



Position Paper: DocBook for Internal Documentation

Proposal 01, 04 March 2003

Document identifier:

draft-stuhec-internalDocBook-01.doc

Location:

Author:

Gunther Stuhec <gunther.stuhec@sap.com>

Abstract:

This position paper outlines the use and definition the simplified docbook specification for the internal documentation of core components.

Status:

This is V01 of the identifier position paper intended for consideration by the OASIS UBL Naming and Design Rules subcommittee and other interested parties.

If you are on the ubl-ndrsc@lists.oasis-open.org list for subcommittee members, send comments there. If you are not on that list, subscribe to the ubl-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to ubl-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

Table of Contents

1	Introduction	3
2	Structure of Documentation	4
3	Simplified DocBook Definition	5
3.1	UBL subset of DocBook	5
3.2	Required Elements	6
3.2.1	Main structure	6
3.2.2	Article Information	6
3.2.3	Sectioning of Chapters	8
3.2.4	Title and Subtitle	9
3.2.5	Paragraphs	10
3.2.6	Listings	11
3.2.7	Note	12
3.2.8	Programlisting and Example	12
3.2.9	Figure	12
3.2.10	Table	13
4	Reference Page	16
5	Example	18
	ICC_ Terms. Details	18
	<i>Status / Owner</i>	18
	<i>History</i>	18
	<i>Definition</i>	18
	<i>Usage</i>	18
	<i>Structure</i>	18
	<i>Description</i>	19
	<i>Conditions of Integrity</i>	19
	<i>Technical Comments</i>	19
	<i>Example (Instance)</i>	19
	Appendix A. Bibliography	20
	Appendix B. Notes	21

1 Introduction

Documentation shall be embedded directly in the schema modules. That means that the internal documentation is information enclosed in `<xsd:documentation>` within `<xsd:annotation>` to telling the reader the details or information about the definition, usage and structure of each core component or business information entity.

This section describes the use of the `<xsd:documentation>` element for providing the internal documentation. Other sections address the issues around what the content of documentation should be at various points within the schema module, within the parameters described here.

UBL requires the use of embedded documentation for two reasons:

- Ease of maintenance – the file containing the module schema also contains the documentation, allowing comparison at a glance, and editing within a single screen.
- Ease of use – especially in auto-generation scenarios, having the structural content described by the schema instance itself and the relevant human-readable documentation makes the auto-generation of documentation much easier.

2 Structure of Documentation

Each internal documentation of core component or business information entity should have a <xsd:documentation> with the following information:

- Dictionary Entry Name – Is the title of each internal documentation and represents the complete notation of core component or business information entity according the naming conventions of UN/CEFACT ebXML Core Component Technical Specification.
- Developer's Name – Complete name of the developer (including the email-address)
- Definition – Definition of a core component or business information entity in short and easy understandable sentences.
- Publication Date – Date of publication of the core component or business information entity
- Version – Current version of the core component or business information entity
- Change History – History of changes and short description of the changes in content that have been made to the previous version. This should enable the reader to recognize changes without having to search through the whole document
- Example (Instance) – Representation of at least one example of XML instance of the specific core component or business information entity
- Structure – Detailed table of the structure, terms and properties of the elements, subelements or attributes within the structure.
- Description of substructure and content – Detailed description what each element or attribute does and how it is used
 - Conditions of integrity – Explain the condition of integrity both within the structure and in relationship to the other core components or business information entities. It could be showed by different types of structural diagrams, behaviour diagrams or model management diagrams.
 - Usage – Real cases, where each core component or business information entity can be used.
 - Technical Comments – Specific notes to the usage or structure of core components or business information entity.

3 Simplified DocBook Definition

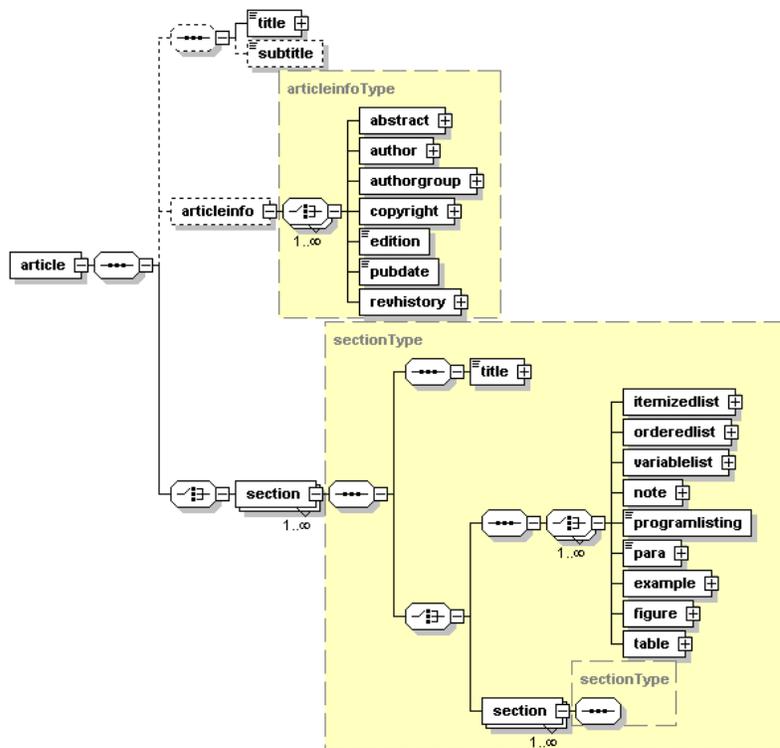
The DocBook markup language, maintained by the OASIS consortium, is specifically suited for technical documentation. It provides a rich set of tags to describe the content of especially software documentation. And the Simplified Docbook is a small subset of the DocBook markup language.

The simplified DocBook will be used for single documents like the internal documentation itself. It has a sufficient element set for it. It is although freely available as the DocBook markup language. And it is based for 100% on the DocBook markup language. That means that all elements well documented, and comes with a good set of production tools.

Simplified DocBook is recommended for a number of reasons -- it's a very verbose schema, it covers almost every possible requirement for technical and other forms of documentation, and it is quickly becoming the standard schema for open-source and open-content information. There are also a number of open-source tools available for processing DocBook instances, some of which work quite well. Additionally, Norman Walsh's modular DocBook stylesheets are freely available, very complete, and under active development.

3.1 UBL subset of DocBook

The the UBL compliant DocBook schema could be a subset of the simplified DocBook. Because, that not all elements are necessary for the internal documentation. The following structure shows the UBL compliant subset of the the simplified DocBook schema:



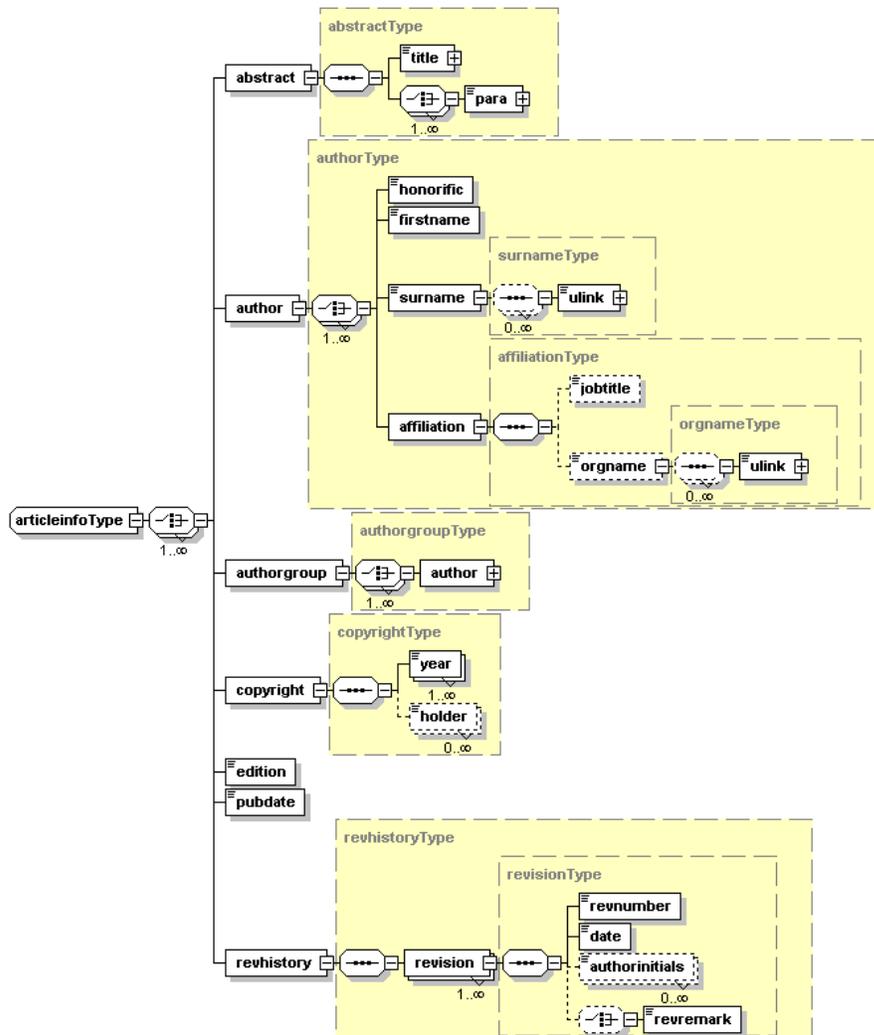
3.2 Required Elements

3.2.1 Main structure

- article - The Article element is the only one root element. It 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.
- title – Title is the dictionary entry of a section of a document or of a formal block-level element.
- subtitle – Represents the subtitle of the core component or business information entity and can be used for the UBL name.
- articleinfo – This is the metainformation about each core component or business information entity.
- section – This is the top-level sectioning of each chapter. Following chapters are recommended in the following order:
 - Usage
 - Structure
 - Description
 - Conditions of Integrity
 - Technical Comments
 - Examples

3.2.2 Article Information

Article information will be represented by the elementgroup “articleinfo” and includes the metainformation of each core component or business information entity.



This metainformation are described by the following individual elements:

- abstract – Abstract includes the definition of each core component or business information entity. It includes the title and the para as mandatory subelements. The title itself shall have allways the word “Defintion”. The elementgroup para must be used for the definition text itself.
- author – Author includes the primary author of the core component or business information entity. The author includes the following elements:
 - email – for the email address of the author (does not exit in the simplified docbook),
 - honorific – for the title of an author,
 - firstname – for the given name of an author,
 - surname – for the family- or lastname of an author,
 - affiliation – for the institutional affiliation of an author. This affiliation includes the elements “jobtitle” for the title of the job in an organization and “orgname” for the

name of the organization, where the author works for. The organization can have a hyperlink by the child element “ulink”.

- **authorgroup** – This is the wrapper of author information when a core component or business information entity has multiple authors or some another authors made some additional revisions. The authorgroup can have on ore more child elements “author”, which are based on the same structure of author as described above.
- **copyright** – It may be holds the information about the date and holder of a copyright, if the specific core component or busines information entity does have a copyright.
- **edition** – This is the current version number of the edition of a core component or business information entity.
- **pubdate** – This is the date of the publication of a core component or business information entity.
- **revhistory** – This element group includes the revisions to the core components, business information entities or its internal documentation. The revhistory can have on or more elements “revision” for the information about changes of the specifitc core component, business information entity or document. The element “revision” includes the child elements “revnumber”, “date”, “authorinitials” and “revremark” for it. The remarks for the changes can be done by the element “revremark”.

3.2.3 Sectioning of Chapters

Each chapter of the body of internal documentation are separated by the elementgroup “section”. This elementgroup has the following subchilds:

- **title** – Title of each main chapter (or subchapters). The following titles of main chapters are recommended:
 - Usage
 - Structure
 - Description
 - Conditions of Integrity
 - Technical Comments
 - Examples

Each chapter can has at least on of the following child elements:

- **itemizedlist** – A list in which each entry is marked by a bullet. “itemizedlist” should be used for the detailed description of each subelement or attribute of a core component or business information entity.
- **orderedlist** – A list in which each entry is marked with a sequentially incremenented label, like numeral or letter.
- **variablelist** – A list in which each entry is componesd of a set of one term and an associated description of this term. The variablelist could be used for enumerations and the description of enumerations.
- **note** – A note is an explanatory information of the chapter.

- programlisting – A literal listing of all or a part of the XML schema.
- para – A para represents a paragraph of a text.
- example – A formal (with title) example of the XML instance
- figure – A formal (with title) figure, which will be illustrated by a mediaobject.
- table – A formal (with title) table for describing the structure or usage.
- section – The recursive section can be used for some subsections for more detailed information.

The child elements, which are described above, should be used in the following chapters:

- Usage – itemizedlist, orderedlist, note, para, figure, table, section
- Structure – table, note
- Description – itemizedlist, orderedlist, variablelist, note, para, table, section
- Conditions of Integrity – itemizedlist, orderedlist, note, programlisting, para, figure, table, section
- Technical Comments – itemizedlist, orderedlist, note, programlisting, para, figure, table, section
- Examples – note, para, example

3.2.4 Title and Subtitle

The “title” identifies the titles of

- article (mandatory) – Identifies the dictionary entry name of each core component or business information entity.
- abstract (mandatory) – The title of abstract shall have always “Definiton”.
- section (mandatory) – Represents one the following headers on the first level:
 - Usage
 - Structure
 - Description
 - Conditions of Integrity
 - Technical Comments
 - Examples
- itemizedlist (optional) – May be identifies the name of the itemizedlist.
- orderedlist (optional) – May be identifies the name of the orderedlist.
- variablelist (optional) – May be identifies the name of the variablelist.
- note (optional) – Gives optionally the note a header name.

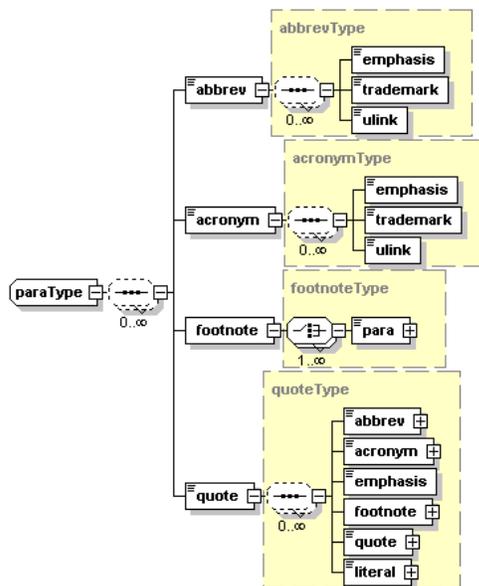
- example (optional) – Gives optionally the example a header name.
- figure (mandatory) – Every figure must have a title.
- table (mandatory) – Every formalized table must have a title.

The title itself can have a footnote. It represents a mark and a body of an additional note, which can be presented at the bottom of a viewed or printed page.

The “subtitle” is for the subtitle of an article. It may be represents the XML based tag name of the equivalent dictionary entry name.

3.2.5 Paragraphs

The element “para” which will be almost used for the paragraphs in all different sections and in the entries of the different tables. The para may have a mixed content. The content can be a string and some further elements within this strings. This elements can be used for abbreviations, acronyms, footnotes and quotes. The “para” be based on the following structure:



The child elements of “para” can be:

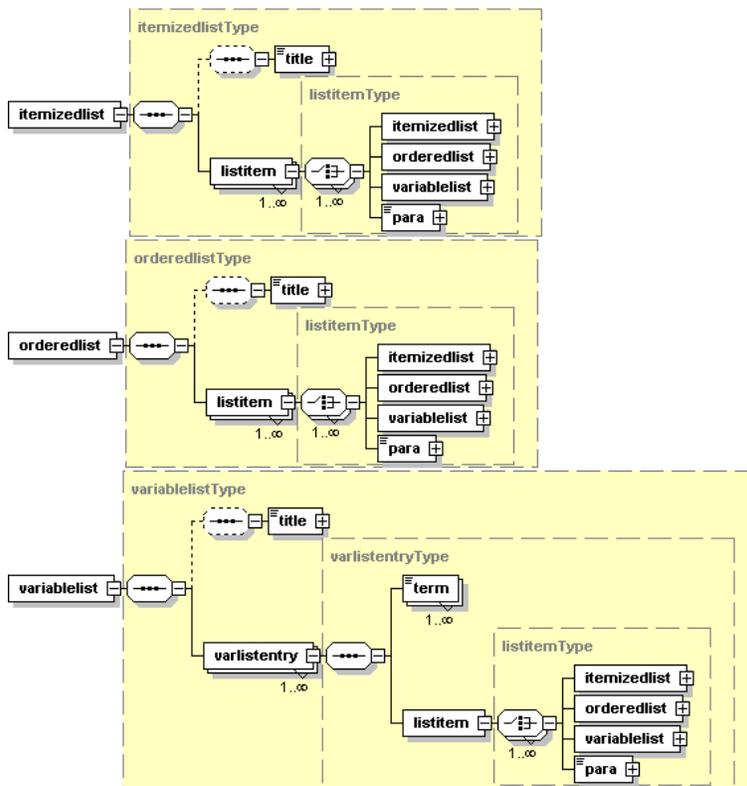
- abbrev – This element can be used for an abbreviation. This abbreviation may be expressed in an emphasis text by using the element “emphasis”. A trademark can be expressed by the element “trademark”. Additionally, a hyperlink can be represented by the element “uLink”.
- acronym – This element can be used for an often pronounceable word made from the initial (or selected) characters of a name or phrase. This acronym can be expressed in an emphasis text and can have include an additional trademark. Although, the hyperlink can be expressed by using the element “uLink”.
- footnote – Every paragraph can have a footnote by using the element “footnote”.
- quote – Specific text can be represented in an inline quotation. The element 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. The abbreviations, acronyms, emphasis, literals and other quotations can be expressed in the the inline quotations.

3.2.6 Listings

There existing three different types of listings:

- itemizedlist – A list in which each entry is marked by a bullet. “itemizedlist” should be used for the detailed description of each subelement or attribute of a core component or business information entity.
- orderedlist – A list in which each entry is marked with a sequentially incremented label, like numeral or letter.
- variablelist – A list in which each entry is composed of a set of one term and an associated description of this term. The variablelist could be used for enumerations and the description of enumerations.

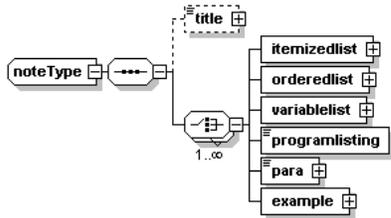


All three different types of lists can have optionally a title. The elements itemizedlist and orderedlist do have the child element “listitem” directly. This listitem is a wrapper around an item in a list and can occur on or more times within that list. This listitem can have a choice of one ore more paragraphs with the element “para”, or it can have one ore more another sub lists, which based on itemizedlist, orderlist or variablelist. The variablelist contains on ore more varlistentry as an immediate child element. This varlistentry may has on or more terms (element term) for a word

or phrase being defined or described in the variable list and on or more listitems, which has the same structure as in “itemizedlist” and “orderedlist”.

3.2.7 Note

The note can be used for a short notice about definition, description, usage, integrity or technical comments.

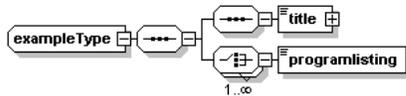


The note may be have a title and can have on or more following child elements:

- Listings: itemizedlist, orderedlist, variablelist
- Paragraphs with para
- Programlisting which includes XML-Schema and example which includes XML-Instances.

3.2.8 Programlisting and Example

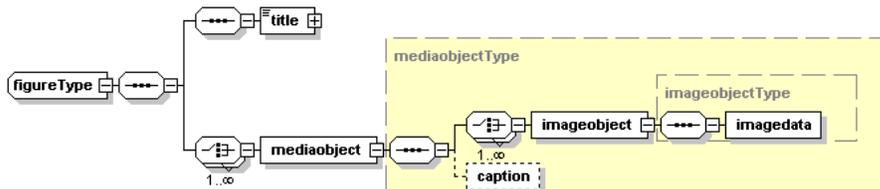
The programlisting may be occur within the elementgroup note or must occur at least on time in the elementgroup example.



The programlisting is a literal of listing all or a part of XML based structure. If programlisting occurs as a child in para directly, it should be used for the representation of a part of the UBL based XML schema of the specific core component or business information entity. The programlisting within example should be used for XML instances. Each example can have one or more programlisting and must have a title for identifying the title of the XML instance.

3.2.9 Figure

The figure is an illustration with a title, which can be used for a



3.2.10 Table

The table should be used for the structural information of ACCs and BCCs or ABIEs and BBIEs. But it can also be used for another tableized information of:

- Description
- Conditions of Integrity and
- Technical Comments

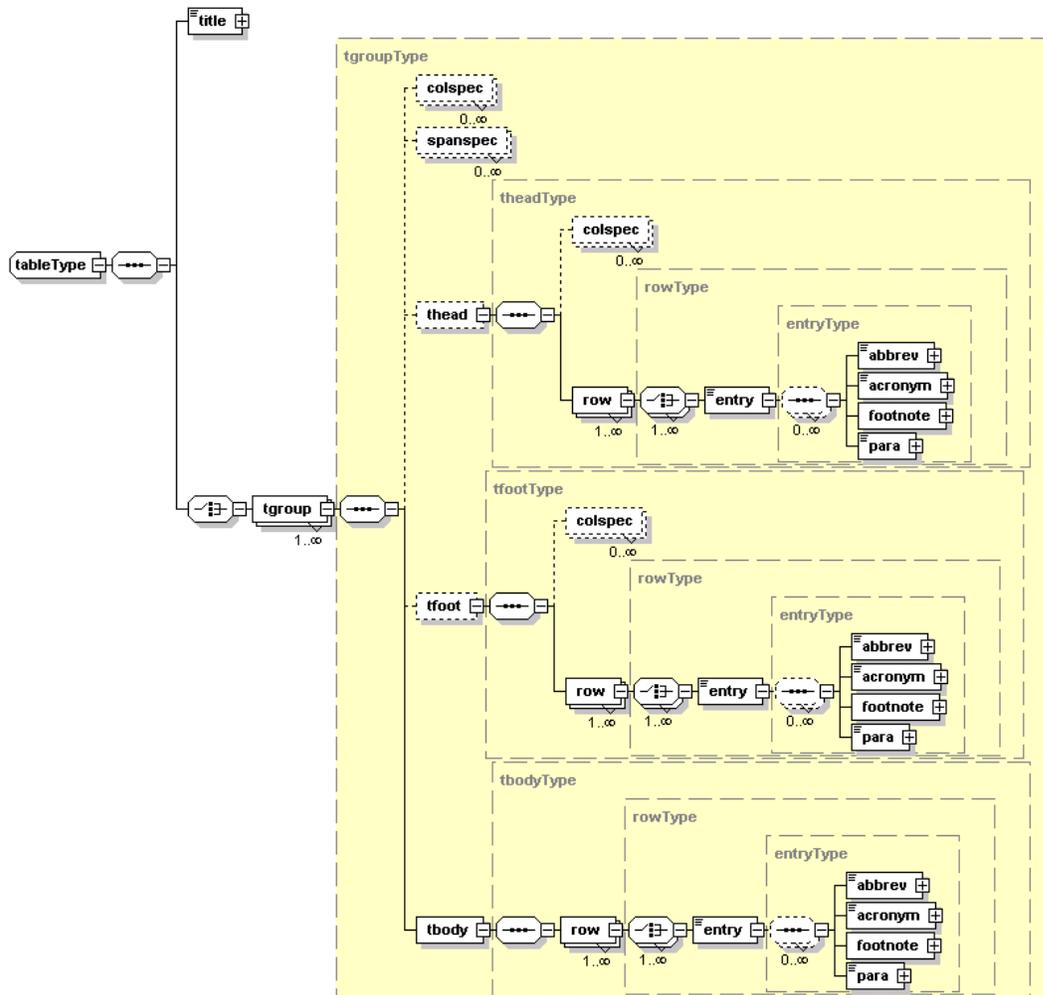
The table structure of ABIE/BBIEs should have be the following representation:

ABIE / BBIE		Cat.	Object Class	Property	Repr. / Assoc.	Type	Len.	Card.	Remarks
<Name of ABIE>									
	<Name of BBIE or ASBIE>								
	...								

The table structure of ACC/BCCs should have be the following representation:

ACC / BCC		Cat.	Object Class Qual.	Object Class	Property Qual.	Property	Repr. / Assoc. Qual.	Repr. / Assoc.	Type	Len.	Card.	Remarks
<Name of ACC>												
	<Name of BCC or ASCC>											
	...											

For the representation of the complete table structure, the following child elements are necessary:



- title – The name of each table.
- tgroup – The wrapper of the content of each table.

The element tgroup specifies a number of columns and contains all the header, body and footer rows, along with any additional column or span specifications necessary to express the geometry of the table. For defining the characteristics, columns and rows are the following will be available:

- colspec – This element includes attributes to specify the presentation characteristics of entries in a column of a table.
- spanspec – This element includes attributes for formatting information for a spanned column in a table.
- thead – This parent element defines the header of a table, which can consist of one or more rows.
- tfoot – This parent element represents the footer, which can consist of one or more rows.
- tbody – This parent element is a wrapper for the rows of a table that form the body of the table.

The element groups “thead”, “tfoot” and “tbody” have nearly the similar substructure. This element groups includes the following subelements:

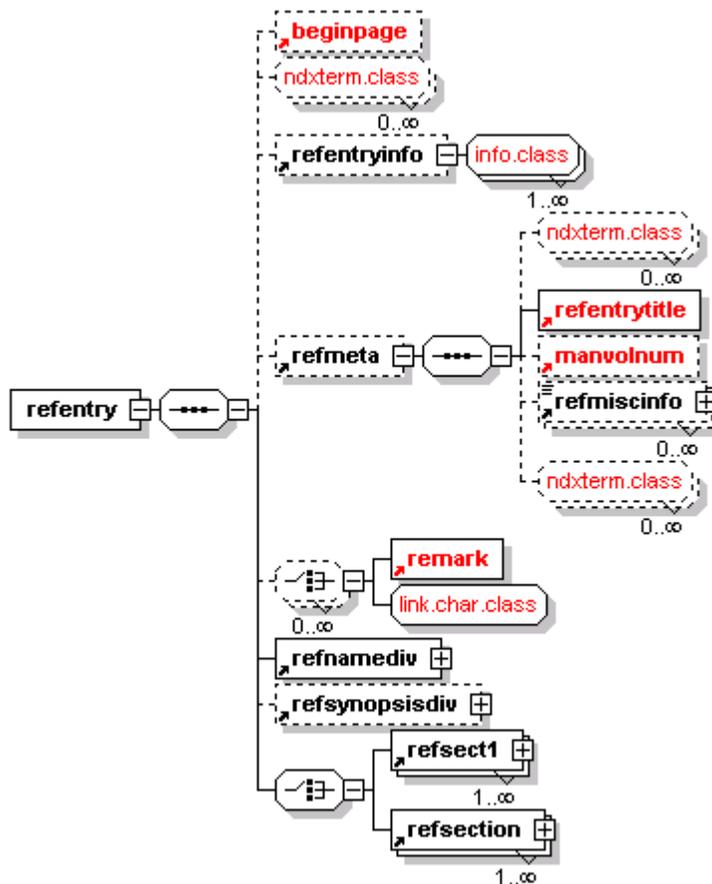
- colspec – This element will be used in “thead” and “tfoot” only. It includes attributes to specify the presentation characteristics of header or footer entries in a column of a table.
- row – Each of this element represents a row of a body, footer or header. The row can have on ore more entries. This entries can have one ore more paragraphs (para), abbreviations or acronyms. Each entry can have an additionally footnote.

4 Reference Page

The documentation of the core components or business information entities are not normally an article. This documentation can be compared with the common UNIX “manpage”. But the problem is that this manpage can not be represented by using the simplified docbook. If this happens so, that we’re preferring the manpage structure, it is more useful to use the “reference page” of the complete “docbook” structure.

The reference page or manual page in DocBook was inspired by, and in fact designed to reproduce, the common UNIX “manpage” concept. 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.

The structure of reference page is roughly:



In the sequence above, the elements that may appear in a RefEntry, only two are obligatory: RefNameDiv and RefSect1.

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.

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, there are only three levels of sectioning elements: RefSect1, RefSect2, and RefSect3.

5 Example

The following example shows a short output of a documentation of an aggregate core components.

ICC_Terms. Details

Status / Owner

Owner: X, Y

Version: 1.0

Date: March 4, 2003

History

Definition

ICCTermsDetails are commercial terms on terms of delivery that conform to the rules set down by the International Chamber of Commerce (ICC).

Usage

InternationalCommerceTermsDetails is used for defining the terms of delivery that the business partners have agreed for the transmission of a purchase order.

Structure

“InternationalCommerceTermsDetails” is made up of the child elements ClassificationCode from Core Component Type “Code” and Location name from Core Component Type “Text”.

GDT		Category	Object Class	Property	Representation	Type	Length	Cardinality
ICCDetails		complexType	ICC	Details				
	ClassificationCode	element	InternationalCommerceTerms	Classification	Code	xsd:token	3	1
	TransferLocationName	element	InternationalCommerceTerms	TransferLocation	Name	xsd:string	1..28	0..1

Description

ClassificationCode: Coded display of the abbreviations used internationally for characterizing terms of delivery

TransferLocationName: Location (place, port of shipment, port of destination, place of destination) to which the code above refers, for example, the port of shipment for FOB.

ICCTermsDetails is a three-digit character field and can have the following values (based on UN/CEFACT DE 4053, or derived from INCOTERMS):

- EXW – Ex Works
- FCA – Free Carrier
- FAS – Free Alongside Ship
- FOB – Free on Board
- CFR – Cost & Freight
- CIF – Cost, Insurance & Freight to named destination
- CPT – Freight, Carriage paid to destination
- CIP – Freight, Carriage, Insurance to destination
- DAF – Delivery at frontier – Named place
- DES – Delivered Ex Ship – Named port of destination
- DEQ – Delivered Ex Quay – Duty paid, Named port
- DDU – Delivered duty unpaid to destination
- DDP – Delivered duty paid to destination

Conditions of Integrity

<Diagram>

Technical Comments

Example (Instance)

```
<ICCTerms>  
  <ClassificationCode>FOB</ClassificationCode >  
  <LocationName>Hamburg</LocationName>  
</ICCTerms >
```

Appendix A. Bibliography

Appendix B. Notes

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

Copyright © The Organization for the Advancement of Structured Information Standards [OASIS] 2001. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself does not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.