OASIS Committee Note

OASIS Specification Publishing in DocBook XML Version 0.8

Working Draft 09

03 September 2020

This stage:
> https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.html (Authoritative)
> https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.pdf
> https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.xml

Previous stage:
> N/A

Latest stage:
> N/A

Technical Committee:
> OASIS TC Administration

Editor:
> G. Ken Holman (gkholman@CraneSoftwrights.com), Crane Softwrights Ltd.

Additional artifacts:
> This prose specification is one component of a Work Product which also includes:

> • publishing materials: https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.zip

Related work:
> This methodology supersedes guidelines for the DocBook XML-based authoring and publishing of OASIS specifications described in https://docs.oasis-open.org/templates/DocBook/spec-0.5/oasis-specification.html.

Abstract:
> This working draft describes an environment for writing and publishing an OASIS specification or an OASIS note using DocBook XML.

Status:
> This is a work in progress contributed to the OASIS TC administration and does not at this time represent the consensus of any particular OASIS Technical Committee. There are no plans to make this a formal Committee Specification as it is merely an internal document made available to committee members to support the publishing process.

Citation format:
> When referencing this specification the following citation format should be used:

[**OASIS-DOCBOOK-TEMPLATE-V0.8**] *OASIS DocBook Template V0.8.* 03 September 2020. OASIS Working Draft 09. https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.html.

Notices

Table of Contents

Appendixes

1 Introduction

1.1 Overview

This document details an environment [**Spec-0.8-ZIP**] and methodology for writing an OASIS specification or an OASIS note document using DocBook XML markup, and publishing the resulting document to HTML and PDF results conforming to OASIS layout conventions.

These stylesheets use XSLT [**XSLT**] as the transformation expression language to produce the final HTML result. XSLT is also used to produce the intermediate XSL-FO [**XSL-FO**] pagination semantics, which in turn are available to be processed to produce printable results as a PDF file using an appropriate processor (not included).

1.2 Terminology

1.2.1 Key words

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

1.3 References

[**ant**] Apache Software Foundation *Ant Java-based Build Tool* (Another Neat Tool).

[**DocBook**] OASIS Committee Draft 4.5 *The DocBook 4.5 Document type*, OASIS 01 October 2006, *OASIS DocBook 4.5 Document Type 01 October 2006 HTML,* , *DocBook 4.5 XML DTD,* , *DocBook Stylesheets Version 1.79.1*.

[**FOP**] Apache Software Foundation *Formatting Objects Processor*.

[**Ibex Signature Edition**] Visual Programming Limited. *Ibex XSL-FO Processor*.

[**MathML**] W3C *MathML DTD*.

[**Saxon**] Michael Kay *Saxon 6.5.5 XSLT 1.0 Processor*.

[**Spec-0.8-ZIP**] *OASIS Specification Publishing in DocBook XML Version 0.8. ZIP package.* 03 September 2020. *https://docs.oasis-open.org/templates/DocBook/spec-0.8/oasis-specification-0.8-wd09.zip*

[**RFC 2119**] *Key words for use in RFCs to Indicate Requirement Levels*, March 1997. S. Bradner. IETF (Internet Engineering Task Force) RFC 2119, *http://www.ietf.org/rfc/rfc2119.txt*

[**xjparse**] Norm Walsh *xjparse XML validation invocation*.

[**xml-assoc**] *Associating Style Sheets with XML documents Version 1.0.* 29 June 1999. James Clark. W3C Recommendation. *http://www.w3.org/1999/06/REC-xml-stylesheet-19990629*

[**XSL-FO**] *Extensible Stylesheet Language (XSL) Version 1.1.* 5 December 2006. Anders Berglund. W3C Recommendation. *http://www.w3.org/TR/2001/REC-xsl-20011015/*

[**XSLT**] *XSL Transformations (XSLT) Version 1.0.* 16 November 1999. James Clark. W3C Recommendation. *http://www.w3.org/TR/1999/REC-xslt-19991116*

2 Installation and use

2.1 Installation overview

These stylesheets are zipped [**Spec-0.8-ZIP**] in a turnkey environment ready to unpack for offline use. The environment is used to validate DocBook XML, render HTML and create XSL-FO suitable for processing with an XSL-FO engine (which is not included). Of course one is not obliged to use the processors referenced as one can use any XML DTD validation tool and any XSLT processor.

The package can be downloaded from https://docs.oasis-open.org/templates/DocBook/spec-0.8/ oasis-specification-0.8-wd09.zip and is expected to be unzipped into a local directory for offline use. The examples in this documentation assume the package is unzipped in a Windows environment into the `c:\` directory, and in a Linux environment in one's home directory. For example purposes, it is assumed that the ZIP file contents (all files starting with `spec-0.8/`) are installed in the `c:\oasis\spec-0.8\` directory or `~/oasis/spec-0.8/`.

See Appendix C, *Publishing choreography and orchestration (Non-Normative)* for details on a set of invocations for validating OASIS specification documents and producing an HTML rendering suitable for web browsers and an XSL-FO result suitable for an XSL-FO engine. Neither a browser nor an XSL-FO engine are themselves included in the package.

2.2 Package contents

2.2.1 Package directories

The significant directories of the package are as follows, shown for a Windows environment but identical in a Linux environment:

```
c:\oasis\  - base directory for complete installation
  \spec-0.8                 - base directory for rendering
  \spec-0.8\css             - HTML appearance
  \spec-0.8\stylesheets     - HTML and XSL-FO production stylesheets
  \spec-0.8\template        - example specification and example note in XML
  \spec-0.8\docbook         - base directory for DocBook DTD
  \spec-0.8\docbook\xsl      - base directory for stylesheet library
  \spec-0.8\docbook\xsl\fo   - base directory for XSL-FO library
  \spec-0.8\docbook\xsl\html - base directory for HTML library
  \spec-0.8\htmlruntime     - embedded installation in a distribution
  \spec-0.8\mathml2         - base directory for MathML2 DTD
  \spec-0.8\mathml3         - base directory for MathML3 DTD
  \spec-0.8\validation      - example validation processes
```

The DocBook DTD components in the `docbook/` directory were obtained from http://www.docbook.org/xml/4.5/docbook-xml-4.5.zip.

The DocBook XSL components in the `docbook/xsl/` directory were obtained from http://sourceforge.net/projects/docbook/files/docbook-xsl-doc/docbook-xsl-1.69.1.zip, where the base directory of the unzipped package is renamed "`xsl/`". This allows one to obtain another release of the DocBook stylesheets, unzip them into a directory, rename the base directory and replace the `xsl/` installation directory (though this is not necessary as the distributed version satisfies the requirements set out by OASIS TC Administration). The `xsl/fo/` and `xsl/html/` directories should be in place when the unpacking, renaming and replacing is completed. Note that three installation files `install.sh`, `.CatalogManager.properties.example`, and `.urilist` have been removed from the 1.69.1 directories per the instructions in the `INSTALL` file, as two of them violate OASIS file naming guidelines.

2.2.2 Runtime package contents

The `htmlruntime/` directory is a suitable subset of the other directories for use when distributing the HTML subset of this publishing environment embedded in your specification distribution for runtime rendering of XML documents in HTML. The directory name "`htmlruntime`" can be renamed as desired to be anything at all. The XML of the specification uses stylesheet association (see Section 3.7, " Stylesheet association") to point to the HTML stylesheet in the runtime package directory distributed with the XML.

An example of the use of this runtime configuration is in http://docs.oasis-open.org/ubl/os-UBL-2.2/ where the file `UBL-2.2.xml` includes a stylesheet association directive to engage the HTML stylesheets in the "`db/`" directory (having renamed it from "`htmlruntime/`").

> **Note**
>
> This `UBL-2.2.xml` also illustrates a number of features including the use of the `condition=` attribute documented in Section 3.3, "DocBook content filtering"

2.3 Engaging the DocBook DTD

The DocBook Document Type Definition (DTD) expresses the constraints of using elements and attributes, but not necessarily the conventions imposed by the OASIS TC Administration. Just having your specification validate with the DTD does not guarantee that you've used the correct or appropriate DocBook constructs to produce the required output. This package includes Section 3.5, " Pro forma templates" illustrating the conventional use of the DocBook constructs.

The catalog.xml OASIS Catalog will resolve DocBook and MathML DTD public identifiers to the appropriate DTD fragments.

To assert in the online OASIS environment that your XML document satisfies the constraints of DocBook, point to the posted copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd"
```

To assert in an embedded environment that your XML document satisfies the constraints of DocBook, point to the embedded copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "htmlruntime/spec-0.8/docbook/docbookx.dtd"
```

To assert in an offline local environment that your XML document satisfies the constraints of DocBook, point to the installation copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "file:///c:/oasis/spec-0.8/docbook/docbookx.dtd"
```

or

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "file:///Users/admin/z/oasis/spec-0.8/docbook/docbookx.dtd"
```

2.4 The OASIS HTML stylesheets

There are four HTML stylesheets in the package:

- `oasis-specification-html-offline.xsl`

- `oasis-specification-html.xsl`

- `oasis-note-html-offline.xsl`

- `oasis-note-html.xsl`

For each type of OASIS document (specification and note) there are two stylesheets you must choose from in order to create one of two versions of the rendering based on how that rendering will be used: online or offline.

The online stylesheets (without a suffix) are used to create final online versions to send to OASIS TC Administration (this will use embedded references to online CSS stylesheets and the online base URI in accordance with OASIS TC Administration requirements). Please review the results with TC Administration in case any rules or procedures have changed.

The offline stylesheets are used for creating artefacts that work self-contained within the publishing environment, without making any references to the OASIS web site. By default, these self-contained results will use absolute URI references to the CSS stylesheet and the OASIS logo, which work fine in the publishing environment but are not portable offline to other environments that do not support the absolute URI values.

To ensure portability across all environments, a top-level processing instruction is added to the specification that directs the creation of relative URI references to the CSS stylesheet and OASIS logo (and must be adjusted to accommodate the target deployment environment):

```
<?oasis-spec-base-uri htmlruntime/spec-0.8/?>
```

For an OASIS specification, use this stylesheet during development and local review to create a local instance (this will use the local base URI, either absolute or relative based on the presence of the processing instruction):

```
c:\oasis\spec-0.8\stylesheets\oasis-specification-html-offline.xsl
```

For an OASIS specification, use this stylesheet in the offline environment to create the final online version to send to OASIS TC Administration (this will use embedded references to online CSS stylesheets and the online base URI in accordance with OASIS TC Administration requirements):

```
c:\oasis\spec-0.8\stylesheets\oasis-specification-html.xsl
```

For an OASIS note, use this stylesheet during development and local review to create a local instance (this will use the local base URI, either absolute or relative based on the presence of the processing instruction):

```
c:\oasis\spec-0.8\stylesheets\oasis-note-html-offline.xsl
```

For an OASIS note, use this stylesheet in the offline environment to create the final online version:

```
c:\oasis\spec-0.8\stylesheets\oasis-note-html.xsl
```

In each case the resulting file should be suitable for viewing in a web browser.

2.5 The OASIS specification XSL-FO stylesheet

There are four XSL-FO stylesheets for PDF creation in the package:

- `oasis-specification-fo-a4.xsl`

- `oasis-specification-fo-us.xsl`

- `oasis-note-fo-a4.xsl`

- `oasis-note-fo-us.xsl`

To convert a DocBook instance following the conventions used in this specification into an instance of XSL-FO assuming A4 page dimensions or US-letter page dimensions, use one of these stylesheets:

```
c:\oasis\spec-0.8\stylesheets\oasis-specification-fo-a4.xsl
c:\oasis\spec-0.8\stylesheets\oasis-specification-fo-us.xsl
c:\oasis\spec-0.8\stylesheets\oasis-note-fo-a4.xsl
c:\oasis\spec-0.8\stylesheets\oasis-note-fo-us.xsl
```

The resulting file should be suitable for an XSL-FO engine to render into a printed format such as a PDF file.

Note

Some XSL-FO engines will automatically translate relative URI strings in links to absolute URI strings as part of the formatting process. This will leave absolute references to your local file system in the generated PDF, rather than the desired relative URI strings needed. Usually the resulting PDF, when not encrypted or locked, can be hacked to manually translate the absolute URI strings to the required relative URI strings.

2.6 Stylesheet invocation parameters

There are four typical invocation parameters with defaults:

- `automatic-output-filename=`**`yes-or-no`**

  - when set to "`no`" (the default) the stylesheet processor invocation output target is not ignored

  - when set to "`yes`" the stylesheets will ignore the invocation output target and will automatically produce the output file named using the content of two child elements of `<article-info>` as follows:

    `<productname>-<productnumber>`

- `oasis-base=`**`yes-or-no`** (HTML only)

  - when set to "`no`" (the default) there is no addition of a `<base>` metadata element in the HTML result

  - when set to "`yes`" the stylesheets will create an HTML `<base>` metadata element extracting the URI from the first `<articleinfo><release-info>` element where the `role=` attribute contains the reserved partial string "`OASIS-specification-this`" and the URI contains the string "`.htm`"

- `html.stylesheet=`**`URI-string-to-OASIS-CSS-file`** (HTML only)

  - use this to inject a relative URI or other URI to your placement of the OASIS CSS file usually found in the publishing environment at "`css/oasis-spec.css`"

  - when not specified the URI that is used is hardwired to either your local publishing environment (when using the offline HTML stylesheet) or hardwired to the OASIS server publishing environment (when using the online HTML stylesheet)

- `oasis.logo=`**`URI-string-to-OASIS-logo-png`** (HTML only)

  - use this to inject a relative URI or other URI to your placement of the OASIS logo file usually found in the publishing environment base directory at "`OASISLogo.png`"

- when not specified the URI that is used is hardwired to either your local publishing environment (when using the offline HTML stylesheet) or hardwired to the OASIS server publishing environment (when using the online HTML stylesheet)

3 Authoring in XML

3.1 Structured editing

An important objective of using XML markup when writing content is to separate what you are writing from how it is formatted and presented. Moreover, describing the individual components of your writing uniquely can allow machine processing of your content. Such machine processing can identify constructs and process them individually as required for the processed result.

The vocabulary of XML markup is the level of granularity used to identify constituent information items, and the collection of element and attribute labels applied to the granules.

Applying styling to documents is an example of machine processing. The processed result is a formatted representation of your document. When many authors use the same vocabulary, or a given author creates many documents with the same vocabulary, a single set of processes will produce consistently formatted results across the document set.

This process is analogous to using styles found in most desktop publishing applications, however by removing from the author the ability to inject arbitrary formatting of information items in their content, two benefits are realized: (a) the author no longer needs to think about formatting, only about appropriately labeling the information items in the content; and (b) authors cannot inadvertently format components of a document with incorrect or inconsistent results.

Many writers do, however, prefer the use of their favorite desktop publishing applications and OASIS has posted at https://docs.oasis-open.org/templates/ a number of templates for widely-adopted word processing programs to be used in the creation of OASIS specifications. Users must be diligent in the use of styles in order to ensure their work conforms to the presentation conventions requested of OASIS. There is an obligation incumbent on the writer to verify their work has not violated the requested conventions, and there are no automated tools with which to validate the constraints have been respected.

When creating OASIS specifications using XML the burden of formatting is placed on the stylesheets, not on the writer. The obligation of the writer is only to be conformant to the DocBook document model for which the included stylesheets have been designed, and there are automated validation tools with which the writer can validate the constraints have not been violated. The writer is also obligated to follow conventions that reflect the requirements of the OASIS TC Administration for specification documents, though these are not validated by the DocBook document models. The conventions are, however, reflected in a template included in this package.

Any resulting violations to the formatting conventions can usually be ascribed to the stylesheets and, when repaired, the authored content usually does not need to change (barring any need to distinguish in the XML an ambiguity that is then being distinguished in the stylesheets).

3.2 DocBook vocabulary

This environment and methodology is based on using the `<article>` document type of the DocBook 4.5 [**DocBook**] vocabulary for authoring the content in XML markup. These explanatory documents do not attempt to teach the DocBook vocabulary, as there are many online and printed references and manuals on the use of DocBook.

The DocBook Document Type Definition (DTD) expresses the constraints of using elements and attributes, but not necessarily the conventions imposed by the OASIS TC Administration. Just having your specification validate with the DTD does not guarantee that you've used the correct or appropriate DocBook constructs to produce the required output. This package includes Section 3.5, " Pro forma templates" illustrating the conventional use of the DocBook constructs.

The [catalog.xml](#) OASIS Catalog will resolve DocBook and MathML DTD public identifiers to the appropriate DTD fragments.

To assert in the online OASIS environment that your XML document satisfies the constraints of DocBook, point to the posted copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd"
```

To assert in an embedded environment that your XML document satisfies the constraints of DocBook, point to the embedded copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "htmlruntime/spec-0.8/docbook/docbookx.dtd"
```

To assert in an offline local environment that your XML document satisfies the constraints of DocBook, point to the installation copy of the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "file:///c:/oasis/spec-0.8/docbook/docbookx.dtd"
```

or

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "file:///Users/admin/z/oasis/spec-0.8/docbook/docbookx.dtd"
```

Note that there is a layer of additional constraints imposed on top of the DocBook metadata constructs in order to appropriately identify the constituent metadata components of the OASIS requirements for specifications. No validation constraints for this additional metadata layer are included in this environment, so it is the responsibility of the writer to visually confirm these needs have been satisfied in the rendered results. Missing components are typically the result of the writer not having correctly used the metadata indicators within the standard DocBook constructs.

The additional constraints are illustrated and documented in [Section 3.5, " Pro forma templates"](#) and included with these materials.

3.3 DocBook content filtering

It is possible to have body content that is specific to the OASIS rendering while not being applicable to another rendering of the XML. This is signaled using:

```
condition="oasis"
```

If a `condition=` attribute exists and the content does not include the case-sensitive five-character string "`oasis`", then the content is not rendered using these stylesheets. An example of this is the content destined for the ISO ITTF rendering only, which has `condition="isosts"` and so is ignored by these stylesheets.

3.4 DocBook augmentations for OASIS specifications

These OASIS specification stylesheets recognize the following additional facilities or interpretations when using the DocBook vocabulary:

• `<articleinfo><productname>` (see [Section 2.6, "Stylesheet invocation parameters"](#))

- `<articleinfo><productnumber>` (see)

- `<articleinfo><releaseinfo role="OASIS-specification-this">`*{uri}*

  - reserved role identifying a URI belonging under the "This Version" heading

- `<articleinfo><releaseinfo role="OASIS-specification-previous">`*{uri}*

  - reserved role identifying a URI belonging under the "Previous Version" heading

- `<articleinfo><releaseinfo role="OASIS-specification-latest">`*{uri}*

  - reserved role identifying a URI belonging under the "Latest Version" heading

- release information role suffix "`-draft`"

  - the "*{uri}*" content is not interpreted as a URI but only simply as DocBook content

- release information role suffix "`-authoritative`"

  - the entry is suffixed with the string "(Authoritative)" as is required by OASIS conventions for one of the renditions

- `<articleinfo><releaseinfo role="committee">`*{text}*

  - identifying the technical committee section

- `<articleinfo><legalnotice role="related">`*{content}*

  - identifying the related work section

- `<articleinfo><legalnotice role="namespaces">`*{content}*

  - identifying the namespaces section

- `<articleinfo><legalnotice role="status">`*{content}*

  - identifying the status section

- `<articleinfo><legalnotice role="notices">`*{content}*

  - identifying the notices section

- `<phrase role="keyword">`*{text}*`</phrase>`

  - uppercase the text assuming the text is in English

- `<table role="font-size-XX">`

  - use a text font of XX for the content of cells in a table, as in `<table role="font-size-90%">`:

    Table 1. Table with font size of 90%

row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

    or as in `<table role="font-size-50%">`:

    Table 2. Table with font size of 50%

row 1, column 1	row 1, column 2

row 2, column 1	row 2, column 2

... rather than using the body font size:

Table 3. Table with no use of font size role attribute

row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

- note that the font actually rendered is up to the user agent and so any given size may end up being rendered using a built-in size (e.g. 90% interpreted as 100%)

- `<programlisting role="font-size-XX">`

  - use a text font of XX for the content of text in a program listing, as in `<programlisting role="font-size-80%">`:

    ```
    This is the text of a program listing where
    newlines are reflected in the output and spacing "     " is preserved.
    ```

    or as in `<programlisting role="font-size-50%">`:

    ```
    This is the text of a program listing where
    newlines are reflected in the output and spacing "     " is preserved.
    ```

    ... rather than using the body font size:

    ```
    This is the text of a program listing where
    newlines are reflected in the output and spacing "     " is preserved.
    ```

  - note that the font actually rendered is up to the user agent and so any given size may end up being rendered using a built-in size (e.g. 90% interpreted as 100%)

- `<?pb?>`

  - inject a page break (PDF only)

- `<?lb?>`

  - inject a line break

3.5 Pro forma templates

This environment includes pro forma templates of a DocBook XML instance for each of a specification and a note. These have placeholders for all the OASIS specification metadata found in the word processing templates. Each XML template has one of every item of metadata and can be used as a guideline when creating the metadata for a new document. The prose of the document includes a detailed narrative of the declaration and use of each of the metadata components.

The template also includes some limited guidelines to the use of the DocBook vocabulary that have been carried over from previous versions of the documentation. With time it is hoped to embellish more in this section for the benefit of the reader.

The pro forma secification ptemplate is included as part of the package of this environment and if the document you are reading is part of a completely installed collection then it can be found at template/spec-docbook-template-0.8-wd09.xml and rendered at template/spec-docbook-template-0.8-wd09.html. Please see Appendix B, *Sample specification template* for an analysis of the XML document.

Likewise, the pro forma note template also is included and it can be found at template/note-docbook-template-0.8-wd09.xml and rendered at template/note-docbook-template-0.8-wd09.html.

3.6 Real-world examples

An example of an OASIS Committee Specification that uses this DocBook XML environment is found at http://docs.oasis-open.org/ubl/os-UBL-2.2/.

An example of an OASIS Committee Note that uses this DocBook XML environment is found at http://docs.oasis-open.org/ubl/UBL-2.1-ASN.1/v1.0/cn01/.

In both examples the `htmlruntime/` subdirectory is part of the document's distribution, with the name of that directory changed to be `db/`.

In both examples the XML version of the document is the authoritative version of the document. At this time of writing, the "latest version" of the XML document is not supported on the OASIS server; navigate to the latest version of the HTML document and use the "this version" of the XML from that page.

3.7 Stylesheet association

Not included in the final published XML source of OASIS specifications are the stylesheet association [**xml-assoc**] processing instructions that are very handy conveniences to use during the authoring process. This methodology mandates their removal from the final published documents, but encourages their use when writing in order to streamline the writer's review of the formatted content.

Stylesheet association is not widely employed in the general XML community, typically due to (a) limited awareness by writers of their benefit; and (b) limited conforming support in web browsers.

An XSLT stylesheet association processing instruction is structured with two pseudo attributes as follows, indicating the type of the stylesheet being engaged and the location of the stylesheet file producing the HTML rendition:

```
<?xml-stylesheet type="text/xsl" href="place-URL-here"?>
```

The benefits of having embedded a stylesheet association processing instruction at the top of one's XML document are realized by opening the XML document being written in a web browser that (a) is aware of the processing instruction; (b) has a conforming XML processor with which to process the stylesheet files; and (c) has a conforming XSLT processor with which to engage the stylesheet files. The act of opening the XML document in the browser engages the HTML stylesheet against the XML content producing the HTML rendition on the browser canvas. At any time during the writing process, merely refreshing the web browser canvas re-engages the stylesheet producing the rendition of the revised content. This removes the necessity to engage the standalone invocation of the stylesheet environment to produce a reviewable rendition.

For example, this document was authored in an environment where the locally-installed offline version of the publishing environment is available in the local directory `../spec-0.8/`, thus allowing the following stylesheet association processing instruction placed at the top of the XML file before the document type declaration to render the document in an XSLT-aware web browser:

```
<?xml-stylesheet type="text/xsl" href="file:///Users/admin/z/oasis/
spec-0.8/stylesheets/oasis-specification-html.xsl"?>
```

When a document is being authored in an environment where a copy of this publishing environment is embedded with the document directories, the following stylesheet association processing instruction would point to the embedded copy of the stylesheet fragment:

```
<?xml-stylesheet type="text/xsl"
                 href="htmlruntime/spec-0.8/stylesheets/oasis-specification-html.xsl"?>
```

Note

When embedding this stylesheet environment in a document's distribution environment, it is important to prune the stylesheet directories to retain only that which is required for HTML

rendering. The only directories needed in the spec-0.8/docbook/xsl directory are: xsl/common, xsl/html and xsl/lib. All other directories should be removed.

When a document is being authored in an environment where the remotely-installed online version of the publishing environment is to be used, the following stylesheet association processing instruction would work (modulo the latency aspects of accessing resources over the Internet):

```
<?xml-stylesheet type="text/xsl" href="http://docs.oasis-open.org/
templates/DocBook/spec-0.8/stylesheets/oasis-specification-html.xsl"?>
```

Many people believe that stylesheet association is a transient property of an XML document and should not be included in any "official" normative XML content. This methodology, therefore, mandates that any such processing instruction that might be used by the author during the writing phase be removed from the final published XML instance.

3.8 Document writing rules

Included in the package is the Schematron file oasis-spec-note.sch and sample implementation oasis-spec-note.xsl, with the set of writing integrity and style rules for OASIS documents used by the UBL TC in the creation of the UBL hub document using the oXygenXML editing tool. Engaged while writing the document, these rules highlight problem areas that need to be addressed, while providing Schematron Quick Fixes for some of them to help automate the edits required.

Some of the integrity rules are employed for consistency checks. Other such rules enforce content structuring required by the OASIS Technical Advisory Board. Yet other rules promote quicker transformation to ISO format for ITTF publishing of International Standards.

At the time of writing the integrity rules include:

- sections and appendixes cannot have hanging content, that is, each must contain only subsections or no subsections, but not a mixture of subsections and other content

- paragraphs cannot have leading white space or be empty or contain lists or block quotes

- `<ulink>` must have the same content value as the `url=` value (unless the element has the attribute `conformance="skip"`)

- text content cannot contain the BOM (`&#xfffe;`) character

- a paragraph cannot contain a itemized list, ordered list or block quote as these are considered block-level constructs that are peers to paragraphs

- bibliographic entries must have an identifier and must be referenced from somewhere or it will be flagged as an error … use `<bibliomixed conformance="skip"` to allow the entry to be unreferenced

The content and style rules are not engaged by default. The `<article>` document element can have a `conformance=` attribute with name tokens indicating various style rules to employ. For example:

```
<article conformance="quotes eBuzzwords ids">
```

At the time of writing the style rules include those engaged with the following case insensitive name tokens:

- `eBuzzwords`

  - in all text an eBuzzword (e.g. "eCommerce") must be indicated to be a phrase with no language as in `<phrase lang="none">eCommerce</phrase>` so as not to interfere with spell-checking

- ids

  - every section, appendix, table and figure must have an `id=` whose value is its title, converted to upper case, stripped to only word characters and digits, spaces and slashes converted to hyphens, and prefixed with "S-", "A-", "T-" or "F-" as required (unless the section has the attribute `conformance="skip"`)

- quotes

  - an ASCII single quote in non-literal text is flagged that it should be the `’` entity; if you really need the ASCII ' character then mark an ancestral element to skip conformance as in: `<para conformance="skip">...'...</para>`

  - an ASCII double quotes in non-literal text are flagged that they should be the `“` and `”` entities; if you really need the ASCII " character then mark an ancestral element to skip conformance as in: `<para conformance="skip">...'...</para>`

3.9 MathML

Following the guidance from http://www.sagehill.net/docbookxsl/MathML.html the DOCTYPE declaration of the document must redefine the content entities for equations as follows to support MathML, changing the prefix as required for your MathML**[MathML]** content.

When the OASIS HTML runtime directory is available in your environment, you can point to it:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
  "db/spec-0.8/docbook/docbookx.dtd"
[
<!ENTITY % MATHML.prefixed "INCLUDE">
<!ENTITY % MATHML.prefix "mml">
<!ENTITY % equation.content "(alt?, (graphic+|mediaobject+|mml:math))">
<!ENTITY % inlineequation.content
              "(alt?, (inlinegraphic+|inlinemediaobject+|mml:math))">
<!ENTITY % mathml PUBLIC "-//W3C//DTD MathML 2.0//EN"
        "htmlruntime/spec-0.8/mathml2/mathml2.dtd">
%mathml;
]>
```

When the OASIS HTML runtime directory is not available in your environment, you can point to the online DTD, noting that it will take the time to parse by going online to fetch the DTD:

```
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
  "db/spec-0.8/docbook/docbookx.dtd"
[
<!ENTITY % MATHML.prefixed "INCLUDE">
<!ENTITY % MATHML.prefix "mml">
<!ENTITY % equation.content "(alt?, (graphic+|mediaobject+|mml:math))">
<!ENTITY % inlineequation.content
              "(alt?, (inlinegraphic+|inlinemediaobject+|mml:math))">
<!ENTITY % mathml PUBLIC "-//W3C//DTD MathML 2.0//EN"
        "http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">
%mathml;
]>
```

There is stylesheet support for both block-level equations:

Equation 1. Test of MathML standalone

$$C_\mathrm{t} = \frac{M}{F}$$

… and inline equations $C_\mathrm{t} = \frac{M}{F}$ included, but the XSL-FO processor must support MathML to render to PDF. Note that the free invocations in Section C.3, " Sample invocations" do not support MathML in PDF.

4 Validation

For an XML document to be properly published into the target results, it must pass both Section 3.2, " DocBook vocabulary" and Section 3.8, "Document writing rules" constraints.

In the "validation/" subdirectory is the `test.bat` example of calling `validate.bat`, and the `test.sh` example of calling `validate.sh`, for candidate OASIS DocBook files, the first with no problems, the second with bad XML syntax, and the third with a writing rule violation:

```
#############################################################
Validating good.xml
#############################################################
============== Phase 1: DTD validation ==============
No DTD validation errors.
============ Phase 2: Writing rules validation ============
No writing rule validation errors.

#############################################################
Validating syntaxbad.xml
#############################################################
============== Phase 1: DTD validation ==============
org.xml.sax.SAXParseException; systemId: syntaxbad.xml;
 lineNumber: 247; columnNumber: 5; Element type "section" must be followed by
 either attribute specifications, ">" or "/>".
    at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
    at org.apache.xerces.jaxp.SAXParserImpl$JAXPSAXParser.parse(Unknown Source)
    at org.apache.xerces.jaxp.SAXParserImpl.parse(Unknown Source)
    at javax.xml.parsers.SAXParser.parse(SAXParser.java:274)
    at com.nwalsh.parsers.XJParser.dtdParse(Unknown Source)
    at com.nwalsh.parsers.XJParser.parse(Unknown Source)
    at com.nwalsh.parsers.XJParse.run(Unknown Source)
    at com.nwalsh.parsers.XJParse.main(Unknown Source)
Exception in thread "main" java.lang.NullPointerException
    at com.nwalsh.parsers.XJParser.printParseStats(Unknown Source)
    at com.nwalsh.parsers.XJParse.run(Unknown Source)
    at com.nwalsh.parsers.XJParse.main(Unknown Source)
Attempting validating, namespace-aware parse
Fatal error: syntaxbad.xml:247:5:Element type "section" must be followed by
 either attribute specifications, ">" or "/>".

#############################################################
Validating rulesbad.xml
#############################################################
============== Phase 1: DTD validation ==============
No DTD validation errors.
============ Phase 2: Writing rules validation ============
The id= identifier must match the title.: /article/section[1]

Error at char 7 in xsl:value-of/@select on line 209 column 83 of oasis-spec-note.xsl:
  XTMM9000: Processing terminated by xsl:message at line 209 in oasis-spec-note.xsl
Processing terminated by xsl:message at line 209 in oasis-spec-note.xsl
```

As a convenience to users of DocBook not following the OASIS rules, the `dtd.bat` and `dtd.sh` called by the validation can be invoked independently. OASIS project editors are encouraged to do full validation including the document writing rules.

5 Vendor print extensions

The XSL-FO generated by these print stylesheets automatically engage the document metadata extensions supported by both Antenna House and RenderX. Other document-wide extensions can be specified by the user by using processing instructions as children of the document element.

For example, turning on simple line numbering using the Antenna House extension can be done using:

```
<article status="Committee Specification Draft 01">
  <?axf line-number show?>
  <articleinfo>
...
```

Many line-numbering features can finesse the presentation of the line numbers as described at http://www.antennahouse.com/product/ahf60/docs/ahf-ext.html#line-number, for example:

```
<article status="Committee Specification Draft 01">
  <?axf line-number show?>
  <?axf line-number-position end?>
  <?axf line-number-offset 2mm?>
  <?axf line-number-width 20pt?>
  <?axf line-number-text-align start?>
  <?axf line-number-font-family serif?>
  <?axf line-number-font-size 10pt?>
  <?axf line-number-font-weight normal?>
  <?axf line-number-font-style italic?>
  <?axf line-number-initial 1?>
  <?axf line-number-start 5?>
  <?axf line-number-interval 5?>
  <?axf line-number-reset page?>
  <articleinfo>
...
```

Other document-wide attribute extensions for Antenna House can be specified using the processing instruction:

```
<?axf {attribute-name} {attribute-value}?>
```

Document-wide attribute extensions for RenderX can be specified using the processing instruction:

```
<?rx {attribute-name} {attribute-value}?>
```

Document-wide attribute extensions for other XSL-FO engines can be specified using the processing instruction:

```
<?ext {extension-namespace-URI} {attribute-name} {attribute-value}?>
```

Note in all cases any white-space sequences in the attribute value are collapsed to single spaces before the attribute is added to the `<fo:page-sequence>` element of the XSL-FO result for the content. No quotes are used for the attribute value, even if the value includes spaces, as the attribute begins with the first character after the white-space characters after the attribute name, and ends with the last character before the "`?>`" sequence.

Line numbers are turned off for tables when rendering with Antenna House.

Appendix A Acknowledgments (Non-Normative)

This specification environment was written with the generous and appreciated assistance of Paul Knight, Chet Ensign, Mary McRae, Will Norris and Jon Bosak.

Appendix B Sample specification template

This section describes the instance structure and metadata of the DocBook XML instance template/spec-docbook-template-0.8-wd09.xml, a template geared for use with the specification stylesheets. The summarized markup copied from the instance is split into successive `programlisting` fragments interspersed with a narrative regarding how to use the markup in the fragment. Braces "{}" are used to indicate replaceable information and the braces should not be in the final document. There is no obligation to use the techniques used below, though what is below is successfully producing the result you are reading.

For the purposes of example in the code below, the locally-installed copy of the OASIS publishing environment is in the directory `c:/oasis/spec-0.8/`.

The XML Declaration uses `ISO-8859-1` instead of `UTF-8` as the content is being typed and remembering the two-byte codes of `UTF-8` is sometimes awkward. Alternatively, using `US-ASCII` limits the typing entirely to 7-bit ASCII characters, requiring accented and other high-bit characters to be entered with entities.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

The following is an editing convenience keeping in an XML comment the PUBLIC and SYSTEM identifiers for the document type declaration for each of the online publishing environment, the embedded publishing environment and the local publishing environment. Note how the actual identifiers in this snippet are for the local publishing environment while in the instance you will find the online publishing environment set being used in the document type declaration. When switching back and forth it is a convenience to have a few lines handy for copy/paste, and these are provided at the top of this file (note the URIs for the embedded environment need to be adjusted according to where the directory is located; this snippet is appropriate for this specification document but not for the template location):

```
<!--
For use when a committee document points at the OASIS web site for publishing:
<?xml-stylesheet type="text/xsl"
href="http://docs.oasis-open.org/templates/DocBook/spec-0.8/stylesheets/oasis-specification-html.xsl"?>
<!DOCTYPE article
   PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd"

For use when a committee document points to an embedded installation of DocBook
(note the instructions for embedding DocBook requires pruning some directories
and adjusting the stylesheet and DocBook directories in these declarations):
<?xml-stylesheet type="text/xsl"
      href="../htmlruntime/spec-0.8/stylesheets/oasis-specification-html.xsl"?>
<?oasis-spec-base-uri ../htmlruntime/spec-0.8/?>
<!DOCTYPE article
 PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "../htmlruntime/spec-0.8/docbook/docbookx.dtd"

For use when a committee document is published in a local environment only
(note the instructions for local publishing require adjusting the stylesheet
 and DocBook directories in these declarations):
<?xml-stylesheet type="text/xsl"
href="file:///Users/admin/z/oasis/spec-0.8/stylesheets/oasis-specification-html-offline.xsl"?>
<!DOCTYPE article
   PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
         "file:///Users/admin/z/oasis/spec-0.8/docbook/docbookx.dtd"
-->
<?xml-stylesheet type="text/xsl"
```

```
href="http://docs.oasis-open.org/templates/DocBook/spec-0.8/stylesheets/oasis-specification-html.xsl"?>
<!DOCTYPE article
  PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
        "http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd"
[
```

In the above the XML Stylesheet Association [**xml-assoc**] processing instruction is used to render the XML as HTML for the reader. This is also useful during the authoring process as a convenience for reviewing the rendered version of the authored XML markup when the XML document is open in a conforming browser supporting the association of XSLT stylesheets. After saving any edits to the XML document it is only necessary to refresh the browser to view the rendered results, without requiring any intervening XSLT process.

The internal declaration subset uses general entities in order to ensure consistent use of strings throughout the document. The `&name;` entity has the base name of the document. The `&version;` entity has the version of the document. The `&standard;` has the document status which is used in meta data and is often referenced in the prose along the lines of "`This &standard; includes the ...`".

```
[<!--the document properties-->
<!ENTITY name "spec-docbook-template">
<!ENTITY pversion "0.7">
<!ENTITY version "0.8">
<!ENTITY stage "wd09">
<!ENTITY stagetext "Stage and revision">
<!ENTITY standard "Stage and revision / Alternate stage and revision">
<!ENTITY this-loc   "https://docs.oasis-open.org/templates/DocBook/spec-&version;/template">
<!ENTITY previous-loc "http://docs.oasis-open.org/templates/DocBook/spec-&pversion;/template">
<!ENTITY latest-loc   "https://docs.oasis-open.org/templates/DocBook/oasis-specification/template">
<!ENTITY pubdate "03 September 2020">
```

The document element for OASIS standards is always `article`. The status, which can have different simultaneous status values separated by "/", is reflected in an attribute:

```
<article status="&standard;">
```

The metadata is kept in the DocBook `articleinfo` element, starting with the source code control, the name and the version of the document. The stylesheets can use the these information items in the synthesis of filenames (see Section 2.6, "Stylesheet invocation parameters"). Note the use of the entity references for consistency.

```
    <articleinfo>
    <title>Specification Title Version X.X</title>
    <releaseinfo role="cvs"> $Id: spec-docbook-template-0.8-wd09.xml,v 1.1
2020/09/03 19:40:23 admin Exp $ </releaseinfo>
    <productname>&name;</productname>
    <productnumber>&version;-&stage;</productnumber>
```

The links to the various instantiations of the document and supporting materials are summarized using `releaseinfo` elements with the reserved roles (see Section 3.4, "DocBook augmentations for OASIS specifications") for as many files as you have. OASIS rules require the identification of the authoritative publication. Note again the use of the entity references for consistency.

```
<releaseinfo role="OASIS-specification-this-authoritative"
  >&this-loc;/&name;-&stage;.html</releaseinfo>
<releaseinfo role="OASIS-specification-this"
  >&this-loc;/&name;-&stage;.pdf</releaseinfo>
<releaseinfo role="OASIS-specification-this"
  >&this-loc;/&name;-&stage;.xml</releaseinfo>
```

```
<releaseinfo role="OASIS-specification-previous-authoritative"
  >&previous-loc;/&name;.html</releaseinfo>
<releaseinfo role="OASIS-specification-previous"
  >&previous-loc;/&name;.pdf</releaseinfo>
<releaseinfo role="OASIS-specification-previous"
  >&previous-loc;/&name;.xml</releaseinfo>

<releaseinfo role="OASIS-specification-latest-authoritative"
  >&latest-loc;/&name;.html</releaseinfo>
<releaseinfo role="OASIS-specification-latest"
  >&latest-loc;/&name;.pdf</releaseinfo>
<releaseinfo role="OASIS-specification-latest"
  >&latest-loc;/&name;.xml</releaseinfo>
```

The committee is indicated using `role="committee"`.

```
<releaseinfo role="committee"><ulink url="https://www.oasis-open.org/
committees/">OASIS {official name of technical committee} TC</ulink>
</releaseinfo>
```

The various people sited in the document are in the `authorgroup` element, using `editor` and `author` for their respective players, and the `othercredit` element for the committee chairs.

```
<authorgroup>
  <editor>
    <firstname>Jane</firstname>
    <surname>Doe</surname>
    <affiliation>
      <orgname><ulink url="www.example.com">Example
        Corporation</ulink></orgname>
    </affiliation>
    <email>jane.doe@example.com</email>
  </editor>
  <othercredit>
    <firstname>Mary</firstname>
    <surname>Baker</surname>
    <affiliation>
      <orgname>Another Corporation</orgname>
    </affiliation>
    <email>mary.baker@example.com</email>
  </othercredit>
  <othercredit>
    <firstname>James</firstname>
    <surname>Butcher</surname>
    <affiliation>
      <orgname>Associate Member</orgname>
    </affiliation>
    <email>james.butcher@example.com</email>
  </othercredit>
</authorgroup>
```

The publishing date is constrained by OASIS conventions to the following format:

```
<pubdate>&pubdate;</pubdate>
```

The copyright information is embedded using the `copyright` element, but note that this information has to be entered as well in the text of the notices.

```
<copyright>
  <year>YYYY</year>
  <holder>OASIS Open, Inc. All Rights Reserved.</holder>
</copyright>
```

When you want to relate this document to additional or related works use `legalnotice` with role="additional" or role="related".

```
<legalnotice role="additional"><title>Additional Work Product artifacts</title>
<para>This prose specification is one component of a Work Product which also includes:</para>
...
</legalnotice>


<legalnotice role="related"><title>Related Work</title>
<para>This specification replaces or supersedes:</para>
...
</legalnotice>
```

When you want to document the namespaces use `legalnotice` with role="namespaces".

```
<legalnotice role="namespaces">
  <title>Declared XML namespaces</title>
  <para><ulink url="http://docs.oasis-open.org/ns/{tc-shortname}/xxxx"
      >http://docs.oasis-open.org/ns/{tc-shortname}/xxxx</ulink></para>
</legalnotice>
```

The abstract has its own DocBook element.

```
<abstract>
  <para>{This specification defines...}</para>
</abstract>
```

For the status use `legalnotice` with role="status".

```
<legalnotice role="status"><title>Status</title>
  <para>{Describe the status and stability ...}</para>
  ...
</legalnotice>
```

For the citation use `legalnotice` with role="citation" and take advantage of the entities.

```
<legalnotice role="citation">
<title>Citation format</title>
<para>When referencing this specification the following citation format
  should be used:</para>
<bibliolist>
  <bibliomixed>
    <abbrev>&name;</abbrev>
    <citetitle>Specification Title Version X.X.</citetitle>
    <date>&pubdate;. </date>
    <releaseinfo>OASIS Committee Specification Draft &stage;. </releaseinfo>
    <bibliomisc><ulink url="&this-loc;/&name;.html"
        >&this-loc;/&name;.html</ulink>.</bibliomisc>
  </bibliomixed>
</bibliolist>
</legalnotice>
```

For the notices use `legalnotice` with role="notices" ensuring you use the appropriate version of the legal notices from the markup in this instance.

```
<legalnotice role="notices"><title>Notices</title>
  <para>Copyright © OASIS Open 20??. All Rights Reserved.</para>
  ...
</legalnotice>
```

Thus ends the document metadata. The sections and annexes follow.

```
</articleinfo>
<section id="s.introduction">
...
<appendix>
...
```

Appendix C Publishing choreography and orchestration (Non-Normative)

C.1 Automated processing

There is no obligation to automate the publishing process by choreographing the invocation of validation and production processes, however this methodology offers example orchestrations of these. Note that but for not having a core Ant [**ant**] task for the invocation of an XSL-FO processor, these guidelines would include such an example using that cross-platform Java-based build tool. It that absence, therefore, this section describes both a Windows command prompt and a Linux shell prompt that you may wish to configure for your own use.

The sample choreography invokes three separate processes in turn to produce the results. These processes are made available by Windows batch files and Linux shell scripts that are not included.

C.2 Linux shell script preparation

The example Linux shell script choreography in template/spec-docbook-template.sh assumes the presence of three support shell scripts summarized as follows which are not supplied but see Section 4, "Validation" for examples for DTD validation and XSLT transformation:

- ```
  sh xmldtd.sh  input-XML-file
  ```

- ```
  sh xslt.sh  input-XML-file  stylesheet-XSL-file  output-file
  ```

- ```
  sh xslfo-pdf.sh  input-FO-file  output-PDF-file
  ```

C.3 Sample invocations

The stylesheets have been tested with Saxon for XSLT and Antenna House, Ibex and RenderX for XSL-FO.

While any conforming XML, XSLT and XSL-FO engine can be used in a given environment, the following Java-based applications and their respective invocations are suitable. There is no obligation, however, to use these as any conforming implementations of validation, transformation and pagination can be used. Note these guidelines leave it up to the reader to download the required support and to appropriately set the required CLASSPATH environments for the following invocations to work.

To validate that an XML instance conforms to the constraints of a formal DTD expression, xjparse [**xjparse**] can be used to satisfy the xmldtd script as follows:

```
java -jar xjparse.jar -v %1
```

To transform and XML instance using and XSLT expression, Saxon [**Saxon**] can be used to satisfy the xslt script as follows:

```
java -jar saxon.jar -o %3 %1 %2
```

To transform an XSL-FO instance into a PDF file, FOP [**FOP**] can be used to satisfy the xslfo-pdf script as follows (though note that there may be some conformance challenges with FOP that might present some problems):

```
java org.apache.fop.apps.Fop -fo %1 -pdf %2
```

To transform an XML instance without MathML into PDF in a single step, there are the stylesheets/xslfo-spec-pdf.bat and stylesheets/xslfo-note-pdf.bat example invo-

cation files in the package. These invocation take three arguments (in order): the XML input instance, "us" or "a4" (no quotes) to indicate the page size, and the name of the PDF output file. This requires the `saxon.jar` XSLT 1.0 executable**[Saxon]** and the Ibex Signature Edition created for stylesheets from Crane Softwrights**[Ibex Signature Edition]** (renamed as `ibex-crane-ss.jar`). The digital signature manifest `oasis-specification-fo-manifest.xml` is tied to the stylesheets as published, thus the processor will not work with these stylesheets if they are modified in any way. The support file `ibexconfig.xml` is used to configure font information. The version of Ibex available through Crane does not support MathML.

Appendix D Revision History (Non-Normative)

Revision 0.8 wd09	03 September 2020	gkh
Symbol fonts, user extension of result HTML header content		
Revision 0.8 wd08	05 April 2020	gkh
Updates to HTML generation supporting MathML; add validation		
Revision 0.8 wd07	13 November 2019	gkh
Updates to Schematron writing rules for editing; clean up HTML titles; minor doc fixes		
Revision 0.8 wd06	03 May 2019	gkh
Update to DocBook stylesheet library version 1.79.1; separate Schematron writing rules between those for integrity and those for style		
Revision 0.8 wd05	25 February 2019	gkh
Support processing instruction for relative location of HTML support files		
Revision 0.8 wd04	04 January 2019	gkh
Tweak HTML CSS for images		
Revision 0.8 wd03	02 December 2018	gkh
New note layout from TC Administration		
Revision 0.8 wd02	22 September 2018	gkh
Repaired line numbering for note cover page		
Revision 0.8 wd01	08 September 2018	gkh
New logo and colours from OASIS		
Revision 0.7 wd05	13 September 2017	gkh
Line numbering. Latest TCAdmin tweaks. Latest SQF actions. New status text in templates.		
Revision 0.7 wd04	24 January 2017	gkh
Latest Ibex manifest. Latest SQF actions. New status text in templates.		
Revision 0.7 wd03	19 December 2016	gkh
Tweak URI writing for OASIS logo graphic for non-relative resolution for print stylesheets; latest Ibex manifest		
Revision 0.7 wd02	03 December 2016	gkh
Add vendor print extensions (e.g. line numbering); updated writing rules		
Revision 0.7 wd01	08 November 2016	gkh
Add (and follow) document writing rules and reference external package for ISO Directives Part 2 conversion.		
Revision 0.6 wd12	10 December 2015	gkh
Fix case in "Non-normative" and add display of "Normative".		
Revision 0.6 wd11	7 May 2015	gkh
Accommodate OASIS server redirect. Suppress PDF ulink exposition.		
Revision 0.6 wd10	17 April 2014	gkh
Repair packaging for notes. Rewrite footers to support long project names.		
Revision 0.6 wd09	17 July 2013	gkh
Package update to include stylesheets for notes. Tweak footer column widths to fit copyright in A4 rendition.		
Revision 0.6 wd09	17 February 2013	gkh
Repaired copyright holder logic and boilerplate		
Revision 0.6	14 June 2012	gkh
Latest TC process revisions and TC Admin review		
Revision 0.5	8 July 2010	gkh
New template structure and filename conventions; mimic latest XHTML template		
Revision 0.4	03 Feb 2006	gkh
New IPR and use of revised 0.4 specification publishing environment; mimic latest Word and Open Office templates		
Revision 03	15 Aug 2002	ndw
Changed copyright holder.		
Revision 02	28 May 2002	ndw

Added IPR section.			
Revision 01	26 Apr 2002	ndw	
Reworked after conversations with Eve.			
Revision 00	25 Apr 2002	ndw	
First draft.			