML grows, we will see more and more XML-aware tools that don't implement full ISO 8879 SGML. If your goal is to author DocBook documents with one of this new generation of tools, you will only be able to achieve validity with an XML DocBook DTD.

Although not yet officially adopted by the OASIS DocBook Technical Committee, an XML version of DocBook is available now and provided on the CD-ROM.

B.1 DocBook Instances as XML

Most DocBook documents can be made into well-formed XML documents very easily. With few exceptions, valid DocBook SGML instances are also well-formed XML instances. The following areas may need to be addressed.

B.1.1 System Identifiers

It is common for SGML instances to use only a public identifier in document type and parameter entity declarations:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>Chapter Title</title>
<para>
This <emphasis>paragraph</paragraph> is important.
</para>
</chapter>
```

XML requires a system identifier:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Chapter Title</title>
<para>
This <emphasis>paragraph</paragraph> is important.
</para>
</chapter>
```

If you're used to using catalog files to resolve system identifiers, you may be dismayed to learn that system identifiers are required. Because most tools favor system identifiers over public identifiers, all of the portability that was gained by the use of catalog files seems to have been lost. In the long run, it'll be regained by the fact that XML system identifiers can be URNs, which will have a resolution scheme like catalogs, but what about the short run?

Luckily, there are a couple of options. First, you can tell your tools to use the public identifiers even though system identifiers are present. Simply add:

```
OVERRIDE YES
```

to your catalog files. Alternatively, you can remap system identifers with the `SYSTEM` catalog directive. If you are faced with documents that don't use public identifiers at all, this is probably your only option.

B.1.2 Minimization

If you have used SGML minimization features in your instances:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter id=chap1><title>Chapter Title</title>
<para>
This <emphasis>paragraph</> is important.
</para>
</chapter>
```

they will not be well-formed XML instances. In particular, XMLxml-attrquote Requires that all attribute values be quoted. xml-endtag Does not allow short tag minization.

XML also forbids tag omission, and there are probably a half dozen or so more exotic examples of minimization that you have used. They're all illegal. The easiest way to remove these minimizations is probably with a tool like **sgmlnorm** (included in the SP and Jade distributions, on the CD-ROM).

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                   "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter id="chap1"><title>Chapter Title</title>
<para>
This <emphasis>paragraph</emphasis> is important.
</para>
</chapter>
```

B.1.3 Attribute Default Values

Correct processing of this document may require access to the default attributes:

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>Chapter Title</title>
<para>
Write to us at:
<address>
90 Sherman Street
Cambridge, MA 02140
</address>
</para>
</chapter>
```

xml-defattr `Address` expresses that its content is line-specific with an attribute.

Some XML processing environments are going to ignore the doctype declaration in your document, even if it's present. This is relevant when your instance uses elements that have attributes with default values. The default values are expressed in the DTD, but may not be expressed in your instance. In the case of DocBook, there are relatively few of these, and your stylesheet can probably be constructed to do the right thing in either case. (It essentially treats the attributes as if they had implied values.)

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Chapter Title</title>
<para>
Write to us at:
<address format="linespecific">
90 Sherman Street
Cambridge, MA 02140
</address>
</para>
</chapter>
```

B.1.4 Character and SDATA Entities

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>Chapter Title</title>
<para>
This book was published by O'Reilly&trade;.
</para>
</chapter>
```

xml-sdata

The DocBook DTD defines all of the standard ISO entities automatically, but the ISO definitions use SDATA, which is not allowed in XML. Eventually, ISO (or someone else) will release official ISO standard entity sets that make reference to the appropriate Unicode character for each entity. Until then, the XML version of DocBook is distributed with an unofficial set.

If you use entities in your document, it may be wise to put declarations for them in the internal subset of each instance, because some XML browsers are going to parse the internal subset but not the external subset. If the entity declarations are in your DTD, and the browser does not parse the external subset, the browser won't know how to display the entities in your document.

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd" [
<!ENTITY trade "&#x2122;">
<chapter><title>Chapter Title</title>
<para>
This book was published by O'Reilly&trade;.
</para>
</chapter>
```

B.1.5 Case-Sensitivity

```
<!DocType Book PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<book><title>Book Title</title>
<chapter><title>Chapter Title</Title>
<para>
Paragraph test.
</para>
<PARA>
A second paragraph.
</PARA>
</chapter>
</book>
```

With the standard DocBook SGML declaration, DocBook instances are not case-sensitive with respect to element and attribute names. XML is always case-sensitive. As long as you have used the same case consistently, your XML instances will be well-formed, but it may still be advantageous to do some case-folding because it will

simplify the construction of stylesheets. xml-nmcasekey Keywords in XML are case-sensitive, and must be in uppercase. xml-namecase1 The name declared in the document type declaration, like all other names, is case-sensitive. xml-namecase2 Start and end tags must use the same case. xml-wf1 In XML, `Para` is not the same as `PARA`. Note that this is a validity error (against the XML version of DocBook), but it is not an XML well-formedness error. The use of `para` and `PARA` as distinct names is as legitimate as using `foo` and `bar`, as long as they are properly nested.

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<book><title>Book Title</title>
<chapter><title>Chapter Title</title>
<para>
Paragraph test.
</para>
<para>
A second paragraph.
</para>
</chapter>
</book>
```

B.1.6 No #CONREF Attributes

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<chapter><title>Chapter Title</title>
<indexterm id="idx-bor"><primary>Something</primary></indexterm>
<para>
Paragraph test.
</para>
<indexterm startref="idx-bor">
</chapter>
```

The StartRef attribute on `indexterm` and the OtherTerm attribute on `GlossSee` and `GlossSeeAlso` are `#CONREF` attributes.

In SGML terms, this means that when these attributes are used, the content of the tag is taken to be the same as the content of the tag pointed to by the attribute. xml-conrefstart xml-conref If you have used these attributes, your instance will contain both empty and non-empty versions of these tags.

Your best bet is to transform the `#CONREF` version into an empty tag and let your stylesheet deal with it appropriately.

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Chapter Title</title>
<indexterm id="idx-bor"><primary>Something</primary></indexterm>
<para>
Paragraph test.
</para>
<indexterm startref="idx-bor"/>
</chapter>
```

B.1.7 Only Explicit CDATA-Marked Sections Are Allowed

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY % draft "IGNORE">
<!ENTITY % sourcecode "CDATA">
]>
<chapter><title>Chapter Title</title>
```

```
<![ %draft; [
<para>
Draft paragraph.
</para>
]]>
<para>
The following code is totally out of context:
<programlisting>
<![ %sourcecode; [
if (x < 3) {
  y = 3;
}
]]>
</programlisting>
</chapter>
```

xml-ignore xml-cdata

Parameter entities are not allowed in the body of XML documents (they are allowed in the internal subset). xml-ignore XML instances cannot contain `IGNORE`, `INCLUDE`, `TEMP`, or `RCDATA` marked sections. xml-cdata `CDATA` marked sections must use the "`CDATA`" keyword literally because parameter entities are not allowed.

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
<chapter><title>Chapter Title</title>
<para>
The following code is totally out of context:
<programlisting>
<![CDATA[
if (x < 3) {
  y = 3;
}
]]>
</programlisting>
</chapter>
```

B.1.8 No SUBDOC or CDATA External Entities

```
<!DOCTYPE chapter PUBLIC "-//OASIS//DTD DocBook V3.1//EN" [
<!ENTITY % sourcecode SYSTEM "program.c" CDATA>
]>
<chapter><title>Chapter Title</title>
<para>
The following code is totally out of context:
<programlisting>
&sourcecode;
</programlisting>
</chapter>
```

XML instances cannot use `CDATA` or `SUBDOC` external entities. One option for integrating external `CDATA` content into a document is to employ a pre-processing pass that inserts the content inline, wrapped in a `CDATA` marked section.

`SUBDOC` entities may be more problematic. If you do not require validation, it may be sufficient to simply put them inline. XML namespaces may offer another possible solution.

The result will be something like this:

```
<?xml version='1.0'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
                  "http://www.oasis-open.org/docbook/xml/4.1.2/docbookx.dtd">
```

```
<chapter><title>Chapter Title</title>
<para>
The following code is totally out of context:
<programlisting>
<![CDATA[
int main () {
..
}
]]>
</programlisting>
</chapter>
```

B.1.9 No Data Attributes on Notations

They're not allowed in XML, so don't add any.

B.1.10 No Attribute Value Specifications on Entity Declarations

They're not allowed in XML, so don't add any.

B.2 The DocBook DTD as XML

Converting the DocBook DTD to XML is much more challenging than converting the instances. It is probably not possible to construct an XML DTD that is identical to the validation power of DocBook. The list below identifies most of the issues that must be addressed, and describes how the DocBook XML DTD; deals with them:

Comments are not allowed inside markup declarations Most of them have been moved to comment declarations preceding the markup declaration that used to contain them. A few small, inline comments that seemed like they would be out of context if moved before the declaration were simply deleted.

Name groups are not allowed in element or attribute list declarations The small number of places in which DocBook uses name groups have been expanded.

There's one downside: DocBook uses `%admon.class;` in a name group to define the content model, and attribute lists for elements in the admonitions class. In DocBook XML, this convenience cannot be expressed. If additional admonitions are added, the element and attribute list declarations will have to be copied for them.

No `CDATA` or `RCDATA` declared content `Graphic` and `InlineGraphic` have been made `EMPTY`. The content model for `SynopFragmentRef`, the only `RCDATA` element in DocBook, has been changed to `(arg | group)+`.

No exclusions or inclusions on element declarations They had to be removed.

In DocBook, exclusions are used to exclude the following:

- Ubiquitous elements (`indexterm` and `BeginPage`) from a number of contexts in which they should not occur (such as metadata, for example).
- Formal objects from `Highlights`, `Example`s, `Figure`s and `LegalNotice`s.
- Formal objects and `InformalTable`s from tables.
- Block elements and `Footnote`s from `Footnote`s
- Admonitions, `EntryTbl`s, and `Acronym`s from themselves.

Removing these exclusions from DocBook XML means that it is now valid, in the XML sense, to do some things that don't make a lot of sense (like put a `Footnote` in a `Footnote`). Be careful.

Inclusions in DocBook are used to add the ubiquitious elements (`indexterm` and `BeginPage`) unconditionally to a large number of contexts. In order to make these elements available in DocBook XML, they have been added to most of the parameter entities that include `#PCDATA`. If new locations are discovered where these terms are desired, DocBook XML will be updated.

Elements with mixed content must have `#PCDATA` first. The content models of many elements have been updated to make them a repeatable OR group beginning with `#PCDATA`.

Many declared attribute types (`NAME`, `NUMBER`, `NUTOKEN`, and so on) are not allowed They have all been replaced by `NMTOKEN` or `CDATA`.

No `#CONREF` attributes allowed. The `#CONREF` attributes on `indexterm`, `GlossSee`, and `GlossSeeAlso` were changed to `#IMPLIED`. The content model of `indexterm` was modified so that it can be empty.

Attribute default values must be quoted. Quotes were added wherever necessary.

Appendix C

DocBook Versions

The OASIS DocBook Technical Committee (TC) is committed to the continued evolution of DocBook. As new needs are expressed by the DocBook-user community, the committee will adapt DocBook so it continues to meet the needs of its users.

The TC observes a very cautious policy regarding changes to the DTD. Backward-incompatible changes can only be introduced:

- In major releases (4.0, 5.0, 6.0, and so on)
- Only if the change was described in comments in the DTD in the previous major release

Changes made at point-releases are always backward-compatible to the previous major release. This policy assures that DocBook users always have time to prepare for any coming changes.

This appendix describes the changes planned for DocBook.

C.1 DocBook V3.1

DocBook V3.1, released in February 1999, introduced a number of new elements:

`MediaObject` `MediaObject` is a wrapper around `VideoObject`, `AudioObject`, `ImageObject`, and `TextObject`. The purpose of `MediaObject` is twofold: first, it introduces new element types to include video and audio content; second, it provides the option of alternative representations, including text, for objects in a document. `MediaObject`s also allow `Caption`s that may be longer than a simple title.

Parallel to `MediaObject` are `InlineMediaObject`, an inline version, and `MediaObjectCO`, a media object with `Callout`s.

For now, media objects and graphics exist together in version 5.0. `Graphic` and `InlineGraphic` will be removed from DocBook.

`InformalFigure` A figure wrapper without a title. This element was added largely for symmetry.

`Colophon` An end-of-the-book `Colophon`.

`Section` `Section` is a recursive section. It exists parallel to, and must be used as an alternative to, `Sect1`.

`QandASet` A set of questions and answers.

`Constant` The `Constant` inline is for identifying constants. It has a Class attribute for identifying “limits” as a specific kind of constant.

VarName Many authors have requested a tag for identifying variable names. Most often Literal (with or without a role) has been chosen as a compromise, but you can't question the utility of identifying variable names, so VarName was added.

A tag for identifying variable values does not seem as necessary. For immutable values, a specific tag other than Literal seems unnecessary, and Replaceable exists for values that are supposed to be filled in by the user.

C.2 DocBook V4.1 and DocBook XML V4.1.2

DocBook V4.1 and DocBook XML V4.1.2 became an OASIS Standard in February 2001.

DocBook V4 introduced a number of backward-incompatible changes.

The policy of the DocBook TC is that backward-incompatible changes can only be introduced in a full version release. They must also be described in comments in the DTD at least one full version earlier. In other words, all the planned version 4.0 incompatibilities were announced with version 3.0.

Backward-incompatible changes to DocBook that are planned for version 4.0:

- The DocInfo element will be split out into ChapterInfo, AppendixInfo, and so on. ArtHeader will be renamed to ArticleInfo. SeriesInfo may be discarded because it has become a special case of BiblioSet. BookBiblio will be discarded in favor of a new, more inclusive, structure for BookInfo (and for ArticleInfo, whose earlier ArtHeader form contained BookBiblio).
- The %article.class; entity may be removed from the Book content model, and made part of a new top-level document hierarchy.
- The %nav.class; entity, which appears in several divisions, will allow ToCchap instead of ToC.
- MsgText will be moved from %tech.char.class; to a more appropriate parameter entity.
- The following elements will have their content constrained to the %smallcptr.char.mix; mixture: Action, Command, Database, Filename, Hardware, Interface, KeyCap, Literal, Option, Parameter, Property, and SystemItem.
- AuthorBlurb and Affiliation will be removed from %person.ident.mix; and a new wrapper element will be created to allow association of those two elements with Author name information.
- Epigraph will not be appearing in BlockQuote.
- Comment will be renamed to Remark and will be excluded from itself.
- GlossTerm will be excluded from itself, and may be split into an element that appears in a Glossary and an element that can appear in the main text.
- The Subject attribute on GlossDef will be renamed Keyword.
- Two SegTitle elements will be required in SegmentedList.
- Graphic and InlineGraphic will be declared EMPTY. This change will require that end tags be removed and that any embedded graphic content be stored outside the SGML source and pointed to from an Entityref or Fileref attribute.
- The OptMult and ReqMult values for the Choice attribute on Group will be removed. Use the Rep attribute instead to indicate that the choice is repeatable.
- The content model group inside FuncSynopsis starting with FuncDef will not be available; you will have to use FuncPrototype. Also, you will be able to have a mixture of FuncPrototypes and FuncSynopsisInfos (this is not backward-incompatible all by itself).
- The EnvironVar and Prompt values for the Class attribute on SystemItem will be eliminated; use the EnVar and Prompt elements instead.

C.3 DocBook V4.2

DocBook V4.2 fixed a couple of small bugs and introduced a number of new features. Changes in DocBook V4.2 are explained in the DocBook Document Type Version 4.2 http://www.oasis-open.org/docbook/docbook-4.2.html.

C.4 EBNF Module

FIXME: write this section

C.5 HTML Forms Module

FIXME: write this section

C.6 MathML Module

FIXME: write this section

C.7 SVG Module

FIXME: write this section

Appendix D

Resources

The quantity of information about SGML and XML is growing on a daily basis. This appendix strives to provide both a complete bibliography of the references mentioned explicitly in this book, and a sampling of resources for additional information about DocBook and about SGML and XML in general. Although not all of these resources are focused specifically on DocBook, they still provide helpful information for DocBook users.

D.1 Latest Versions of DocBook

As of July 1998, responsibility for the advancement and maintenance of the DocBook DTD has been transferred from the Davenport Group, which originated it, to the DocBook Technical Committee of OASIS (Organization for the Advancement of Structured Information Standards) at http://www.oasis-open.org/ http://www.oasis-open.org/.

The latest releases of DocBook can be obtained from the official DocBook home page at http://www.oasis-open.org/docbook/ http://www.oasis-open.org/docbook/.

D.2 Resources for Resources

Here's where to find pointers to the subjects you want to find.

The Most Recent Version of This Book The most recent online version of this book can be found at http://docbook.org/.

The Most Recent Version of DocBook The most recent version of DocBook, and the most recent information about the DTD, can be found at the DocBook home page: http://www.oasis-open.org/docbook/.

Robin Cover's SGML/XML Web page Easily the largest and most up-to-date list of SGML/XML resources; can be found at http://www.oasis-open.org/cover/.

`comp.text.sgml` and `comp.text.xml` USENET newsgroups devoted to SGML and XML issues.

FAQs For pointers to several SGML FAQs, see http://www.oasis-open.org/cover/general.html#faq. The XML FAQ is available at http://www.ucc.ie/xml.

XML.com http://www.xml.com/ XML.com http://www.xml.com/, run jointly by Songline Studios and Seybold, is a site devoted to making XML accessible.

D.3 Introductory Material on the Web

These documents provide a good background for a better understanding of SGML and XML.

A Gentle Introduction to SGML A useful and simple document available in its original form at http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html.

A Technical Introduction to XML A close look at the ins-and-outs of XML is available at http://nwalsh.com/docs/articles/xml/ http://nwalsh.com/docs/articles/xml/.

D.4 References and Technical Notes on the Web

Entity Management OASIS Technical Resolution 9401:1997 (Amendment 2 to TR 9401) http://www.oasis-open.org/html/a401.htm.

This document describes OASIS catalog files.

The SGML Declaration The SGML Declaration, http://www.oasis-open.org/cover/wlw11.html by Wayne Wholer.

Table Interoperability: Issues for the CALS Table Model OASIS Technical Research Paper 9501:1995 http://www.oasis-open.org/html/a501.htm.

Exchange Table Model Document Type Definition OASIS Technical Resolution TR 9503:1995 http://www.oasis-open.org/html/a503.htm.

CALS Table Model Document Type Definition OASIS Technical Memorandum TM 9502:1995 http://www.oasis-open.org/html/a502.htm

XML Exchange Table Model Document Type Definition OASIS Technical Memorandum TM 9901:1999 http://www.oasis-open-org/html/a901.htm.

D.5 Internet RFCs

RFCs ("Request for Comments") are standards documents produced by the Internet Engineering Task Force (IETF).

RFC 1630 **http://www.cis.ohio-state.edu/htbin/rfc/rfc1630.html** Universal Resource Identifiers in WWW.

RFC 1736 **http://www.cis.ohio-state.edu/htbin/rfc/rfc1736.html** Functional recommendations for Internet Resource Locators.

RFC 1737 **http://www.cis.ohio-state.edu/htbin/rfc/rfc1737.html** Functional requirements for Uniform Resource Names.

RFC 1738 `http://www.cis.ohio-state.edu/htbin/rfc/rfc1738.html` Uniform Resource Locators (URL).

RFC 3066 `http://www.cis.ohio-state.edu/htbin/rfc/rfc3066.html` Tags for the identification of languages

D.6 Specifications

Here are pointers to the specifications.

The XML Specification `http://www.w3.org/TR/REC-xml` The W3C technical recommendation that defines XML 1.0.

Namespaces in XML `http://www.w3.org/TR/REC-xml-names/` The W3C technical recommendation that defines XML namespaces.

Mathematical Markup Language (MathML) 1.0 Specification `http://www.w3.org/TR/REC-MathML/` The W3C technical recommendation that defines MathML, an XML representation of mathematical equations.

The Unicode Standard, Version 2.0 `http://www.unicode.org/unicode/uni2book/u2.html` The Unicode standard.

Unicode Technical Report #8 `http://www.unicode.org/unicode/reports/tr8.html` Version 2.1 of the Unicode standard.

Dublin Core Metadata Element Set, Version 1.1: Reference Description `http://dublincore.org/documents/dc` Version 1.1 of the Dublin Core Metadata Initiative's Metadata ELement Set.

Getty Thesaurus of Geographic Names `http://www.getty.edu/research/tools/vocabulary/tgn/` A controlled vocabulary of geographic place names.

D.7 Books and Printed Resources

There are also a number of books worth checking out:

[1] *Practical SGML*, Erik van Herwijnen, 0-7923-9434-8, Kluwer Academic Press, 1994.

[2] *The SGML Handbook*, Charles Goldfarb and Yuri Rubinksy, 0-7923-9434-8, 1991, Oxford University Press.

[3] *SGML: an author's guide to the Standard Generalized Markup Language*, Martin Bryan, 0-201-17535-5, 1988, Addison-Wesley Publishing Company.

[4] $GML : The Billion Dollar Secret, Chet Ensign, 0-13-226705-5, 1998, Prentice Hall.$

[5] *Creating Documents with XML*, Chris Maden, 1-56592-518-1, 1999, O'Reilly Associates.

[6] *XML: A Primer*, Simon St. Laurent, 1-5582-8592-X, 1998, MIS:Press/IDG Books Worldwide.

[7] *Understanding SGML and XML Tools*, Peter Flynn, 0-7923-8169-6, 1998, Kluwer Academic Publishers.

[8] *The LaTeX Web Companion*, Michel Goosens and Sebastian Rahtz, 0-201-43311-7, 1999, Addison-Wesley Publishing Company.

[maler96] *Developing SGML DTDs*, Eve Maler and Jeanne El Andaloussi, 0-13-309881-8, Prentice-Hall PTR Upper Saddle River New Jersey , 1996. 4

D.8 SGML/XML Tools

An attempt to provide a detailed description of all of the SGML/XML tools available is outside the scope of this book.

For a list of recent of SGML tools, check out Robin Cover's SGML/XML page at OASIS: http://www.oasis-open.org/cover `http://www.oasis-open.org/cover`.

For a list of XML tools, check out XML.com: `http://www.xml.com/`.

Appendix E

What's on the CD-ROM?

The CD-ROM that accompanies the print version of this book contains a number of useful resources.

Please read the `readme.txt` file in the root directory of the CD-ROM. It describes any last-minute changes or additions that were made to the CD-ROM after this appendix was written.

E.1 DocBook: The Definitive Guide

`/sgml/` contains the sources for this book in SGML.

`/html/` contains an online version of this book in HTML.

`/hepl/` contains a compiled HTML Help version of this book.

`/examples/` contains the complete examples from this book.

E.2 The DocBook DTD

`/dtds/docbook/db30` contains DocBook V3.0.

`/dtds/docbook/db31` contains DocBook V3.1.

`/dtds/docbk/db315` contains DocBk XML V3.1.5.

`/dtds/docbk/sdb3151` contains a simplified DocBk XML V3.1.5.1.

`/dtds/isoents` contains ISO entity sets needed for the DTDs.

`/dtds/usadod` contains the CALS table model DTD needed for the SGML DocBook DTDs.

E.3 Stylesheets

`/style/` contains various stylesheets.

`/style/dsssl/docbook/` contains the Modular DocBook DSSSL stylesheets. These are the Jade stylesheets described in Chapter 4, *Publishing DocBook Documents*.

`/dtds/dsssl/anotess/` contains the Annotated DSSSL stylesheet DTD and related files.

`/style/xsl/` contains various XSL stylesheets.

`/style/xsl/docbook/` contains the XSL DocBook Stylesheets V0.12.

E.4 Other Programs

The `/apps` directory contains source and binary releases of other, related software, including the Jade and XT distributions. The `/bin` directory contains the format script briefly discussed in Chapter 5, *Customizing DocBook*.

Appendix F

Interchanging DocBook Documents

One of the early factors that motivated the development of the DocBook DTD was the desire for companies to interchange documents. In particular, UNIX vendors wanted to be able to interchange common UNIX documentation.

A great deal of effort went into making sure that DocBook could handle most (probably all) of the documents that were likely to be exchanged. This avoids the guaranteed interchange problem of DTD extension.

However, simply using DocBook or a subset of it is not enough to ensure successful interchange. If you send someone your DocBook files, you must also tell the recipient about the markup your documents use and any of your additional markup conventions and processing expectations that impose constraints on processing.

This appendix provides a sample interchange questionnaire to help draw your attention to those areas that might be problematic.

For maximum portability, delivered DocBook documents should be accompanied by a filled-out interchange questionnaire. Because each situation is unique, you may need to supply additional information (such as layout specifications) in order to deliver a complete package.

DocBook and SGML Usage

1. What version of the DTD are you using?
2. Did you use any markup features of the DTD that have been flagged as obsolete (to be removed at the next major version of DocBook)? If so, which ones?
3. Did you extend DocBook in any way, inside or outside the provided customization mechanisms? How? All extensions must be negotiated with the recipient.
4. Did you remove markup from DocBook to create a subset? If you used a subset of DocBook, supply the subset you used. (Note that even the removal of references to ISO entity sets creates a subset.)
5. Did you use the supplied SGML declaration or another one? If you used another one, provide it.
6. Did you use the supplied catalog or another one, or none at all? If you used a catalog other than the one supplied, provide it.
7. If your documents bear no document type declaration, and you parsed them with a document declaration (with or without an internal subset), supply it.
8. Did you add `NOTATION` declarations? If so, what are they? List all data content notations used in your documents.
9. Did you use the `SUBDOC` feature? If so, how did you manage the name spaces of their IDs, if you managed them at all?
10. Did you use character sets other than ISO 8859-1 (Latin 1)? If so, which ones? How did you use them?
11. Did you declare and use character entities and other general entities besides the ISO entity sets? If so, supply the entity declarations and the desired appearance of the additional character entities.
12. Are your document files normalized to include all markup explicitly?

13. Are you supplying a document fragment? If so, have you provided any necessary auxiliary information (such as meta-information) for the fragment? Are there any attribute values that haven't been specified that you expect to inherit from a parent that isn't present?

Processing Requirements and Markup Interpretation

1. What formatting that you applied do you require your interchange partner to apply? For example, where and how must text be generated in order for the documents to make sense?
2. Did you supply your stylesheet and information regarding its format and version?
3. How did you create tables of contents, lists of titles, and indexes? Are they stored in DocBook form? If so, did you generate them (and according to what rules) or create them by hand?
4. If you used the Lang common attribute, why, and to what effect?
5. If you used the Remap common attribute, why, and to what effect?
6. If you used the Role common attribute, why, and to what effect?
7. If you used the effectivity attributes, which did you use, why, and to what effect?
8. What values did you give to the Label attribute and how are they to be interpreted for rendering?
9. What values did you give to the Mark and Override attributes for lists and how are they to be interpreted for rendering?
10. Did you use the Renderas attribute on sections and/or `BridgeHead`s?
11. Did you supply all keyword values you used for attributes whose declared values are not enumerated tokens, along with the expected processing for the occurrence of each keyword?
12. Did you use markup to control width, size, and/or positioning settings (such as "fold-out" or "centered") for graphics, line specific regions, and tables? If so, how?
13. For rendering of `Sidebar`s, must these appear in the flow of the text where they appear in your files, or may they float?
14. Did you use `Callout`s? If so, what are the processing expectations for callout marks?
15. Did you use `ItemizedList`s? If so, what are the processing expectations for the marks on list items and nested lists?
16. For `Graphic` and `InlineGraphic`, what method(s) did you use for providing graphic data: element content, Fileref attribute, or Entityref attribute?
17. For `MediaObject` and `InlineMediaObject`, what method(s) did you use for selecting between alternative presentations?
18. How did you specify column widths in tables? Did you use vertical spans? Did you use horizontal spans?
19. Did you use `EntryTbl`s?
20. If you used the Type attribute on the link elements, why, and to what effect?
21. If you used `XRef`, do your interchange partners need additional information about the semantic of the link? Have you provided it, perhaps with Role?
22. Did you use the Subject attribute on `GlossDef`? If so, did you use a thesaurus of terms? If so, what is it?
23. If you used the Class attribute on `RefMiscInfo`, why and to what effect?
24. If you used `ULink` and provided URLs that are queries, what back-end processing is required to resolve those queries?
25. FileRef or EntityRef is supplied on every `Graphic`, `InlineGraphic`, `AudioData`, `ImageData`, and `VideoData` element. If one is not present, what is the expectation?

26. If your `Bibliography`s or `Glossary`s have special processing expectations, such as the ability to display only those entries that are cited, have you described them?
27. If your `Bibliography`s contain `BiblioEntry`s, what are the processing expectations? Which fields are selected for display? What punctuation is added, and where?
28. Do `GlossTerm`s or other elements have implicit linking relationships that must be obeyed or handled in presentation?
29. Did you use any processing instructions? Why and what for? Are they in entities?
30. What copyfitting have you already done, and for what outputs?
31. Are the `Revision`s in your `RevHistory`s sorted in any particular way?

Miscellaneous

1. Have you checked your files for viruses?
2. If you used `BridgeHead`, have you joined a recovery support group?

Appendix G

DocBook Quick Reference

G.1 DocBook V3.1

Element	Brief Description
Abbrev	An abbreviation, especially one followed by a period
Abstract	A summary
Accel	A graphical user interface (GUI) keyboard shortcut
Ackno	Acknowledgements in an Article
Acronym	An often pronounceable word made from the initial (or selected) letters of a name or phrase
Action	A response to a user event
Address	A real-world address, generally a postal address
Affiliation	The institutional affiliation of an individual
Alt	Text representation for a graphical element
Anchor	A spot in the document
Answer	An answer to a question posed in a QandASet
Appendix	An appendix in a Book or Article
Application	The name of a software program
Area	A region defined for a Callout in a graphic or code example
AreaSet	A set of related areas in a graphic or code example
AreaSpec	A collection of regions in a graphic or code example
Arg	An argument in a CmdSynopsis
ArtHeader	Meta-information for an Article
ArtPageNums	The page numbers of an article as published
Article	An article
Attribution	The source of a block quote or epigraph
AudioData	Pointer to external audio data
AudioObject	A wrapper for audio data and its associated meta-information
Author	The name of an individual author
AuthorBlurb	A short description or note about an author
AuthorGroup	Wrapper for author information when a document has multiple authors or collabarators
AuthorInitials	The initials or other short identifier for an author
BeginPage	The location of a page break in a print version of the document
BiblioDiv	A section of a Bibliography
BiblioEntry	An entry in a Bibliography
BiblioMSet	A “cooked” container for related bibliographic information
BiblioMisc	Untyped bibliographic information
BiblioMixed	An entry in a Bibliography
BiblioSet	A “raw” container for related bibliographic information
Bibliography	A bibliography
BlockQuote	A quotation set off from the main text

Element	Brief Description
Book	A book
BookBiblio	Meta-information about a book used in a bibliographical citation
BookInfo	Meta-information for a Book
BridgeHead	A free-floating heading
CO	The location of a callout embedded in text
Callout	A “called out” description of a marked Area
CalloutList	A list of Callouts
Caption	A caption
Caution	A note of caution
Chapter	A chapter, as of a book
Citation	An inline bibliographic reference to another published work
CiteRefEntry	A citation to a reference page
CiteTitle	The title of a cited work
City	The name of a city in an address
ClassName	The name of a class, in the object-oriented programming sense
CmdSynopsis	A syntax summary for a software command
ColSpec	Specifications for a column in a table
Collab	Identifies a collaborator
CollabName	The name of a collaborator
Colophon	Text at the back of a book describing facts about its production
Command	The name of an executable program or other software command
Comment	A comment intended for presentation in a draft manuscript
ComputerOutput	Data, generally text, displayed or presented by a computer
ConfDates	The dates of a conference for which a document was written
ConfGroup	A wrapper for document meta-information about a conference
ConfNum	An identifier, frequently numerical, associated with a conference for which a document was written
ConfSponsor	The sponsor of a conference for which a document was written
ConfTitle	The title of a conference for which a document was written
Constant	A programming or system constant
ContractNum	The contract number of a document
ContractSponsor	The sponsor of a contract
Contrib	A summary of the contributions made to a document by a credited source
Copyright	Copyright information about a document
CorpAuthor	A corporate author, as opposed to an individual
CorpName	The name of a corporation
Country	The name of a country
Database	The name of a database, or part of a database
Date	The date of publication or revision of a document
Dedication	A wrapper for the dedication section of a book
DocInfo	Meta-data for a book component
Edition	The name or number of an edition of a document
Editor	The name of the editor of a document
Email	An email address
Emphasis	Emphasized text
EnVar	A software environment variable
Entry	A cell in a table
EntryTbl	A subtable appearing in place of an Entry in a table
Epigraph	A short inscription at the beginning of a document or component
Equation	A displayed mathematical equation
ErrorCode	An error code
ErrorName	An error name
ErrorType	The classification of an error message
Example	A formal example, with a title
Fax	A fax number

Element	Brief Description
Figure	A formal figure, generally an illustration, with a title
Filename	The name of a file
FirstName	The first name of a person
FirstTerm	The first occurrence of a term
Footnote	A footnote
FootnoteRef	A cross reference to a footnote (a footnote mark)
ForeignPhrase	A word or phrase in a language other than the primary language of the document
FormalPara	A paragraph with a title
FuncDef	A function (subroutine) name and its return type
FuncParams	Parameters for a function referenced through a function pointer in a synopsis
FuncPrototype	The prototype of a function
FuncSynopsis	The syntax summary for a function definition
FuncSynopsisInfo	Information supplementing the FuncDefs of a FuncSynopsis
Function	The name of a function or subroutine, as in a programming language
GUIButton	The text on a button in a GUI
GUIIcon	Graphic and/or text appearing as a icon in a GUI
GUILabel	The text of a label in a GUI
GUIMenu	The name of a menu in a GUI
GUIMenuItem	The name of a terminal menu item in a GUI
GUISubmenu	The name of a submenu in a GUI
GlossDef	A definition in a GlossEntry
GlossDiv	A division in a Glossary
GlossEntry	An entry in a Glossary or GlossList
GlossList	A wrapper for a set of GlossEntrys
GlossSee	A cross-reference from one GlossEntry to another
GlossSeeAlso	A cross-reference from one GlossEntry to another
GlossTerm	A glossary term
Glossary	A glossary
Graphic	A displayed graphical object (not an inline)
GraphicCO	A graphic that contains callout areas
Group	A group of elements in a CmdSynopsis
Hardware	A physical part of a computer system
Highlights	A summary of the main points of the discussed component
Holder	The name of the individual or organization that holds a copyright
Honorific	The title of a person
ISBN	The International Standard Book Number of a document
ISSN	The International Standard Serial Number of a periodical
ITermSet	A set of index terms in the meta-information of a document
ImageData	Pointer to external image data
ImageObject	A wrapper for image data and its associated meta-information
ImageObjectCO	A wrapper for an image object with callouts
Important	An admonition set off from the text
Index	An index
IndexDiv	A division in an index
IndexEntry	An entry in an index
IndexTerm	A wrapper for terms to be indexed
InformalEquation	A displayed mathematical equation without a title
InformalExample	A displayed example without a title
InformalFigure	A untitled figure
InformalTable	A table without a title
InlineEquation	A mathematical equation or expression occurring inline
InlineGraphic	An object containing or pointing to graphical data that will be rendered inline
InlineMediaObject	An inline media object (video, audio, image, and so on)

Element	Brief Description
Interface	An element of a GUI
InterfaceDefinition	The name of a formal specification of a GUI
InvPartNumber	An inventory part number
IssueNum	The number of an issue of a journal
ItemizedList	A list in which each entry is marked with a bullet or other dingbat
JobTitle	The title of an individual in an organization
KeyCap	The text printed on a key on a keyboard
KeyCode	The internal, frequently numeric, identifier for a key on a keyboard
KeyCombo	A combination of input actions
KeySym	The symbolic name of a key on a keyboard
Keyword	One of a set of keywords describing the content of a document
KeywordSet	A set of keywords describing the content of a document
LegalNotice	A statement of legal obligations or requirements
LineAnnotation	A comment on a line in a verbatim listing
Lineage	The portion of a person's name indicating a relationship to ancestors
Link	A hypertext link
ListItem	A wrapper for the elements of a list item
Literal	Inline text that is some literal value
LiteralLayout	A block of text in which line breaks and white space are to be reproduced faithfully
LoT	A list of the titles of formal objects (as tables or figures) in a document
LoTentry	An entry in a list of titles
ManVolNum	A reference volume number
Markup	A string of formatting markup in text that is to be represented literally
MediaLabel	A name that identifies the physical medium on which some information resides
MediaObject	A displayed media object (video, audio, image, etc.)
MediaObjectCO	A media object that contains callouts
Member	An element of a simple list
MenuChoice	A selection or series of selections from a menu
ModeSpec	Application-specific information necessary for the completion of an OLink
MouseButton	The conventional name of a mouse button
Msg	A message in a message set
MsgAud	The audience to which a message in a message set is relevant
MsgEntry	A wrapper for an entry in a message set
MsgExplan	Explanatory material relating to a message in a message set
MsgInfo	Information about a message in a message set
MsgLevel	The level of importance or severity of a message in a message set
MsgMain	The primary component of a message in a message set
MsgOrig	The origin of a message in a message set
MsgRel	A related component of a message in a message set
MsgSet	A detailed set of messages, usually error messages
MsgSub	A subcomponent of a message in a message set
MsgText	The actual text of a message component in a message set
Note	A message set off from the text
OLink	A link that addresses its target indirectly, through an entity
ObjectInfo	Meta-information for an object
Option	An option for a software command
Optional	Optional information
OrderedList	A list in which each entry is marked with a sequentially incremented label
OrgDiv	A division of an organization
OrgName	The name of an organization other than a corporation

Element	Brief Description
OtherAddr	Uncategorized information in address
OtherCredit	A person or entity, other than an author or editor, credited in a document
OtherName	A component of a persons name that is not a first name, surname, or lineage
POB	A post office box in an address
PageNums	The numbers of the pages in a book, for use in a bibliographic entry
Para	A paragraph
ParamDef	Information about a function parameter in a programming language
Parameter	A value or a symbolic reference to a value
Part	A division in a book
PartIntro	An introduction to the contents of a part
Phone	A telephone number
Phrase	A span of text
Postcode	A postal code in an address
Preface	Introductory matter preceding the first chapter of a book
Primary	The primary word or phrase under which an index term should be sorted
PrimaryIE	A primary term in an index entry, not in the text
PrintHistory	The printing history of a document
Procedure	A list of operations to be performed in a well-defined sequence
ProductName	The formal name of a product
ProductNumber	A number assigned to a product
ProgramListing	A literal listing of all or part of a program
ProgramListingCO	A program listing with associated areas used in callouts
Prompt	A character or string indicating the start of an input field in a computer display
Property	A unit of data associated with some part of a computer system
PubDate	The date of publication of a document
Publisher	The publisher of a document
PublisherName	The name of the publisher of a document
PubsNumber	A number assigned to a publication other than an ISBN or ISSN or inventory part number
QandADiv	A titled division in a QandASet
QandAEntry	A question/answer set within a QandASet
QandASet	A question-and-answer set
Question	A question in a QandASet
Quote	An inline quotation
RefClass	The scope or other indication of applicability of a reference entry
RefDescriptor	A description of the topic of a reference page
RefEntry	A reference page (originally a UNIX man-style reference page)
RefEntryTitle	The title of a reference page
RefMeta	Meta-information for a reference entry
RefMiscInfo	Meta-information for a reference entry other than the title and volume number
RefName	The name of (one of) the subject(s) of a reference page
RefNameDiv	The name, purpose, and classification of a reference page
RefPurpose	A short (one sentence) synopsis of the topic of a reference page
RefSect1	A major subsection of a reference entry
RefSect1Info	Meta-information for a RefSect1
RefSect2	A subsection of a RefSect1
RefSect2Info	Meta-information for a RefSect2
RefSect3	A subsection of a RefSect2
RefSect3Info	Meta-information for a RefSect3
RefSynopsisDiv	A syntactic synopsis of the subject of the reference page
RefSynopsisDivInfo	Meta-information for a RefSynopsisDiv

Element	Brief Description
Reference	A collection of reference entries
ReleaseInfo	Information about a particular release of a document
Replaceable	Content that may or must be replaced by the user
ReturnValue	The value returned by a function
RevHistory	A history of the revisions to a document
RevNumber	A document revision number
RevRemark	A description of a revision to a document
Revision	An entry describing a single revision in the history of the revisions to a document
Row	A row in a table
SBR	An explicit line break in a command synopsis
SGMLTag	A component of SGML markup
Screen	Text that a user sees or might see on a computer screen
ScreenCO	A screen with associated areas used in callouts
ScreenInfo	Information about how a screen shot was produced
ScreenShot	A representation of what the user sees or might see on a computer screen
Secondary	A secondary word or phrase in an index term
SecondaryIE	A secondary term in an index entry, rather than in the text
Sect1	A top-level section of document
Sect1Info	Meta-information for a Sect1
Sect2	A subsection within a Sect1
Sect2Info	Meta-information for a Sect2
Sect3	A subsection within a Sect2
Sect3Info	Meta-information for a Sect3
Sect4	A subsection within a Sect3
Sect4Info	Meta-information for a Sect4
Sect5	A subsection within a Sect4
Sect5Info	Meta-information for a Sect5
Section	A recursive section
SectionInfo	Meta-information for a recursive section
See	Part of an index term directing the reader instead to another entry in the index
SeeAlso	Part of an index term directing the reader also to another entry in the index
SeeAlsoIE	A “See also” entry in an index, rather than in the text
SeeIE	A “See” entry in an index, rather than in the text
Seg	An element of a list item in a segmented list
SegListItem	A list item in a segmented list
SegTitle	The title of an element of a list item in a segmented list
SegmentedList	A segmented list, a list of sets of elements
SeriesInfo	Information about the publication series of which a book is a part
SeriesVolNums	Numbers of the volumes in a series of books
Set	A collection of books
SetIndex	An index to a set of books
SetInfo	Meta-information for a Set
ShortAffil	A brief description of an affiliation
Shortcut	A key combination for an action that is also accessible through a menu
Sidebar	A portion of a document that is isolated from the main narrative flow
SimPara	A paragraph that contains only text and inline markup, no block elements
SimpleList	An undecorated list of single words or short phrases
SimpleSect	A section of a document with no subdivisions
SpanSpec	Formatting information for a spanned column in a table
State	A state or province in an address

Element	Brief Description
Step	A unit of action in a procedure
Street	A street address in an address
StructField	A field in a structure (in the programming language sense)
StructName	The name of a structure (in the programming language sense)
SubSteps	A wrapper for steps that occur within steps in a procedure
Subject	One of a group of terms describing the subject matter of a document
SubjectSet	A set of terms describing the subject matter of a document
SubjectTerm	A term in a group of terms describing the subject matter of a document
Subscript	A subscript (as in H_2O, the molecular formula for water)
Subtitle	The subtitle of a document
Superscript	A superscript (as in x^2, the mathematical notation for x multiplied by itself)
Surname	A family name; in western cultures the “last name”
Symbol	A name that is replaced by a value before processing
SynopFragment	A portion of a CmdSynopsis broken out from the main body of the synopsis
SynopFragmentRef	A reference to a fragment of a command synopsis
Synopsis	A general-purpose element for representing the syntax of commands or functions
SystemItem	A system-related item or term
TBody	A wrapper for the rows of a table or informal table
TFoot	A table footer consisting of one or more rows
TGroup	A wrapper for the main content of a table, or part of a table
THead	A table header consisting of one or more rows
Table	A formal table in a document
Term	The word or phrase being defined or described in a variable list
Tertiary	A tertiary word or phrase in an index term
TertiaryIE	A tertiary term in an index entry, rather than in the text
TextObject	A wrapper for a text description of an object and its associated meta-information
Tip	A suggestion to the user, set off from the text
Title	The text of the title of a section of a document or of a formal block-level element
TitleAbbrev	The abbreviation of a Title
ToC	A table of contents
ToCback	An entry in a table of contents for a back matter component
ToCchap	An entry in a table of contents for a component in the body of a document
ToCentry	A component title in a table of contents
ToCfront	An entry in a table of contents for a front matter component
ToClevel1	A top-level entry within a table of contents entry for a chapter-like component
ToClevel2	A second-level entry within a table of contents entry for a chapter-like component
ToClevel3	A third-level entry within a table of contents entry for a chapter-like component
ToClevel4	A fourth-level entry within a table of contents entry for a chapter-like component
ToClevel5	A fifth-level entry within a table of contents entry for a chapter-like component
ToCpart	An entry in a table of contents for a part of a book
Token	A unit of information
Trademark	A trademark
Type	The classification of a value

Element	Brief Description
ULink	A link that addresses its target by means of a URL (Uniform Resource Locator)
UserInput	Data entered by the user
VarArgs	An empty element in a function synopsis indicating a variable number of arguments
VarListEntry	A wrapper for a set of terms and the associated description in a variable list
VarName	The name of a variable
VariableList	A list in which each entry is composed of a set of one or more terms and an associated description
VideoData	Pointer to external video data
VideoObject	A wrapper for video data and its associated meta-information
Void	An empty element in a function synopsis indicating that the function in question takes no arguments
VolumeNum	The volume number of a document in a set (as of books in a set or articles in a journal)
Warning	An admonition set off from the text
WordAsWord	A word meant specifically as a word and not representing anything else
XRef	A cross reference to another part of the document
Year	The year of publication of a document

G.2 DocBook V4.1.2

All of the elements in DocBook V3.1 *except* ArtHeader, BookBiblio, Comment, DocInfo, InterfaceDefinition, SeriesInfo plus the following additional elements:

Element	Brief Description
appendixinfo	Meta-information for an Appendix
articleinfo	Meta-information for an Article
bibliographyinfo	Meta-information for a Bibliography
chapterinfo	Meta-information for a Chapter
classsynopsis	The syntax summary for a class definition
classsynopsisinfo	Information supplementing the contents of a ClassSynopsis
constructorsynopsis	A syntax summary for a constructor
destructorsynopsis	A syntax summary for a destructor
exceptionname	The name of an exception
fieldsynopsis	The name of a field in a class definition
glossaryinfo	Meta-information for a Glossary
indexinfo	Meta-information for an Index
initializer	The initializer for a FieldSynopsis
interfacename	The name of an interface
methodname	The name of a method
methodparam	Parameters to a method
methodsynopsis	A syntax summary for a method
modifier	Modifiers in a synopsis
ooclass	A class in an object-oriented programming language
ooexception	An exception in an object-oriented programming language
oointerface	An interface in an object-oriented programming language
partinfo	Meta-information for a Part
prefaceinfo	Meta-information for a Preface
refentryinfo	Meta-information for a Refentry
referenceinfo	Meta-information for a Reference

Element	Brief Description
remark	A remark (or comment) intended for presentation in a draft manuscript
revdescription	A extended description of a revision to a document
setindexinfo	Meta-information for a SetIndex
sidebarinfo	Meta-information for a Sidebar
simplemsgentry	A wrapper for a simpler entry in a message set

G.3 DocBook EBNF Module V1.0

The following elements in addition to the elements in the version of DocBook to which this module is added:

Element	Brief Description
constraint	A constraint in an EBNF production
constraintdef	The definition of a constraint in an EBNF production
lhs	The left-hand side of an EBNF production
nonterminal	A non-terminal in an EBNF production
production	A production in a set of EBNF productions
productionrecap	A cross-reference to an EBNF production
productionset	A set of EBNF productions
rhs	The right-hand side of an EBNF production

G.4 DocBook HTML Forms Module V1.0

The HTML Forms Module adds the HTML Form element and related elements.

G.5 DocBook MathML Module V1.0

The MathML Module adds the elements from the MathML namespace.

G.6 DocBook SVG Module V1.0

The SVG Module adds the elements from the SVG namespace.

Appendix H

GNU Free Documentation License

Version 1.1, March 2000

H.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

H.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup

has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

H.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

H.4 COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

H.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C State on the Title page the name of the publisher of the Modified Version, as the publisher.

D Preserve all the copyright notices of the Document.

E Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H Include an unaltered copy of this License.

I Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

H.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents,

make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

H.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

H.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

H.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

H.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

H.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

H.12 How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix I

ChangeLog

This appendix details the changes made between versions.

I.1 Changes since version 2.0.7 (2002-06-17)

```
Changes to en/*

  | 2002-12-29  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * ch00.xml, ch01.xml, ch05.xml: Patches from jrdaily
  |
  |   * ch04.xml: Patches from jrdaily: I'm attaching yet another
  |     revision of the patch. Updated changelog: * Use CDATA
  |     sections in screen and programlisting blocks containing
  |     sample SGML and XML code. * Put quotes around attribute
  |     values in one sample block. Some of the attributes are
  |     attached to <xref> tags; so far as I know, there is no way
  |     to represent an <xref> that is legal in both DocBook SGML
  |     and XML, so I left them SGML-compliant (without a closing
  |     /). * Replace XML with &XML; when not in index terms, and
  |     SGML with &SGML;. This only affects latter parts of the
  |     chapter; older portions already use those entities. *
  |     Replace <literal> with tags such as sgmltag, varname, and
  |     parameter. * Update the Emacs comments at the end of the
  |     document. * Replace &ldquo;/&rdquo; pairs with <quote> tags.
  |     * Eliminate &ldquo;/&rdquo; entities used following an
  |     <xref>; this may be an outdated stylesheet workaround. *
  |     Replace a quoted reference to a section with an <xref>
  |     (Scheme).
  |
  |   * ch04.xml: Fix typo
  |
  | 2002-09-21  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * ch02.xml: Fixed SGML minimization bug; not that it'll matter
  |     in 2e, I suppose
  |
  | 2002-06-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * ch02.xml: Add ID to Making an Index section
  |

Changes to en/stylesheets/*
```

```
  | 2002-12-29  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * tdg.xsl: Improve linking
  |

Changes to en/build/lib/*

  | 2002-12-29  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.pl: Attempt to support run-time prefix for output
  |     files. This will allow multiple versions of the book to be
  |     built. (This may not work yet)
  |
  |   * seealso: Fixed category title
  |

Changes to en/refpages/elements/bookinfo/*

  | 2002-09-25  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Defguide bug #613988: fix bug in copyright
  |     holders
  |

Changes to en/refpages/elements/callout/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Corrected typos and thinkos reported by Robert
  |     P. J. Day
  |

Changes to en/refpages/elements/co/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: New file.
  |

Changes to en/refpages/elements/emphasis/*

  | 2002-08-06  <dcramer@users.sourceforge.net>
  |
  |   * refentry.xml: Changed Emphasis in refentrytitle to emphasis
  |     to make it consistent with other refentrytitles
  |

Changes to en/refpages/elements/equation/*

  | 2002-09-21  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Try spelling Theorem correctly
  |

Changes to en/refpages/elements/errorcode/*

  | 2002-12-28  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Fix errorname usage
```

```
  |

Changes to en/refpages/elements/figure/*

  | 2002-09-21  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Use mediaobject instead of graphic
  |

Changes to en/refpages/elements/funcparams/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Fix typo
  |

Changes to en/refpages/elements/imagedata/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Add an ID so we can xref to a section
  |
  | 2002-09-21  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Specify scalefit=0 on example
  |

Changes to en/refpages/elements/index/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Corrected typos and thinkos reported by Robert
  |     P. J. Day
  |

Changes to en/refpages/elements/nonterminal/*

  | 2002-12-29  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Fix tracker #599968, clarify semantics of def
  |     attribute
  |

Changes to en/refpages/elements/olink/*

  | 2002-07-17  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Fix typo in entity declaration
  |

Changes to en/refpages/elements/qandaset/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Corrected typos and thinkos reported by Robert
  |     P. J. Day
  |

Changes to en/refpages/elements/screenco/*
```

```
  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Corrected typos and thinkos reported by
  |     Robert P. J. Day
  |

Changes to en/refpages/elements/simplemsgentry/*

  | 2002-12-28  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Remove reference to a future change that's
  |     already been made
  |

Changes to en/refpages/elements/synopsis/*

  | 2002-11-26  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * example.1.xml: Corrected typos and thinkos reported by
  |     Robert P. J. Day
  |

Changes to en/refpages/elements/xref/*

  | 2002-12-28  Norman Walsh <nwalsh@users.sourceforge.net>
  |
  |   * refentry.xml: Clarify processing expectations
  |
```

Glossary

attribute Attributes augment the element on which they appear; they also provide additional information about the element.

Attributes appear as name-value pairs in the element's start-tag. For example, to assign the value `hostname` to the Role attribute of `SystemItem`, you would use the mark up: `<systemitem role="hostname">`.

callout A pointer, verbal or graphical or both, to a *component* of an illustration or a text object.

cooked "Cooked" data, as distinct from "raw," is a collection of elements and character data that's ready for presentation. The processor is not expected to rearrange, select, or suppress any of the elements, but simply present them as specified. See also "Raw".

document type declaration (DTD) A set of declarations that defines the names of the elements and their attributes, and that specifies rules for their combination or sequence.

DSSSL Document Style Semantics and Specification Language (ISO/IEC 10179:1996). An international standard stylesheet language for SGML/XML documents.

element Elements define the hierarchical structure of a document. Most elements have start and end tags and contain some part of the document content. Empty elements have only a start tag and have no content.

entity A name assigned (by means of a declaration) to some chunk of data so it can be referred to by that name; the data can be of various kinds (a special character or a chapter or a set of declarations in a DTD, for instance), and the way in which it is referred to depends on the type of data and where it is being referenced: SGML has parameter, general, external, internal, and data entities.

exclusion An exclusion is used in a DTD to indicate that, within the element on which the exclusion occurs, the excluded elements are not valid anywhere within the content of the element.

For example, in DocBook, `Footnote` excludes `Footnote`. This means that footnotes cannot nest, even though `Footnote` contains `Para`, and `Footnote` occurs in the proper content model of `Para`.

See Section 1.4.2.1.

external entity An external entity is a general entity that refers to another document. External entities are often used to incorporate parsable text documents, like legal notices or chapters, into larger units, like chapters or books.

external subset Element, attribute, and other declarations that compose (part of) a document type definition that are stored in an external entity, and referenced from a document's document type declaration using a public or system identifier.

float Text objects like sidebars, figures, tables, and graphics are said to float when their actual place in the document is not fixed. For presentation on a printed page, for instance, a graphic may float to the top of the next page if it is too tall to fit on the page in which it actually falls, in the sequence of words and the sequence of other like objects in a document.

formal public identifier A public identifier that conforms to the specification of formal public identifers in ISO 8879.

FOSI Formatting Output Specification Instance, an SGML document that specifies the appearance or presentation of another SGML document in accordance with the Output Specification DTD defined by MIL-STD-28001C.

general entity An entity referenced by a name that starts with an ampersand (&) and ends with a semicolon. Most of the time general entities are used in SGML documents, not in the DTD. There are two types, external and internal entities, and they refer either to special characters or to text objects like commonly repeated phrases or names or chapters.

GI Generic identifier, proper term for the actual name of an element; `Para` is the generic identifier of the `para` element.

inclusion An inclusion is used in a DTD to indicate that, within the element on which the inclusion occurs, the included elements are valid anywhere within the content of the element.

For example, in DocBook, `Chapter` includes `IndexTerm`. This means that `IndexTerm`s can occur anywhere inside chapters, even inside elements that do not have `IndexTerm`s in their proper content models.

See Section 1.4.2.1.

internal entity A general entity that references a piece of text (including its markup and even other internal entities), usually as a keyboard shortcut.

internal subset Element, attribute, and other declarations that compose (part of) a document type definition that are stored in a document, within the document type declaration.

meta-information Meta-information is information about a document, such as the specification of its author or its date of composition, as opposed to the content of a document itself.

parameter entity An entity usually referenced in the DTD by a name that starts with a percent sign (%) and ends with a semicolon. In DocBook, parameter entities are mainly used to facilitate customization of the DTD, but they can also be used to control marked sections of a document.

processing instruction An essentially arbitrary string preceded by a question mark and delimited by angle brackets that is intended to convey information to an application that processes an SGML instance. For example, the processing instruction `<?linebreak>` might cause the formatter to introduce a line break at the position where the processing instruction occurs.

In XML documents, processing instructions should have the form:

```
<?pitarget param1="value1" param2="value2"?>
```

The *`pitarget`* should be a name that the processing application will recognize. Additional information in the PI should be added using “attribute syntax.”

public identifier An abstract identifier for an SGML or XML document, DTD, or external entity.

raw “Raw” data is just a collection of elements, with no additional punctation or information about presentation. To continue the cooking metaphor, raw data is just a set of ingredients. It’s up to the processor to select appropriate elements, arrange them for display, and add required presentational information. See also “Cooked”.

SGML Standard Generalized Markup Language, an international standard (ISO 8879) that specifies the rules for the creation of platform-independent markup languages for electronic texts.

stylesheet A file that specifies the presentation or appearance of a document; there are several standards for such stylesheets, including CSS, FOSIs, DSSSL, and, most recently, XSL. Vendors often have proprietary stylesheet formats as well.

system identifier In SGML, a local, system-dependent identifier for a document, DTD, or external entity. Usually a filename on the local system.

In XML, a system identifer is required to be a URI.

tag An SGML element name enclosed in angle brackets (<>), used to mark up the semantics or structure of a document. `<Para>` is a tag in DocBook used to mark the beginning of a paragraph.

URI Uniform Resource Identifier, the W3C’s codification of the name and address syntax of present and future objects on the Internet. In its most basic form, a URI consists of a scheme name (such as file, http, ftp, news, mailto, gopher) followed by a colon, followed by a path whose nature is determined by the scheme that precedes it (see RFC 1630).

URI is the umbrella term for URNs, URLs, and all other Uniform Resource Identifiers.

URL Uniform Resource Locator, a name and address for an existing object accessible over the Internet. `http://www.docbook.org` is an example of a URL (see RFC 1738).

URN Uniform Resource Name, the result of an evolving attempt to define a name and address syntax for *persistent* objects accessible over the Internet; `urn:foo:a123,456` is a legal URN consisting of three colon-separated fields: `urn` followed by a namespace identifier, followed by a namespace specifier (see RFC 1737 and RFC 2141 for details).

W3C The World Wide Web Consortium (`http://www.w3.org/`).

wrapper Some elements, such as `Chapter`, have important semantic significance. Other elements serve no obvious purpose except to contain a number of other elements. For example, `BookInfo` has no important semantics; it merely serves as a container for the meta-information about a book. Elements that are just containers are sometimes called "wrappers."

XML The Extensible Markup Language `http://www.w3.org/TR/REC-xml`, a subset of SGML designed specifically for use over the Web.

XSL XML Style Language, an evolving language for stylesheets to be attached to XML documents. The stylesheet is itself an XML document.